Open Journal of Optimization, 2026, 15(1), 1-21
https://www.scirp.org/journal/ojop
ISSN Online: 2325-7091
ISSN Print: 2325-7105

A Modified Primal-Dual Interior Point Method for Solving Convex Quadratic Optimization Problems

John Avoka[1], Stephen B. Twum[2], Christian John Etwire[2]

[1]Department of Applied Mathematical Sciences, Faculty of Computing, Engineering and Mathematical Sciences, Catholic University of Ghana, Sunyani, Ghana
[2]Department of Mathematics, School of Mathematical Sciences, University of Technology and Applied Sciences, Navrongo, Ghana
Email: javoka08@gmail.com, stwum@cktutas.edu.gh, cetwire@cktutas.edu.gh

How to cite this paper: Avoka, J., Twum, S.B. and Etwire, C.J. (2026) A Modified Primal-Dual Interior Point Method for Solving Convex Quadratic Optimization Problems. *Open Journal of Optimization*, **15**, 1-21.
https://doi.org/10.4236/ojop.2026.151001

Received: February 7, 2026
Accepted: March 28, 2026
Published: March 31, 2026

Abstract

This study presents a modified primal-dual interior point method (MPD-IPM) for solving convex quadratic optimization problems. The modification is performed through linearization of the central path and the introduction of an improved initialization strategy derived from the derivative structure of the Lagrangian function, unlike the classical approach, whose iterates follow a nonlinear trajectory. The proposed formulation generates linearized subsidiary constraint equations that reduce curvature effects during path-following. Numerical experiments conducted on benchmark and hypothetically generated problems demonstrate improved iteration counts, enhanced convergence reliability, and reduced computational time, particularly for large-scale instances.

Keywords

Convex Quadratic Programming, Interior-Point Methods, Central Path, Lagrangian Function, Numerical Optimization

1. Introduction

Interior point methods (IPMs) have emerged as one of the most popular classes of methods for solving constrained nonlinear optimization problems in general. Basically, the solution of interior point methods starts from the interior of the region (which is feasible) and not on the boundary of the feasible region [1]. Interior point methods have the ability to explore the sparsity of problems in general due to their robustness to ill-conditioned problems. Among them is the class of Primal-Dual Interior Point methods (PD-IPMs) which has been of particular interest in this research. Studies of it reveal that generally, the iterates move along a curvy path, known as the central path, to optimality. The central path parameter is usually a non-negative parameter which can be described as a perturbation of the optimality conditions for the problem. At each iteration, the parameter serves as a regularization of the linear equations which are solved for in convex optimization problems in particular [2].

Another issue with IPMs in general [1], is that it can be difficult when selecting the value of an initial point of the central path parameter which does not have to be too large (a parameter value that is large is not desirable due to the fact that it takes so long to reduce it to a small target value). In other words, they are not easy to initialize or warm-start, which means the solution of a previous problem is not suitable as an initial start point to a similar problem, due to its inability to enhance convergence.

Active-Set methods are normally preferred over interior-point methods due to the ability of the active set methods to "warm start". The warm start provides a good measure of the optimal active set which is used to initialize the algorithm [3]. Conventionally, active-set methods comprise phases I and II: phase I aims at feasibility and phase II aims at optimality. Warm-start is suitable for the simplex methods used for solving linear optimization problems, especially when the problem size is so huge and the run time of the solution is long. Mostly, for large scale problems, IPMs take fewer iterations to optimality than their active-set counterparts. Since the interior-point methods solve for values of all the variables in the linear systems of the optimization problems at each iteration, this makes interior point methods computationally expensive. However, in active-set methods only a subset of the values of the variables of the linear systems are found [4]. When the problem warm-starts, several iterations are needed to put the iterates on the central path since the warm point is usually not closer to the central path and this causes difficulties for path-following interior point methods [3].

In contrast, a cold start is when an advanced point is not known to be used as a startup. The authors in [5] stated that the simplex method works better than the interior point methods when it comes to solving small problems due to the simplex method having better warm-start points than its IPMs counterparts.

Generally, warm-starting is very essential in most optimization problems whereby increasing the possibility of reaching optimality in fewer iterations may be achieved. We have observed that in the central path neighborhood, the iterates do not move exactly on the path but try to be closer to the central path until optimality is obtained. This is due to the fact that the path of interior point methods is nonlinear.

1.1. The Primal-Dual Interior Point Method

Generally, path following IPMs literally remove the inequality constraints by incorporating them into the objective function with the aid of the logarithmic barrier function [6]. The resulting subproblem is solved approximately to find the optimal solution.

Consider a convex quadratic optimization problem (QP) below.

$$\min f(x) = c^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} Qx \tag{1.1}$$

$$\text{subject to} \quad Ax - b = 0.$$

$$x \geq 0.$$

where $A \in R^{m \times n}$, $b \in R^m$, and $c \in R^n$, the matrix $Q \in R^{n \times n}$ is a positive semidefinite matrix $x, c \in R^n$ and also $A \in R^{m \times n}$ has a full row rank $m \leq n$.

The original problem (1.1) is also known as the primal problem which has an associated dual problem derived from the primal problem. The dual problem of (1.1) is given as (1.2);

$$\max b^{\mathrm{T}} y - \frac{1}{2} x^{\mathrm{T}} Qx \tag{1.2}$$

$$\text{subject to} \quad A^{\mathrm{T}} y + s - Qx = c.$$

$$y \text{ free}, \quad s \geq 0.$$

where $s \in R^n$ and $y \in R^m$. The logarithmic barrier function, $-\mu \sum_{j=1}^{n} \ln x_j$, is introduced to the objective function of (1.1). The logarithmic barrier function, $-\mu \sum_{j=1}^{n} \ln x_j$, is used as a substitute for the inequality constraint $x \geq 0$ in the primal problem as in (1.3).

$$\min f(x) = c^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} Qx - \mu \sum_{j=1}^{n} \ln x_j \tag{1.3}$$

$$\text{subject to} \quad Ax - b = 0.$$

The positive scalar, u in (1.3) is known as the barrier parameter. It is worth noting that the barrier parameter controls the relation between the barrier term and the original problem (1.1).

To prevent the points x from approaching the boundaries of the feasible region (at the boundary, the barrier term blows up) a large value of u is applied to (1.3). The barrier term becomes less influential when u becomes smaller in value and so much attention is paid to the original problem.

If u is continuously reduced from a larger value to zero, the corresponding solutions $x(u)$ forms a path that moves towards an optimum of (1.3). This path is known as the primal central path because it only makes use of the primal problem. Also, a similar central path can be derived for the dual problem which converges to analytic center of the dual solution set. This type of IPMs that makes use of the primal-dual central path is known as the primal-dual interior point method. These methods are considered as the most successful IPMs [7].

The path following methods select a type of neighborhood such as a Euclidean norm, N_2 or an infinity norm, N_∞, and then choose a centering parameter, σ and a step length parameter, α to ensure that every iterate (x^k, y^k, s^k) stays within the chosen neighborhood.

1.2. Some Previous Works

The central path is very important for improved performance in IPMs. Barrier functions are usually used to define the central path. There are so many barrier functions that are used in interior point methods but mostly, the logarithmic function is used.

Generally, the central path is our guide to a strict complementarity solution. Due to the nonlinearity of the equation that determines the central path, iterates do not stay on the central path even if the initial interior point is perfectly centered [8]. In this light, the proximity measure is used to control and keep iterates in an approximate neighborhood of the central path. Usually, this measure depends on the current x and s, and a value of u on the central path. The proximity measure quantifies how close the iterate is to the point corresponding to u on the central path. Generally, care needs to be taken to handle rank deficient Jacobians. [9] proposed a primal-dual interior-point method that was penalized for convex quadratic programs and for appropriate values of the parameters, the original problem solution is recovered by the algorithm. However, the iterations for this method did not follow the central path to optimality but moved within the central neighborhood until an optimal solution was attained.

[10] developed a continuous trajectory which converges for convex optimization problems with linear constraints provided certain conditions are met. They modified the original central path substantially, and this resulted in obtaining interior-point continuous trajectories that provide solutions to ordinary differential equation (ODE) systems [10]. The iterates move along the central paths to and fro and so the iterates do not stay on a straight/central path. Also, IPMs possess an unequalled ability to identify the essential subspace in which the optimal solution is hidden [11]. Interior point methods usually add the log barrier to the objective function. The log barrier is equivalent to the complementarity condition. The complementarity condition forms a path that is nonlinear in nature and so the iterates move in the central path neighborhood until optimality is attained [11].

[12] developed a PD-IPM that is regularized. Their algorithm is used for solving quadratic programming problems by applying PD-IPM for fewer of iterations, using a starting point that is not feasible. In their algorithm, the iterations do not stay on the central path throughout the iterations.

2. Development of the Modified PD-IPM (MPD-IPM)

In this section, we demonstrate how the classical PD-IPM works and subsequently demonstrate how it is modified. Three cases of the problem are considered: those involving only equality constraints; those involving inequality constraints; and those with mixed constraints. The PD-IPM is modified as far as the linearization of the path and initialization of iterates are concerned. The existing PD-IPM applies a logarithmic barrier function $-\mu\sum_{j=1}^{n}\ln x_j$ to the objective function of (1.3) to replace the non-negativity constraint $x \geq 0$, so that iterates move closer to but not on the boundary as:

$$\min\left(f(x)-\mu\sum_{j=1}^{n}\ln x_j\right).$$

Overview of the Classical PD-IPM. We considered the problem used by [13] as stated in (2.1):

$$\min\left(f(x)-\mu\sum_{j=1}^{n}\ln x_j\right) \tag{2.1}$$

$$\text{subject to: } Ax-b=0.$$

The Lagrangian function of (2.1) is given as:

$$L(x,y,\mu)=c^{\mathrm{T}}x+\frac{1}{2}x^{\mathrm{T}}Qx-y^{\mathrm{T}}(Ax-b)-\mu\sum_{j=1}^{n}\ln x_j \tag{2.2}$$

where y and u are vectors of Lagrange multipliers and the Central Path parameter respectively. The first order necessary optimality conditions of (2.1) which are also sufficient for the problem are:

$$\nabla_x L(x,y,s)=Qx+c-A^{\mathrm{T}}y-\mu X^{-1}=0 \tag{2.3}$$

$$\nabla_y L(x,y,s)=Ax-b=0 \tag{2.4}$$

where $X^{-1}=diag\left\{\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\cdots,\frac{1}{x_n}\right\}$. Letting $s=\mu X^{-1}e$, yields $Xse=\mu e$, where e is a unit vector. The first order optimality conditions (2.3) and (2.4) can be written as:

$$F_0(x,y,s)=0=\begin{bmatrix} Ax-b \\ A^{\mathrm{T}}y+s-Qx-c \\ Xse-\mu e \end{bmatrix} \tag{2.5}$$

With further manipulations, (2.5) can be written as (2.6):

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^{\mathrm{T}} & I \\ S & 0 & X \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix}=\begin{bmatrix} \mathcal{E}_{\mathrm{p}} \\ \mathcal{E}_{\mathrm{d}} \\ \mathcal{E}_{\mathrm{u}} \end{bmatrix}=\begin{bmatrix} b-Ax \\ c-A^{\mathrm{T}}y-s+Qx \\ \sigma\mu e-Xse \end{bmatrix} \tag{2.6}$$

where $S=diag(s_1,s_2,\cdots,s_n)$.

The set of all values of μ which are positive traces a continuous central path which is called the primal-dual central trajectory. The path $\{x(\mu),y(\mu),s(\mu):\mu>0\}$ is usually a curve.

We seek to linearize the central paths of convex quadratic optimization problems, since linear paths are expected to be shorter than curvy paths to the solution.

2.1. Linearization of the Central Path

The general format of a convex quadratic optimization problem is:

$$\text{Minimize } f_0(x)$$

$$\text{subject to } g_i(x)=0,\ i=1,2,\cdots,m,$$

$$h_j(x)\le 0,\ j=1,2,\cdots,p,$$

where $f_0(x)$ is a convex quadratic objective function, $g_i(x)$ is linear (or quadratic) equality constraints and $h_j(x)$ is linear (or quadratic) inequality constraints.

Interior-point methods approximate the nonlinear central via Newton linearization, and under suitable assumptions, the resulting algorithm converges to the optimal solution in polynomial time.

The proposed linearized path is derived from the derivative of the Lagrange function with respect to the decision variables.

Consider the convex quadratic optimization problem:

$$\min f(x) = \frac{1}{2}x^{\mathrm{T}}Qx + c^{\mathrm{T}}x + q,$$

$$\text{subject to } \; Ax = b. \;\; x \geq 0,$$

where Q is symmetric positive semidefinite and A has a full row rank.

Assumptions

We assume that:

1) The feasible region is nonempty.

2) Q is symmetric positive definite.

3) Slater's condition holds

4) The KKT point exists and is bounded

The linearized central path (LCP) derived from the gradient of the Lagrangian function satisfies the first optimality conditions when the barrier parameter tends to zero.

The classical central path satisfies:

$$Qx + c - A^{\mathrm{T}}y - s = 0$$

$$Ax - b = 0$$

$$XSe = \mu e$$

The proposed linearized relations are obtained from the combinations of gradient components:

$$f_i(x)g_j(x) - g_i(x)f_j(x) = 0$$

These relations represent first-order linear combinations of the stationarity conditions.

As $\mu \to 0$, the complementarity condition implies that:

$$x_i s_i \to 0.$$

And the nonlinear central trajectory approaches the KKT system. Since the LCP is constructed from the same gradient structure, its limit satisfies the KKT conditions.

Hence, the LCP is a first-order consistent approximation of the classical central path.

Case 1. We consider the case where the problem is purely equality constrained as:

$$\min\left(f(x) - \mu\sum_{j=1}^{n}\ln x_j\right) \tag{2.7}$$

$$\text{subject to } \; g_i(x) = 0, \;\; \forall i = 1,2,\cdots,m; \;\; j = 1,2,\cdots,n.$$

The Lagrangian function of (2.7) is given as:

$$L(x,y,u) = f(x) - \sum_{i=1}^{m}\lambda_i g_i(x) - \mu\sum_{j=1}^{n}\ln x_j \tag{2.8}$$

The first order necessary optimality conditions (in scalar form) of (2.8) are stated below:

$$\frac{\partial f(x)}{\partial x_j}-\sum_{i=1}^{m}\lambda_i\frac{\partial g_i(x)}{\partial x_j}-\frac{\mu}{x_j}=0,\quad i=1,2,\cdots,m;\ j=1,2,\cdots,n \tag{2.9}$$

$$\frac{\partial f(x)}{\partial \lambda_i}-g_i(x)=0,\quad i=1,2,\cdots,m. \tag{2.10}$$

It is noted that the linearization of the central path of problem (2.7) depends only on (2.9) which involves n equations in n unknown variables of x and m unknown variables of λ. Therefore, if we are able to eliminate $\lambda_i\left(i=1,2,\cdots,m\right)$ from (2.9), then we can solve (2.9) independently of (2.10) to obtain u (the linearized central path). Secondly, we observe, on the other hand, that (2.10) cannot be solved independently of (2.9), since it contains no u in it. With the observations made, therefore, we proceed to eliminate λ_i ($i\in\{1,2,\cdots,m\}$) by some means in (2.9), so that we can obtain equations involving only the n unknown variables of x. An expanded version of (2.9) is given by the Equations (2.11).

$$\begin{aligned}&\frac{\partial f(x)}{\partial x_j}-\lambda_1\frac{\partial g_1(x)}{\partial x_j}-\frac{\mu}{x_j}-\lambda_2\frac{\partial g_2(x)}{\partial x_j}-\frac{\mu}{x_j}-\cdots-\lambda_{m-1}\frac{\partial g_{m-1}(x)}{\partial x_j}\\&-\frac{\mu}{x_j}-\lambda_m\frac{\partial g_m(x)}{\partial x_j}-\frac{\mu}{x_j}=0,\quad j=1,2,\cdots,n.\end{aligned} \tag{2.11}$$

As was noted earlier, $g_i(x)$ and $\frac{\partial g_1(x)}{\partial x_j}$ both vanish at x^* for any i and j, in spite of λ_i [14]. Therefore, λ_i can be viewed as arbitrary for all i, as far as finding u from (2.9) and therefore from (2.11) is concerned. This means we can choose λ_i arbitrarily in our quest to solve (2.9) independently of (2.10). By choosing $\lambda_i\neq 0$ for some $i\in\{1,2,\cdots,m\}$ and $\lambda_s=0$ for all $s\neq i$, $s\in\{1,2,\cdots,m\}$, we obtain the result in (2.12):

$$\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}=\lambda_i\frac{\partial g_i(x)}{\partial x_j},\quad j=1,2,\cdots,n;\forall i\in\{1,2,\cdots,m\} \tag{2.12}$$

From (2.12), we can eliminate λ_i, by taking ratios of the j^{th} and the $(j+1)^{\text{th}}$ equations ($j=1,2,\cdots,n-1$), leading to:

$$\frac{\frac{\partial f(x)}{\partial x_1}-\frac{u}{x_1}}{\frac{\partial f(x)}{\partial x_2}-\frac{u}{x_2}}=\frac{\frac{\partial g_i(x)}{\partial x_1}}{\frac{\partial g_i(x)}{\partial x_2}},\frac{\frac{\partial f(x)}{\partial x_2}-\frac{u}{x_2}}{\frac{\partial f(x)}{\partial x_3}-\frac{u}{x_3}}=\frac{\frac{\partial g_i(x)}{\partial x_2}}{\frac{\partial g_i(x)}{\partial x_3}},\cdots,\frac{\frac{\partial f(x)}{\partial x_{n-1}}-\frac{u}{x_{n-1}}}{\frac{\partial f(x)}{\partial x_n}-\frac{u}{x_n}}=\frac{\frac{\partial g_i(x)}{\partial x_{n-1}}}{\frac{\partial g_i(x)}{\partial x_n}}$$

which is generalized as (2.13):

$$\frac{\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}}{\frac{\partial f(x)}{\partial x_{j+1}}-\frac{\mu}{x_{j+1}}}=\frac{\frac{\partial g_i(x)}{\partial x_j}}{\frac{\partial g_i(x)}{\partial x_{j+1}}},\quad j=1,2,\cdots,n-1;\ \forall i\in\{1,2,\cdots,m\} \tag{2.13}$$

The result in (2.13) leads to (2.14):

$$\frac{\partial f(x)}{\partial x_j}\frac{\partial g_i(x)}{\partial x_{j+1}}-\frac{\partial f(x)}{\partial x_{j+1}}\frac{\partial g_i(x)}{\partial x_j} = \mu\left[\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+1}}-\frac{1}{x_{j+1}}\frac{\partial g_i(x)}{\partial x_j}\right],\quad j=1,2,\cdots,n-1;\forall i\in\{1,2,\cdots,m\} \tag{2.14}$$

Comparing (2.6) of [14] and (2.14), we have:

$\mu\left[\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+1}}-\frac{1}{x_{j+1}}\frac{\partial g_i(x)}{\partial x_j}\right]=0$. Since $\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+1}}-\frac{1}{x_{j+1}}\frac{\partial g_i(x)}{\partial x_j}\neq 0$, it implies $\mu=0$.

We obtain:

$$\mu=\frac{\partial f(x)}{\partial x_j}\frac{\partial g_i(x)}{\partial x_{j+1}}-\frac{\partial f(x)}{\partial x_{j+1}}\frac{\partial g_i(x)}{\partial x_j} \tag{2.15}$$

The result in (2.15) produces for each i, $n-1$ linear equations, which we refer to as *Linearized Central Paths* (LCP) (See [14]) and very much equivalent to Subsidiary Constraint Equations (SCE) encountered under modification of the Lagrange method (See [14]). From numerical experimental work, as was observed with SCE under MLM, the $n-1$ LCP produced from (2.15) occasionally yield amongst the set of LCP some redundant ones, and depending on the number of redundant ones appearing in the set, there may not be sufficient number of variables in the LCP for finding x^*. This is dealt with in the same way as was done for SCE, by taking ratios in arbitrary order, instead of consecutively, indicating that for LCP too (as was observed about SCE), a very large number of possible forms of the LCP can be constructed (See [14]).

For the purpose of this work, an ordering of the ratios leading to other forms of LCP that seem to avoid producing redundant equations are given by (2.16):

$$\frac{\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}}{\frac{\partial f(x)}{\partial x_{j+1}}-\frac{u}{x_{j+1}}}=\frac{\frac{\partial g_i(x)}{\partial x_j}}{\frac{\partial g_i(x)}{\partial x_{j+1}}},\frac{\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}}{\frac{\partial f(x)}{\partial x_{j+2}}-\frac{u}{x_{j+2}}}=\frac{\frac{\partial g_i(x)}{\partial x_j}}{\frac{\partial g_i(x)}{\partial x_{j+2}}},\cdots,\frac{\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}}{\frac{\partial f(x)}{\partial x_n}-\frac{u}{x_n}}=\frac{\frac{\partial g_i(x)}{\partial x_j}}{\frac{\partial g_i(x)}{\partial x_n}} \tag{2.16}$$

where $j\in\{1,2,\cdots,n-1\}$ and $i\in\{1,2,\cdots,m\}$. The result in (2.16) is generalized as (2.17):

$$\frac{\frac{\partial f(x)}{\partial x_j}-\frac{\mu}{x_j}}{\frac{\partial f(x)}{\partial x_{j+s}}-\frac{\mu}{x_{j+s}}}=\frac{\frac{\partial g_i(x)}{\partial x_j}}{\frac{\partial g_i(x)}{\partial x_{j+s}}},\quad j\in\{1,2,\cdots,n-1\},s=1,2,\cdots,n-1;\ \forall i \tag{2.17}$$

It follows from (2.17) that:

$$\frac{\partial f(x)}{\partial x_j}\frac{\partial g_i(x)}{\partial x_{j+s}}-\frac{\partial f(x)}{\partial x_{j+s}}\frac{\partial g_i(x)}{\partial x_j} = \mu\left[\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+s}}-\frac{1}{x_{j+s}}\frac{\partial g_i(x)}{\partial x_j}\right],\quad j\in\{1,2,\cdots,n-1\},s=1,2,\cdots,n-1;\ \forall i \tag{2.18}$$

Comparing (2.9) of [14] and (2.18), we have:

$\mu\left[\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+s}}-\frac{1}{x_{j+s}}\frac{\partial g_i(x)}{\partial x_j}\right]=0$. Since $\frac{1}{x_j}\frac{\partial g_i(x)}{\partial x_{j+s}}-\frac{1}{x_{j+s}}\frac{\partial g_i(x)}{\partial x_j}\neq 0$, it implies $\mu=0$.

We have:

$$\mu=\frac{\partial f(x)}{\partial x_j}\frac{\partial g_i(x)}{\partial x_{j+s}}-\frac{\partial f(x)}{\partial x_{j+s}}\frac{\partial g_i(x)}{\partial x_j} \tag{2.19}$$

Similar to SCE generation, taking aggregates of the set of equations under the LCP obtained from (2.19) appear to be more effective at avoiding the redundancy phenomenon (as observed from numerical experimentation). Therefore, summing from the k^{th} LCP of (2.19), ($k=j\geq 1$), involving the equations $k,k+1,\cdots,n-1,n$, we obtain the following linearized central paths:

$$\mu=\frac{1}{n-1}\left[\frac{\partial f(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial g_i(x)}{\partial x_s}-\frac{\partial g_i(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial f(x)}{\partial x_s}\right],\ 1\leq k\leq n-2 \tag{2.20}$$

$$\mu=\frac{1}{2}\left[\left(\sum_{s=k-1}^{k}\frac{\partial f(x)}{\partial x_s}\right)\frac{\partial g_i(x)}{\partial x_{s=k+1}}-\left(\sum_{s=k-1}^{k}\frac{\partial g_i(x)}{\partial x_s}\right)\frac{\partial f(x)}{\partial x_{s=k+1}}\right],\ k=n-1 \tag{2.21}$$

$$\begin{aligned}\mu=\frac{1}{2}\Bigg[&\frac{1}{n-1}\left(\frac{\partial f(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial g_i(x)}{\partial x_s}-\frac{\partial g_i(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial f(x)}{\partial x_s}\right)\\&+\frac{1}{2}\left(\left(\sum_{s=k-1}^{k}\frac{\partial f(x)}{\partial x_s}\right)\frac{\partial g_i(x)}{\partial x_{s=k+1}}-\left(\sum_{s=k-1}^{k}\frac{\partial g_i(x)}{\partial x_s}\right)\frac{\partial f(x)}{\partial x_{s=k+1}}\right)\Bigg]\end{aligned} \tag{2.22}$$

It is noted that two (2) or more linearized central paths, may be obtained from (2.20) or (2.21), indicating further the variety of the forms of aggregates of LCP that may be created from (2.22).

2.2. Initialization of the Modified Primal-Dual IPM

This section describes the initialization of the MPD-IPM for a convex quadratic optimization problem with equality constraints as in (2.7). For a classical interior point method, the algorithm is initialized as follows:

Counter for the iteration, $k=0$;

$(x^0,y^0,s^0)\in N_S(\gamma,\beta)$, where (x^0,y^0,s^0) is the primal-dual point;

$$\mu^0=\frac{(x^0)^{\mathrm{T}}s^0}{n},\text{ where } \mu^0 \text{ is the barrier parameter} \tag{2.23}$$

From (2.20), (2.21) and (2.22), it is observed that the barrier parameter u, partly varies with the decision variables of the subsidiary constrained equations and so the new algorithm is initialized by extracting the absolute values of the coefficients of the decision variables as the starting point for the algorithm.

For example, from (2.22), the initial value of the algorithm is given by;

$$\begin{aligned}x^0=\frac{1}{2}\Bigg[&\frac{1}{n-1}\left(\frac{\partial f(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial g_i(x)}{\partial x_s}-\frac{\partial g_i(x)}{\partial x_k}\sum_{s=k+1}^{n}\frac{\partial f(x)}{\partial x_s}\right)\\&+\frac{1}{2}\left(\left(\sum_{s=k-1}^{k}\frac{\partial f(x)}{\partial x_s}\right)\frac{\partial g_i(x)}{\partial x_{s=k+1}}-\left(\sum_{s=k-1}^{k}\frac{\partial g_i(x)}{\partial x_s}\right)\frac{\partial f(x)}{\partial x_{s=k+1}}\right)\Bigg]\end{aligned} \tag{2.24}$$

s^0 is a vector of ones and y^0 is free.

The proposed algorithm only modifies μ; that is (2.23) and the rest of the algorithm takes the shape of the classical PD-IPM.

Under Assumptions 1 - 4, the sequence x^k generated by the MPD-IPM converges to KKT point of the convex quadratic program due to the following reasons:

Sketch of Argument

1) The algorithm uses the classical PD-IPM, which is globally convergent for convex problems.

2) The only **modification** is the initialization vector derived from the LCP coefficients.

3) Since the initialization remains strictly feasible (by construction of absolute coefficients), the iterates remain in the interior region.

4) Convexity ensures that any limit point satisfying stationarity is globally optimal.

Therefore, the MPD-IPM preserves the global convergence properties of the classical primal-dual interior point method.

Since the MPD-IPM modifies only the initialization while maintaining the classical primal-dual update structure, the polynomial-time complexity of the underlying interior-point method is preserved.

Case 2. We consider the case where the problem has inequality constraints such as:

$$h_j(x) \leq 0$$

With linear (or quadratic) inequality constraints, active constraints are identified by using MATLAB codes. And so, problems involving case 2 are reduced to case 1.

The linearization of the central path in this case follows the same principles as used for modifying the Lagrangian method (See [14]). By identifying therefore the active constraints, problems of case 2 become equivalent to problem (2.7), and so the methodology for linearizing problem (2.7) is the same as that for linearizing the central path of problems involving case 2. The central paths derived under (2.7) are the same as the central paths of (2.25). For this reason, the central paths (2.20), (2.21) and (2.22) derived under (2.7) will also be used to produce results for inequality constrained problems.

It must be noted that the initialization of the PD-IPM for an equality constrained problem is the same as the initialization of the PD-IPM for case 2 since case 2 is reduced to case1 when active constraints are identified.

2.3. Initialization Strategy

The coefficients of the LCP are extracted and their absolute values are used as the starting vector:

$$x^{(0)} = |\text{coefficients of LCP}|$$

This replaces the classical initialization and places the initial iterate closer to the effective central region. See Appendix A for the implementation of the new algorithm.

3. Numerical Results

To evaluate the effectiveness and robustness of the proposed method, ten (10) convex quadratic optimizations were considered. Three (3) of these problems are small-scale benchmark instances adopted from Jian *et al.* (2017). Specifically, problems HS48, HS51 and HS52 from that study are presented in this paper as problems 1, 2, and 3 respectively. The remaining seven (7) problems were hypothetically generated to further examine the performance of the algorithm under controlled computational setting.

All computations were performed in MATLAB R2021a using the interior-point algorithm of fmincon.

Performance metrics included: Number of iterations, Computational time, Objective function values, function evaluations and convergence status.

3.1. Comparing the Performance of [15] with the MPD-IPM

Three (3) problems were selected to compare with the performance of [15] with the MPD-IPM. We compared the number of iterations, the objective function values, function evaluations and run time of both methods. The following keys are used in the **Table 1** below: IV = Initial Values, #Itr = Number of iterations, OFV = Objective function Value, Sol. = Solution, FE = Function evaluations, DNC = Do Not Converge and CONV. = Converge.

Table 1. Comparing [15] with that of MPD-IPM algorithm, for problem P1 - P3.

Problem	Method	IV	#Itr	Time (s)	OFV	Sol	FE
P1	[15] Method	See [15]	21	0.03	2.2808e−05	Same	55
	MPD-IPM	See **Table 2**	4	0.03	0.0000	Same	31
P2	[15] Method	See [15]	29	0.03	5.2930	Same	132
	MPD-IPM	See **Table 2**	6	0.03	5.3266	Same	50
P3	[15] Method	See [15]	31	0.06	4.0734	Same	48
	MPD-IPM	See **Table 2**	6	0.03	4.0930	Same	31

From **Table 1** compares the proposed MPD-IPM with the algorithm in [15]. The MPD-IPM required significantly fewer iterations while maintaining comparable objective values. Computational times were similar due to small problem sizes.

3.2. Hypothetical Problems

In general, for all the constructed problems, the convex quadratic objective function is given by;

$$\min f(x)_{x\in R^n} = x^{\mathrm{T}}Qx + c^{\mathrm{T}}x + q,$$

$$\text{subject to} \quad Ax \le b$$

$$A_{eq}x = b_{eq}.$$

where $x \in R^n$ denotes the decision variable, $Q \in R^{n\times n}$ is a symmetric positive semidefinite matrix defining the quadratic term, $c \in R^n$ is the linear coefficient vector, $q \in R$ is a scalar constant, $A \in R^{m\times n}$ and $A_{eq} \in R^{m\times n}$ are the constraint matrices and $b \in R^m$ and $b_{eq} \in R^m$ are the right-hand-side vectors.

Problem, P4 (*n* = 25)

This is a quadratic optimization problem constructed with 25 decision variables and 15 linear inequality constraints in the form:

$$\min f(x) = x^{\mathrm{T}}Qx + c^{\mathrm{T}}x + 41.0044,$$

$$\text{subject to} \quad Ax \le b.$$

The matrix $Q \in R^{25\times 25}$ has diagonal elements equal to $\frac{231}{2}$ and off-diagonal elements equal to $\frac{225}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{15\times 25}$ is a submatrix of size 15 × 25 extracted from the 25 × 25 identity matrix and $b_i = \sum_{j=1}^{25} A_{ij}$, $i = 1, 2, \cdots, 15$.

Constant, $A \in R^{m\times n}$ is the constraint matrix and $b \in R^m$ is the right-hand-side vector.

Problem, P5 (*n* = 40)

This is a quadratic optimization problem formulated with 40 decision variables and 3 linear inequality constraints in the form:

$$\min f(x) = x^{\mathrm{T}}Qx + c^{\mathrm{T}}x + 25,$$

$$\text{subject to} \quad Ax \le b.$$

The matrix $Q \in R^{40\times 40}$ has diagonal elements equal to $\frac{47}{2}$ and off-diagonal elements equal to $\frac{45}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{3\times 40}$ is a submatrix of size 3 × 40 extracted from the 40 × 40 identity matrix and $b_i = \sum_{j=1}^{40} A_{ij}$, $i = 1, 2, 3$.

Constant, $A \in R^{m\times n}$ is the constraint matrix and $b \in R^m$ is the right-hand-side vector.

Problem, P6 (*n* = 100)

This is a quadratic optimization problem constructed with 100 decision variables and 5 linear inequality constraints in the form:

$$\min f(x) = x^{\mathrm{T}}Qx + c^{\mathrm{T}}x + 2.5,$$

$$\text{subject to} \quad Ax \le b.$$

The matrix $Q \in R^{100\times 100}$ has diagonal elements equal to $\frac{517}{2}$ and off-diagonal elements equal to $\frac{415}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{5\times 100}$ is a submatrix of size 5 × 100 extracted from the 100 × 100 negative unit matrix and $b_i = \sum_{j=1}^{100} A_{ij}$, $i = 1, 2, \cdots, 5$.

Problem, P7 (*n* = 250)

This is a quadratic optimization problem constructed with 250 decision variables and 3 linear inequality constraints in the form:

$$\min f(x) = x^{\mathrm{T}} Q x + c^{\mathrm{T}} x + 12.5,$$

$$\text{subject to } Ax \le b.$$

The matrix $Q \in R^{250\times 250}$ has diagonal elements equal to $\frac{85}{2}$ and off-diagonal elements equal to $\frac{55}{2}$ and $c = e$, where e is a vector of ones.

Problem, P8 (*n* = 150)

This is a quadratic optimization problem formulated with 150 decision variables and 3 linear inequality constraints in the form:

$$\min f(x) = x^{\mathrm{T}} Q x + c^{\mathrm{T}} x + 11,$$

$$\text{subject to } Ax \le b.$$

The matrix $Q \in R^{150\times 150}$ has diagonal elements equal to $\frac{103}{2}$ and off-diagonal elements equal to $\frac{77}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{150\times 150}$ is a submatrix of size 3 × 150 extracted from the 25 × 25 Pascal coefficients matrix and $b_i = b_i = \sum_{j=1}^{150} A_{ij}$, $i = 1, 2, 3$.

Problem, P9 (*n* = 500)

This is a quadratic optimization problem formulated with 500 decision variables and 15 linear equality constraints in the form:

$$\min f(x) = x^{\mathrm{T}} Q x + c^{\mathrm{T}} x - 32,$$

$$\text{subject to } Ax = b.$$

The matrix $Q \in R^{500\times 500}$ has diagonal elements equal to $\frac{243}{2}$ and off-diagonal elements equal to $\frac{225}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{15\times 500}$ is a submatrix of size 15 × 500 extracted from the 500 × 500-unit matrix and $b_i = b_i = \sum_{j=1}^{500} A_{ij}$, $i = 1, 2, \cdots, 15$.

Problem, P10 (*n* = 70)

This is a quadratic optimization problem formulated with 500 decision variables and 5 linear inequality constraints and 3 linear equality constraints in the form:

$$\min f(x) = x^{\mathrm{T}} Q x + c^{\mathrm{T}} x - 32,$$

$$\text{subject to } Ax \le b,$$

$$Ax_{eq} = b_{eq}.$$

The matrix $Q \in R^{70\times70}$ has diagonal elements equal to $\frac{243}{2}$ and off-diagonal elements equal to $\frac{225}{2}$ and $c = e$, where e is a vector of ones.

The constraint matrix $A \in R^{3\times70}$ is a submatrix of size 3 × 70 extracted from the 70 × 70-unit matrix and $b_i = \sum_{j=1}^{70} A_{ij}$, $i = 1,2,3$.

Also, the constraint matrix $A_{eq} \in R^{5\times70}$ is a submatrix of size 5 × 70 extracted from the 70 × 70-unit matrix and $(b_{eq})_i = \sum_{j=1}^{70} (A_{eq})_{ij}$, $i = 1,2,\cdots,5$.

3.3. Initialization of the Proposed Algorithm

The algorithm was initialized with the following decision variables presented in Table 2. All initial values satisfy the non-negativity constraints and provide a feasible starting point for the primal -dual interior point iterations.

Table 2. Shows the Initial Values for the various problems.

Problem	Initial Values for the MPD-IPM
P1	[1, 1, 1, 1, 1]
P2	[1, 0, 1, 0, 0]
P3	[1, 0, 1, 0, 0]
	Initial Values for the CPD-IPM and the MPD-IPM
P4	$x_i^{(0)} = \begin{cases} 1296.0, i = 1 \\ 1297.4, i = 2,3,\cdots,25 \end{cases}$
P5	$x_i^{(0)} = \begin{cases} 19.0125, i = 1 \\ 0.4875, i = 2,3,\cdots,37 \\ 0.7375, i = 38,39 \\ 0.0125, i = 40 \end{cases}$
P6	$x_i^{(0)} = \begin{cases} 539.0550, i = 1 \\ 5.4450, i = 2,3,\cdots,97 \\ 8.1950, i = 98,99 \\ 0.0550, i = 100 \end{cases}$
P7	$x_i^{(0)} = \begin{cases} 0.0007, i = 1,2,\cdots,247 \\ 0.0293, i = 248,249 \\ 0.0610, i = 250 \end{cases}$
P8	$x_i^{(0)} = \begin{cases} 962.0433, i = 1 \\ 6.4567, i = 2,3,\cdots,146 \\ 9.7067, i = 148,149 \\ 0.0433, i = 150 \end{cases}$
P9	$x_i^{(0)} = \begin{cases} 60.6285, i = 1 \\ 56.1375, i = 2,3,\cdots,500 \end{cases}$
P10	$x_i^{(0)} = \begin{cases} 59.8821, i = 1 \\ 55.4464, i = 2,3,\cdots,70 \end{cases}$

3.4. Comparing the Performance of the CIPM with the MPD-IPM

In this section, the performance of the modified primal-dual interior method is compared with that of the classical interior point method to evaluate its performance, which is presented in Table 3 below.

Table 3. Comparing CIPM with that of MPD-IPM, for problem P4 - P10.

Problem	Method	IV	#Itr	Time (s)	OFV	Sol.	FE
P4	CPD-IPM	Same	20	0.1782	41.0000	Same	549
	MPD-IPM	Same	2	0.0480	41.000	Same	93
P5	CPD-IPM	Same	4	0.0771	24.9778	Same	292
	MPD-IPM	Same	4	0.0606	24.9778	Same	209
P6	CPD-IPM	Same	23	0.4099	1.0413	DNC	2413
	MPD-IPM	Same	4	0.0691	2.4980	CONV.	509
P7	CPD-IPM	Same	20	0.3677	9.0913	DNC	2415
	MPD-IPM	Same	14	0.1174	12.9819	CONV.	1771
P8	CPD-IPM	Same	19	0.3419	9.0913	DNC	3042
	MPD-IPM	Same	14	0.2174	10.9820	CONV.	2280
P9	CPD-IPM	Same	5	5.4533	−32.0044	DNC	3016
	MPD-IPM	Same	2	0.0193	37.9680	CONV.	1003
P10	CPD-IPM	Same	23	0.4417	−7.7697	Same	1730
	MPD-IPM	Same	3	0.0976	−7.7697	Same	287

Table 3 compares MPD-IPM with the classical PD-IPM. The key findings from these comparisons are: the MPD-IPM produced significant iterations, faster computational times, successful convergence in problems where CPD-IPM failed and improved robustness for high dimensional problems. For example, problem P9 (n = 500), CPM-IPM failed to converge, while MPD-IPM converged in 2 iterations.

3.5. Convergence Verification

At convergence, the final residual norms obtained are presented in Table 4. The algorithm terminates when the primal residual, $\left|r_p\right| \le 10^{-6}$, the dual residual, $\left|r_d\right| \le 10^{-6}$ and the complementarity residual, $\left|r_c\right| \le 10^{-6}$.

Table 4. Final residuals from the modified primal-dual interior point method.

Problem	Final Primal Residual, $\left\|r_p\right\|_2$	Final Dual Residual, $\left\|r_d\right\|_2$	Final Complementarity Residual, $\left\|r_c\right\|_2$
P4	3.8737e−06	6.8935e−06	0.0000e+00
P5	1.0913e−06	3.1577e−06	0.0000e+00
P6	5.9367e−08	2.0127e+00	0.0000e+00
P7	1.0000e−10	0.0000e+00	0.0000e+00
P8	5.7011e−06	0.0000e+00	0.0000e+00
P9	3.8730e−06	0.0000e+00	0.0000e+00
P10	1.7321e−08	0.0000e+00	0.0000e+00

From Table 4, the primal feasibility residuals are below the tolerance of 10^{-6}, confirming that the equality and inequality constraints are satisfied to high precision. Also, the dual feasibility residuals remain within the prescribed stopping tolerance of 10^{-6} and these demonstrate dual feasibility. The complementarity residuals converged to zero, indicating that central path conditions are satisfied and complementarity slackness holds.

Hence, the computed solutions satisfy the KKT system to acceptable numerical accuracy and are therefore considered optimal.

4. Conclusion

The modified primal-dual interior point method improves classical performance through central path linearization and structured initialization. Computational evidence demonstrates enhanced robustness, faster convergence, and better scalability. The approach provides a viable alternative for large-scale convex quadratic optimization problems. Also, the residual analysis verified that the final solutions satisfy the optimality conditions to the required numerical accuracy, highlighting the reliability of the approach.

Future Direction

Future research could explore applying the method to practical problems in finance, engineering, energy, or machine learning to demonstrate robustness and usefulness.

Funding

No external funding was received for this research.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Plaxco, J., Valdes, A. and Stojko, W. (2014) Interior-Point Method for LP-Optimization. CHE 345 Springs.

[2] Gill, P.E., Kungurtsev, V. and Robinson, D.P. (2016) A Stabilized SQP Method: Global Convergence. *IMA Journal of Numerical Analysis*, **37**, 407-443. https://doi.org/10.1093/imanum/drw004

[3] Han, Z. (2015) Primal-Dual Active-Set Methods for Convex Quadratic Optimization with Applications. Ph.D. Thesis, Lehigh University.

[4] Shustrova, A. (2015) Primal-Dual Interior Methods for Quadratic Programming. Ph.D. Thesis, University of California.

[5] Abdullahi, U.A., Suleiman, A. and Suleiman, A. (2017) Comparative Analysis of Simplex Method and the Family of Interior Point Method. *IOSR Journal of Mathematics*, **13**, 11-15.

[6] Boyd, S. and Vandenberghe, L. (2004) Convex Optimization. Cambridge University Press.

[7] Yan, Y. (2014) Active-Set Prediction in Quadratic Programming using Interior Point Methods and Controlled Perturbations. arXiv: 1409.5891.

[8] Pólik, I. and Terlaky, T. (2010) Interior Point Methods for Nonlinear Optimization. In: Di Pillo, G. and Schoen, F., Eds., *Nonlinear Optimization*, Springer, 215-276. https://doi.org/10.1007/978-3-642-11339-0_4

[9] Friedlander, M.P. and Orban, D. (2012) A Primal-Dual Regularized Interior-Point Method for Convex Quadratic Programs. *Mathematical Programming Computation*, **4**, 71-107. https://doi.org/10.1007/s12532-012-0035-2

[10] Qian, X., Liao, L., Sun, J. and Zhu, H. (2018) The Convergent Generalized Central Paths for Linearly Constrained Convex Programming. *SIAM Journal on Optimization*, **28**, 1183-1204. https://doi.org/10.1137/16m1104172

[11] Gondzio, J. (2019) Interior Point Methods for Large Scale Optimization. In: Book of Abstracts, 11.

[12] Pougkakiotis, S. and Gondzio, J. (2020) An Interior Point-Proximal Method of Multipliers for Convex Quadratic Programming. *Computational Optimization and Applications*, **78**, 307-351. https://doi.org/10.1007/s10589-020-00240-9

[13] Gondzio, J. (2013) Convergence Analysis of an Inexact Feasible Interior Point Method for Convex Quadratic Programming. *SIAM Journal on Optimization*, **23**, 1510-1527. https://doi.org/10.1137/120886017

[14] Twum, B., Christian, J.S. and John, A.E. (2024) A Modified Lagrange Method for Solving Convex Quadratic Optimization Problems. *Open Journal of Optimization*, **13**, 1-20.

[15] Jian, J., Zeng, H., Ma, G. and Zhu, Z. (2017) Primal-Dual Interior Point QP-Free Algorithm for Nonlinear Constrained Optimization. *Journal of Inequalities and Applications*, **2017**, Article No. 239. https://doi.org/10.1186/s13660-017-1500-2

Appendix A. Shows the MPD-IPM Algorithm

```
  n   =   ?                          %    Input the number of number decision variables
x=sym('x',[1 n])
var=(x);
i=1:n;
fun=@(x)                             (% Input the objective function)
f =   ?                              (% input the objective function)
x0= []];                             % input the initial values
H=eye(n);
Aeq=[];
beq=[];
A = [];
b=[];
lb=[]
ub=[]
nonlcon=[]
options = optimoptions(@fmincon,'Algorithm','interior-point','Display','off')
[~, ~, ~, ~, lambda] = fmincon(fun, x0, A, b, Aeq,beq,lb,ub,nonlcon,options)
q=lambda.eqlin
t=nnz(q);
if t~=0
l=A(t,:)
else
l=A(1,:)
end
syms(sym('x',[1 n]))
g=l*x'
[gx]=gradient(g,x)
[fx]=gradient(f,x)
K=n-2:n-1
ii=2:n
eqn1=(1/n-1)*(fx(1)*sum(gx(ii))-gx(1)*sum(fx(ii)));
(2.20)                        % Subsidiary Constraint Equation
 [cxy txy]=coeffs(eqn1, var)
zvar=setdiff(var,txy)
xx=symvar(eqn1)
 Coef_matrix=zeros(numel(eqn1),numel(xx));
     j=1
     [v1,v2]=coeffs(eqn1(j),xx) ;
     if isequal(size(v1),size(xx));
        c1=double([0]);
else
    disp('the last coefficient of v1 is zero')
```

```
            c1=double([v1(end)])
         end
  if isequal(size(txy),size(xx))
xvar=setdiff(var,txy);
h1=[cxy, zeros(1,length(zvar))]
z1=double(h1)
  else
         s1=cxy(1:end-1)
       h1=[s1, zeros(1,length(zvar))]
       z1=double(h1)
  end

c=abs(z1);
d1=[c]
options = optimoptions(@fmincon,'Algorithm','interior-point','Display','off');
tStart=tic
[x,fval,exitflag,output,lambda]=fmincon(fun,[d1],[],[],[],[],[],[],[],op-
tions)                    % d1 initial value
a1=x;
m1=fval;
p1=[output.iterations];
e1=exitflag
tEnd=toc(tStart);
r1=tEnd;
eqn2=0.5*((gx(n)*sum(fx(K))-fx(n)*sum(gx(K))));
(2.21)                    %      Subsidiary Constraint Equation
[dxy fxy]=coeffs(eqn2, var)
xvar=setdiff(var,fxy)
xx=symvar(eqn2)
  Coef_matrix=zeros(numel(eqn2),numel(xx))
         j=1
         [u1,u2]=coeffs(eqn2(j),xx)

         if isequal(size(u1),size(xx))
            c2=double([0])
else
       disp('the last coefficient of v1 is zero')
            c2=double([u1(end)])
         end

  if isequal(size(fxy),size(xx))
xvar=setdiff(var,fxy)
h2=[dxy, zeros(1,length(xvar))]
```

```
z2=double(h2)
  else
        s2=dxy(1:end-1)
      h2=[s2, zeros(1,length(xvar))]
      z2=double(h2)
  end
c=abs(z2);
d2=[c]
options = optimoptions(@fmincon,'Algorithm','interior-point','Display','off');
tStart=tic;
[x,fval,exitflag,output,lambda]=fmincon(fun,[d2],[],[],[],[],[],[],[],options)
                                      %    d2 initial    Value
a2=x;
m2=fval;
p2=[output.iterations];
e2=exitflag
tEnd=toc(tStart);
r2=tEnd;
eqn3=0.5*(eqn1+eqn2);                                        %     (2.22)
[bxy qxy]=coeffs(eqn3, var)
yvar=setdiff(var,qxy)
xx=symvar(eqn3)
  Coef_matrix=zeros(numel(eqn3),numel(xx))
        j=1
        [w1,w2]=coeffs(eqn3(j),xx)

      if isequal(size(w1),size(xx))
            c3=double([0])
else
      disp('the last coefficient of w1 is zero')
            c3=double([w1(end)])
        end
  if isequal(size(qxy),size(xx))
yvar=setdiff(var,qxy)
h3=[bxy, zeros(1,length(yvar))]
z3=double(h3)
  else
        s3=bxy(1:end-1)
      h3=[s3, zeros(1,length(yvar))]
      z3=double(h3)
  end

c=abs(z3)
```

```
d3=[c]
options = optimoptions(@fmincon,'Algorithm','interior-point','Display','off');
tStart=tic
[x,fval,exitflag,output,lambda]=fmincon(fun,[d3],[],[],[],[],[],[],[],options)
                    % d3 Initial Value
a3=x;
m3=fval;
p3=[output.iterations];
e3=exitflag
tEnd=toc(tStart);
r3=tEnd;
A=[e1,e2,e3];
[minA,maxA]=bounds(A)
y1=([m1;p1;r1]);
y2=([m2;p2;r2]);
y3=([m3;p3;r3]);
if maxA == e1 && m1 < m2 && m1 < m3
     disp(y1);
elseif maxA == e2 && m2 < m1 && m2 < m3
     disp(y2);
else
     disp(y3);
end
y1=([a1]);
y2=([a2]);
y3=([a3]);
if maxA ==e1 && m1 < m2 && m1 < m3
     disp(y1);
elseif maxA==e2 && m2 < m1 && m2 < m3
     disp(y2);
else
     disp(y3);
```