Infeasible primal–dual interior point methods based on the kernel function for convex QCQPs

Mohamed Selamat[1], **Mounia Laouar**[2,*], **Mahmoud Brahimi**[3]

[1,2,3] Laboratory of Partial Differential Equations, University of Batna 2, Batna 05000, Algeria

[1] email: mohamed.selamat@univ-batna2.dz
[2] email: m.laouar@univ-batna2.dz
[3] email: m.brahimi@univ-batna2.dz

Abstract

We study convex quadratically constrained quadratic programming (QCQP) problems through the lens of primal–dual interior-point methods based on kernel functions. In contrast to standard feasible interior-point approaches, we develop an infeasible method capable of starting from an arbitrary initial point that does not necessarily satisfy the primal or dual constraints. Under standard convexity assumptions and the existence of optimal solutions, the resulting convex QCQP enjoys strong duality, and its optimal solutions are fully characterized by the Karush–Kuhn–Tucker (KKT) conditions. We introduce a kernel-function-based barrier framework that replaces the classical logarithmic barrier, leading to a parametrized perturbed KKT system with explicit primal and dual residuals. This system defines an infeasible central path, whose neighborhood is followed using exact Newton directions derived from the chosen kernel function. We demonstrate that this approach provides a flexible and unified framework for designing and analyzing efficient Newton-based algorithms for QCQP, with potential extensions to broader classes of conic optimization problems.

Keywords: Convex QCQP, infeasible interior-point method, kernel function, primal–dual Newton method, nonlinear optimization.

MSC 2010 subject classifications 90C20, 90C22, 90C30, 90C51.

1 Introduction

Mathematical optimization plays a fundamental role in many scientific and industrial domains, ranging from financial portfolio management to engineering system design. Among the different classes of optimization problems, Quadratically Constrained Quadratic Programs (QCQP) constitute a particularly important family that generalizes both linear programming and quadratic programming.

The history of numerical resolution of convex optimization problems dates back to the pioneering work of Legendre [13] and Gauss [8] in the early 19th century on the least squares method, considered the first algorithmically solved convex problem. Later, Cauchy [2] proposed in 1847 the first gradient descent method, paving the way for modern iterative approaches. The general theoretical framework was established by Karush [10] in 1939 and formalized by Kuhn and Tucker [11] in 1951 with the famous KKT optimality conditions.

In the field of solution methods,interior-point methods (IPM) have occupied a central place since the pioneering work of Frisch [7] in the 1950s, who introduced the logarithmic barrier function. Carroll [1] developed practical applications in 1961, while Fiacco and McCormick [6] formalized the complete theory of barrier and penalty methods (SUMT) in 1968. However, it was the publication of Karmarkar's [9] revolutionary algorithm in 1984 that renewed interest in these methods by demonstrating their superior efficiency over the simplex method for certain large-scale problems.

Modern interior-point methods rely on the use of kernel functions that maintain iterates inside the feasible domain by penalizing approach to the boundaries. These methods present several advantages: polynomial convergence for convex problems, efficiency on large-scale problems, and ability to exploit sparse matrix structure. However, classical Fiacco-McCormick approaches [6] require a strictly feasible initial point, which can be as difficult to obtain as solving the original problem itself.

This limitation has motivated the development of **infeasible start methods** that allow the algorithm to begin with a point violating the constraints. These modern primal-dual methods, particularly filter approaches and regularization techniques, have considerably improved the robustness and practical applicability of interior-point methods.

In this work, we consider a Quadratically Constrained Quadratic Program (QCQP) under convexity assumptions. Problem (P1) is formulated as follows:

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m, \end{cases} \tag{P1}$$

where $f(x) := \frac{1}{2}x^\top Q_0 x + c_0^\top x + d_0$ is the convex quadratic objective function (with $Q_0 \succeq 0$) and the constraints $g_i(x) := \frac{1}{2}x^\top Q_i x + c_i^\top x + d_i$ are also quadratic and convex (with $Q_i \succeq 0$ for $i = 1, \ldots, m$). The matrices $Q_i \in \mathbb{R}^{n\times n}$ are symmetric.

The main objective of this work is to develop and analyze an infeasible-start interior-point method for solving problem (P1). More precisely, we propose to use a kernel function (logarithmic barrier) in the context where both the objective function and all constraints are quadratic and convex, while allowing an initial point that does not necessarily satisfy the constraints. This approach combines the advantages of problem convexity with the flexibility of infeasible-start methods, thus eliminating the need to find a strictly feasible point before starting the algorithm.

The convex and quadratic structure of problem (P1) allows us to fully exploit the theoretical properties of self-concordant barrier functions while developing an efficient primal-dual algorithm. We will establish the convergence conditions of our approach and demonstrate its practical efficiency on representative examples.

The remainder of this document is organized as follows: Section 2 presents the solution of convex QCQPs. Section 3 describes the theoretical properties of the new kernel. Section 4 analyzes the theoretical complexity. Section 5 presents numerical results and performance analysis. finally, Section 6 concludes this work.

2 Solving convex QCQP

This section presents the convexity and regularity conditions that guarantee strong duality and the validity of the KKT conditions for QCQP. We then introduce the Lagrangian formulation and the associated dual problem. Based on this, an infeasible primal–dual interior-point scheme grounded in kernel functions is developed. Finally, we describe the general algorithmic framework for following the central path and ensuring convergence to the optimal solution.

Regularity assumptions

In order to ensure that problem (P1) is convex and that strong duality holds, we impose the following assumptions:

(A1) The objective function $f(x)$ is convex, which holds if the matrix Q_0 is positive semidefinite:

$$Q_0 \succeq 0.$$

(A2) Each inequality constraint function $g_i(x)$ is convex, which requires that

$$Q_i \succeq 0, \quad i = 1, \dots, m.$$

(A3) (Slater condition) There exists a strictly feasible point $\bar{x} \in \mathbb{R}^n$ such that

$$g_i(\bar{x}) < 0, \quad i = 1, \dots, m.$$

Under (A1)–(A3), strong duality holds ($p^\star = d^\star$), and the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for optimality.

Under these assumptions, the QCQP becomes a convex optimization problem for which every local minimizer is a global minimizer.

Lagrangian and primal–dual formulation

The Lagrangian function associated with problem (P1) is defined as

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) = \tfrac{1}{2} x^\top Q_0 x + c_0^\top x + d_0 + \sum_{i=1}^{m} \lambda_i \left(\tfrac{1}{2} x^\top Q_i x + c_i^\top x + d_i \right),$$

where $\lambda \in \mathbb{R}^m$ (with $\lambda_i \geq 0$) are the Lagrange multipliers associated with the inequality constraints.

The dual function is defined as

$$\theta(\lambda) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda),$$

and the corresponding dual problem is

$$\begin{aligned} \max_{\lambda} \quad & \theta(\lambda) \\ \text{subject to} \quad & \lambda \geq 0. \end{aligned}$$

If assumptions (A1)–(A3) hold, then strong duality applies, i.e.,

$$p^\star = d^\star,$$

and any optimal primal–dual solution $(x^\star, \lambda^\star)$ of problem (P1) satisfies the Karush–Kuhn–Tucker (KKT) optimality conditions:

$$\begin{cases} g_i(x^\star) \leq 0, \quad i = 1, \dots, m, & \text{(Primal feasibility)} \\ \lambda_i^\star \geq 0, \quad i = 1, \dots, m, & \text{(Dual feasibility)} \\ \lambda_i^\star g_i(x^\star) = 0, \quad i = 1, \dots, m, & \text{(Complementary slackness)} \\ \left(Q_0 + \sum_{i=1}^m \lambda_i^\star Q_i\right) x^\star + \left(c_0 + \sum_{i=1}^m \lambda_i^\star c_i\right) = 0. & \text{(Stationarity)} \end{cases} \tag{P2}$$

These conditions are necessary and sufficient for optimality in the convex QCQP case.

2.1 An infeasible primal–dual interior-point approach

We now develop an infeasible primal–dual interior-point method (IPM) for solving the convex QCQP problem (P1). This approach does not require the starting point to satisfy the primal or dual constraints.

To apply the interior-point framework, we introduce slack variables $s \in \mathbb{R}^m$ to convert the inequality constraints into equality constraints coupled with nonnegativity constraints:

$$g_i(x) + s_i = 0, \quad s_i \geq 0, \quad i = 1, \ldots, m.$$

The full set of KKT conditions for problem (P1) is then equivalent to solving the system $\mathbf{F}(x, \lambda, s) = 0$, where

$$\mathbf{F}(x, \lambda, s) = \begin{pmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ g(x) + s \\ \Lambda s \end{pmatrix} = \begin{pmatrix} \mathbf{r}_d \\ \mathbf{r}_p \\ \mathbf{r}_{cs} \end{pmatrix} = 0,$$

with $\Lambda := \operatorname{diag}(\lambda)$, where $\mathbf{r}_d$ is the dual residual, $\mathbf{r}_p$ is the primal residual (associated with the equality form of the inequality constraints), and $\mathbf{r}_{cs}$ is the complementarity residual. At a strictly interior KKT point we have $\lambda > 0$ and $s > 0$.

2.2 Kernel function and infeasible central path

Let $\psi : (0, \infty) \to \mathbb{R}$ be a twice continuously differentiable kernel function satisfying:

(K1) ψ is strictly convex, i.e., $\psi''(t) > 0$ for all $t > 0$;

(K2) $\psi(1) = 0$ and $\psi'(1) = 0$;

(K3) $\lim_{t \to 0^+} \psi(t) = \lim_{t \to \infty} \psi(t) = +\infty$.

Such a function naturally induces a barrier for the inequality constraints $g_i(x) \leq 0$, $i = 1, \ldots, m$.

For a given barrier parameter $\mu > 0$ and contraction parameter $\theta \in (0, 1)$, the kernel-based infeasible primal–dual framework incorporates residuals from the initial iterate. Let $\mathbf{r}_d^0$ and $\mathbf{r}_p^0$ denote the initial dual and primal residuals, respectively, evaluated at the starting point (x^0, λ^0, s^0).

At iteration k, the infeasible central path is defined by the perturbed KKT system with slack variables $s \in \mathbb{R}^m$:

$$\begin{cases} \nabla_x \mathcal{L}(x, \lambda) = \mathbf{r}_d^0 (1 - \theta)^k, & \text{(Dual residual)} \\ g(x) + s = \mathbf{r}_p^0 (1 - \theta)^k, & \text{(Primal residual)} \\ \lambda_i s_i = \mu, \quad i = 1, \ldots, m, & \text{(Kernel-based centrality)} \end{cases} \tag{1}$$

where the centrality condition $\lambda_i s_i = \mu$ is derived from the kernel optimality condition.

Specifically, for strictly positive $\lambda_i > 0$ and $s_i > 0$, the barrier function

$$\Psi_\mu(x, \lambda, s) := \sum_{i=1}^{m} \psi\left(\frac{\lambda_i s_i}{\mu}\right)$$

yields the first-order optimality condition

$$\frac{\partial \Psi_\mu}{\partial \lambda_i} = \frac{s_i}{\mu} \psi'\left(\frac{\lambda_i s_i}{\mu}\right) = 0, \qquad i = 1, \ldots, m.$$

Since $s_i > 0$, this is equivalent to

$$\psi'\left(\frac{\lambda_i s_i}{\mu}\right) = 0, \qquad i = 1, \ldots, m.$$

By properties (K1)–(K2), the equation $\psi'(t) = 0$ has the unique solution $t = 1$, yielding the centrality condition

$$\lambda_i s_i = \mu, \qquad i = 1, \ldots, m.$$

Infeasible central path. Under assumptions (K1)–(K3) and standard regularity conditions, for each $\mu > 0$ and iteration k, system (1) admits a solution $(x^k(\mu), \lambda^k(\mu), s^k(\mu))$ that does *not* require initial feasibility. The set of such solutions defines the *infeasible central path*.

As $k \to \infty$ (with appropriately decreasing μ at each iteration), the residuals satisfy

$$\mathbf{r}_d^0(1-\theta)^k \to 0 \quad \text{and} \quad \mathbf{r}_p^0(1-\theta)^k \to 0,$$

and the sequence of iterates converges:

$$\lim_{k\to\infty} (x^k(\mu_k), \lambda^k(\mu_k)) = (x^\star, \lambda^\star),$$

where $(x^\star, \lambda^\star)$ satisfies the KKT system (P2) for problem (P1).

Relationship to the standard approach. When the initial point is feasible (i.e., $\mathbf{r}_d^0 = 0$ and $\mathbf{r}_p^0 = 0$), system (1) reduces to the standard feasible kernel-based central path:

$$\begin{cases} \nabla_x \mathcal{L}(x, \lambda) = 0, \\ g(x) + s = 0, \\ \lambda_i s_i = \mu, \quad i = 1, \ldots, m, \\ g_i(x) < 0,\ s_i > 0,\ \lambda_i > 0, \quad i = 1, \ldots, m. \end{cases} \tag{2}$$

2.3 Proximity measure and Newton direction

For the QCQP problem (P1), we introduce slack variables to convert the inequality constraints into equality form:

$$g_i(x) + s_i = 0, \quad s_i > 0, \quad i = 1, \ldots, m.$$

Along the feasible manifold, this implies

$$s_i = -g_i(x), \quad i = 1, \ldots, m.$$

At points satisfying strict feasibility ($g_i(x) < 0$), we have $s_i > 0$.

Let $s \in \mathbb{R}^m$ be the vector of slack variables, and $\lambda \in \mathbb{R}^m$ be the vector of Lagrange multipliers, both assumed to be strictly positive ($s > 0, \lambda > 0$). We define the scaled variables

$$v_i := \sqrt{\frac{\lambda_i s_i}{\mu}}, \quad i = 1, \ldots, m,$$

which measure the proximity of the i-th complementarity product to the central value μ. The *kernel-based proximity measure* is

$$\Psi(v) := \sum_{i=1}^{m} \psi(v_i),$$

and the *norm-based proximity measure* is defined based on the gradient of $\Psi(v)$:

$$\delta(v) := \frac{1}{2}\|\nabla\Psi(v)\| = \frac{1}{2}\sqrt{\sum_{i=1}^{m}\big(\psi'(v_i)\big)^2}.$$

The ℓ_2-norm of the vector $\mathbf{h}(v) = \big(\psi'(v_1), \ldots, \psi'(v_m)\big)^\top$ is a key metric for analyzing the complexity and convergence of kernel-based interior-point methods.

Newton direction. At each iteration k, we compute the Newton direction $(\Delta x, \Delta\lambda, \Delta s)$ by linearizing the perturbed infeasible KKT system (1).

The Hessian of the Lagrangian is

$$H(x,\lambda) = Q_0 + \sum_{i=1}^{m}\lambda_i Q_i,$$

which is symmetric and positive semidefinite under assumptions (A1)–(A2).

The gradient of the Lagrangian with respect to x is

$$\nabla_x\mathcal{L}(x,\lambda) = \left(Q_0 + \sum_{i=1}^{m}\lambda_i Q_i\right)x + \left(c_0 + \sum_{i=1}^{m}\lambda_i c_i\right) = H(x,\lambda)\,x + \left(c_0 + \sum_{i=1}^{m}\lambda_i c_i\right).$$

The Jacobian matrix of the constraint functions is

$$J(x) = \begin{bmatrix}\nabla g_1(x)^\top \\ \vdots \\ \nabla g_m(x)^\top\end{bmatrix} = \begin{bmatrix}(Q_1x + c_1)^\top \\ \vdots \\ (Q_mx + c_m)^\top\end{bmatrix} \in \mathbb{R}^{m\times n},$$

where $\nabla g_i(x) = Q_ix + c_i$ for each $i = 1, \ldots, m$.

The Newton system for the infeasible primal–dual method is:

$$\begin{bmatrix}H(x,\lambda) & J(x)^\top & 0 \\ J(x) & 0 & I \\ 0 & S & \Lambda\end{bmatrix}\begin{bmatrix}\Delta x \\ \Delta\lambda \\ \Delta s\end{bmatrix} = -\begin{bmatrix}\mathbf{r}_d^k \\ \mathbf{r}_p^k \\ \mathbf{r}_{cs}\end{bmatrix}, \tag{3}$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$, $S = \operatorname{diag}(s_1, \ldots, s_m)$, $I \in \mathbb{R}^{m\times m}$ is the identity matrix, and the residuals are:

- $\mathbf{r}_d^k = \nabla_x\mathcal{L}(x,\lambda) - \mathbf{r}_d^0(1-\theta)^k$ (dual residual),
- $\mathbf{r}_p^k = g(x) + s - \mathbf{r}_p^0(1-\theta)^k$ (primal residual),
- $\mathbf{r}_{cs} = \Lambda s - \mu\mathbb{1}$ (complementarity residual).

Here, $\mathbf{r}_d^0$ and $\mathbf{r}_p^0$ are the initial residuals evaluated at the starting point (x^0, λ^0, s^0), and $\theta \in (0,1)$ is the contraction parameter from (1).

Reduced system. To reduce the computational cost, system (3) can be simplified by eliminating Δs. From the third block-row:

$$S\Delta\lambda + \Lambda\Delta s = -\mathbf{r}_{cs} = -(\Lambda s - \mu\mathbb{1}) = \mu\mathbb{1} - \Lambda s.$$

Solving for Δs:

$$\Delta s = \Lambda^{-1}(\mu\mathbb{1} - \Lambda s - S\Delta\lambda) = \Lambda^{-1}(\mu\mathbb{1} - \Lambda s) - \Lambda^{-1}S\,\Delta\lambda.$$

Substituting into the second block-row ($J(x)\Delta x + \Delta s = -\mathbf{r}_p^k$):

$$J(x)\Delta x + \Lambda^{-1}(\mu \mathbb{1} - \Lambda s) - \Lambda^{-1} S\, \Delta\lambda = -\mathbf{r}_p^k.$$

Rearranging terms yields the following $(n+m)$-dimensional symmetric reduced system in $(\Delta x, \Delta\lambda)$:

$$\begin{bmatrix} H(x,\lambda) & J(x)^\top \\ J(x) & -\Lambda^{-1}S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_d^k \\ -\mathbf{r}_p^k - \Lambda^{-1}(\mu\mathbb{1} - \Lambda s) \end{bmatrix}. \tag{4}$$

Once Δx and $\Delta\lambda$ are computed from (4), the slack variable step Δs is recovered using:

$$\Delta s = \Lambda^{-1}(\mu \mathbb{1} - \Lambda s) - \Lambda^{-1} S\, \Delta\lambda.$$

The matrix $\Lambda^{-1}S = \operatorname{diag}(s_1/\lambda_1, \ldots, s_m/\lambda_m)$ is a positive diagonal matrix that acts as a scaling factor in the $(2,2)$-block of the reduced system. This scaling adapts to the current iterate and plays a crucial role in the convergence properties of the method.

2.4 Algorithmic framework

We now present the complete kernel-based infeasible primal–dual interior-point method. At each outer iteration, the barrier parameter μ is reduced by a factor $(1-\theta)$, and the inner loop applies full Newton steps to decrease the kernel-based proximity measure below a prescribed threshold τ.

Algorithm 1 Kernel-based infeasible primal–dual interior-point method for convex QCQP

Input: tolerance $\varepsilon > 0$; parameters $\theta \in (0,1)$, $\tau > 0$, $\eta \in (0,1)$
Choose an initial (possibly infeasible) point (x^0, λ^0, s^0) with $\lambda^0 > 0$ and $s^0 > 0$
Compute initial residuals

$$\mathbf{r}_d^0 = \nabla_x \mathcal{L}(x^0, \lambda^0), \qquad \mathbf{r}_p^0 = g(x^0) + s^0$$

Set $\mu^0 > 0$, $k \leftarrow 0$, and $(x, \lambda, s) \leftarrow (x^0, \lambda^0, s^0)$
while $\mu^k > \varepsilon$ **do**
    Set current target residuals and barrier parameter

$$\widehat{\mathbf{r}}_d^k = (1-\theta)^k \mathbf{r}_d^0, \qquad \widehat{\mathbf{r}}_p^k = (1-\theta)^k \mathbf{r}_p^0, \qquad \mu^{k+1} = \mu^0 (1-\theta)^k$$

    Compute scaled variables $v_i = \dfrac{\lambda_i s_i}{\mu}$ and the proximity measure $\delta(v)$
    while $\delta(v) > \tau$ **do** ▷ Inner centering loop
        Form the residuals for the Newton step

$$\mathbf{r}_d^k = \nabla_x \mathcal{L}(x, \lambda) - \widehat{\mathbf{r}}_d^k, \quad \mathbf{r}_p^k = g(x) + s - \widehat{\mathbf{r}}_p^k, \quad \mathbf{r}_{cs} = \Lambda s - \mu \mathbb{1}$$

        Compute the **exact Newton direction** $(\Delta x, \Delta\lambda, \Delta s)$ by solving (3) (or the reduced system (4))
        Compute maximal step lengths $\alpha_{\mathrm{p}}^{\max}, \alpha_{\mathrm{d}}^{\max} \in (0,1]$ such that

$$s + \alpha_{\mathrm{p}}^{\max} \Delta s > 0, \qquad \lambda + \alpha_{\mathrm{d}}^{\max} \Delta\lambda > 0$$

        Set $\alpha \leftarrow \eta \min\{\alpha_{\mathrm{p}}^{\max}, \alpha_{\mathrm{d}}^{\max}\}$
        Update the iterates:

$$x \leftarrow x + \alpha \Delta x, \quad \lambda \leftarrow \lambda + \alpha \Delta\lambda, \quad s \leftarrow s + \alpha \Delta s$$

        Update $v_i = \sqrt{\dfrac{\lambda_i s_i}{\mu}}$ and recompute $\delta(v)$
    end while
    $k \leftarrow k + 1$ ▷ Centering achieved for current μ; proceed to next outer iteration
end while
Output: approximate optimal solution $(x^\star, \lambda^\star, s^\star) = (x, \lambda, s)$

Remarks on the algorithm.

1. This algorithm is infeasible in the sense that the initial point does not need to satisfy the primal or dual constraints; feasibility and complementarity are recovered asymptotically along the kernel-based infeasible central path.

2. At each outer iteration k, the barrier parameter is updated as $\mu^{k+1} = \mu^0(1-\theta)^k$, and the target residuals decrease geometrically with rate $(1-\theta)$.

3. The inner loop ensures that the iterates remain in a neighborhood of the infeasible central path by maintaining $\delta(v) \le \tau$.

4. The step length parameter $\eta \in (0,1)$ ensures that the iterates remain strictly positive ($\lambda > 0$, $s > 0$). In practice, η is typically chosen close to 1 (e.g., $\eta = 0.99$).

3 Theoretical properties of the new kernel function

In this section, we introduce a new kernel function designed to satisfy all the classical conditions required by primal-dual interior-point methods. We then examine its fundamental analytical properties, which are essential for establishing the convergence and stability of the algorithm.

The kernel function is defined, for $t > 0$, by

$$\begin{aligned} \psi : \quad \mathbb{R}_+^* &\longrightarrow \mathbb{R} \\ t &\longmapsto \frac{m}{te^{t/p}} + \frac{\beta}{t} + \left(\frac{m(\frac{1}{p}+1)}{2e^{1/p}} + \frac{\beta}{2} \right) t^2 - \frac{m(\frac{1}{p}+3)}{2e^{1/p}} - \frac{3\beta}{2}. \end{aligned} \tag{5}$$

where $p > 1$, $m > 0$, and $\beta > 1$. We also define the constant

$$A := \psi''(1) = \frac{m\left(\frac{1}{p}+1\right)}{e^{1/p}} + \beta.$$

For $t > 0$, the derivatives of ψ are given by

$$\begin{cases} \psi'(t) = -\dfrac{m(\frac{t}{p}+1)}{t^2 e^{t/p}} - \dfrac{\beta}{t^2} + At, \\ \psi''(t) = \dfrac{m\left((\frac{t}{p})^2 + 2\frac{t}{p} + 2\right)}{t^3 e^{t/p}} + \dfrac{2\beta}{t^3} + A, \\ \psi'''(t) = -\dfrac{m\left((\frac{t}{p})^3 + 3(\frac{t}{p})^2 + 6\frac{t}{p} + 6\right)}{t^4 e^{t/p}} - \dfrac{6\beta}{t^4}. \end{cases} \tag{6}$$

Lemma 3.1. *The function ψ defined in (5) and with these derivatives defined in (6) satisfy:*

1. $\psi''(t) > 0$ *for all* $t > 0$;
2. $\psi(1) = 0$ *and* $\psi'(1) = 0$;
3. $\psi'''(t) < 0$ *for all* $t > 0$;
4. $t\psi''(t) + \psi'(t) > 0$ *for all* $t > 0$;
5. $t\psi''(t) - \psi'(t) > 0$ *for all* $t > 0$;
6. *For all* $t > 1$ *and* $\beta > 1$,

$$\psi''(t)\psi'(\beta t) - \beta\, \psi'(t)\psi''(\beta t) > 0.$$

Remark 3.1. *The function ψ satisfies the standard kernel-function conditions (K1)–(K3): strict convexity, normalization at $t = 1$, and divergence as $t \to 0^+$ or $t \to +\infty$.*

Definition 3.1. *For a vector $v \in \mathbb{R}^m_{++}$, the associated barrier function is*

$$\Psi(v) = \sum_{i=1}^{m} \psi(v_i).$$

The normalized proximity measure is defined as

$$\delta(v) = \frac{1}{2}\|\nabla\Psi(v)\| = \frac{1}{2}\sqrt{\sum_{i=1}^{m} \big(\psi'(v_i)\big)^2}.$$

Lemma 3.2 ([14]). *For a twice differentiable function $\psi : (0, \infty) \to \mathbb{R}$, the following statements are equivalent:*

1. *$\psi(e^{\zeta})$ is convex in ζ;*
2. *$\psi(\sqrt{\zeta})$ is convex in ζ;*
3. *$t\psi''(t) + \psi'(t) > 0$ for all $t > 0$;*
4. *$t\psi''(t) - \psi'(t) > 0$ for all $t > 0$.*

Lemma 3.3. *Since $\psi'''(t) < 0$ for all $t > 0$, the function ψ satisfies:*

Case $t < 1$:
$$\frac{\psi''(1)}{2}(t-1)^2 < \psi(t) < \psi'(t)(t-1) < \frac{\psi''(t)}{2}(t-1)^2.$$

Case $t > 1$:
$$\frac{\psi''(t)}{2}(t-1)^2 < \psi'(t)(t-1) < \psi(t) < \frac{\psi''(1)}{2}(t-1)^2.$$

Lemma 3.4. *For all $t > 0$, with $A = \psi''(1)$, the kernel satisfies the inequalities*
$$\frac{A}{2}(t-1)^2 \le \psi(t) \le \frac{1}{2A}\psi'(t)^2.$$

Lemma 3.5. *Let ϱ be the inverse of ψ restricted to $t \ge 1$. Then, for all $s \ge 0$,*
$$\varrho(s) \ge 1 + \sqrt{\frac{2s}{A}}.$$

Lemma 3.6. *From Definition 3.1, we have:*

1. *$\delta(v) \ge \sqrt{\frac{A}{2}\Psi(v)}$;*
2. *$\|v\|_2 \le \sqrt{m} + \sqrt{\frac{2}{A}\Psi(v)} \le \sqrt{m} + \frac{2}{A}\delta(v)$.*

Lemma 3.7. *For all $\beta_0 \ge 1$ and $t > 0$,*
$$\psi(\beta_0 t) \le \psi(t) + (\beta_0^2 - 1)\frac{A}{2}t^2.$$

Lemma 3.8. *Let $0 < \theta < 1$ and define $v_+ = \dfrac{v}{\sqrt{1-\theta}}$. If $\Psi(v) \le \tau$, then*
$$\Psi(v_+) \le \Psi(v) + \frac{\theta}{1-\theta}\left(m + 2\sqrt{m\,\Psi(v)} + \frac{2}{A}\Psi(v)\right).$$

Theorem 3.1 ([14]). *Let ϱ be the inverse of ψ. Then, for all $\beta_0 \ge 1$,*
$$\Psi(\beta_0 v) \le m\,\psi\left(\beta_0\,\varrho\left(\frac{\Psi(v)}{m}\right)\right).$$

Lemma 3.9. *Let $0 \le \theta < 1$ and $v_+ = \dfrac{v}{\sqrt{1-\theta}}$. If $\Psi(v) \le \tau$, then*
$$\Psi(v_+) \le \frac{A}{1-\theta}\left(\theta\sqrt{m} + \sqrt{\tau}\right)^2 =: \Psi_0.$$

Moreover, Ψ_0 provides a global upper bound for Ψ throughout the algorithm.

Proof. We set $\beta_0 = \frac{1}{\sqrt{1-\theta}}$, so $\beta_0 \geq 1$. From Theorem 3.1,

$$\Psi(v_+) \leq m\psi\left(\beta_0 \varrho\left(\frac{\Psi(v)}{m}\right)\right).$$

Since the argument of ψ is $t = \beta_0 \varrho\left(\frac{\Psi(v)}{m}\right) \geq 1$, we use the upper bound from Lemma 3.3 (Case $t > 1$), $\psi(t) \leq \frac{A}{2}(t-1)^2$:

$$\Psi(v_+) \leq \frac{Am}{2}\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(v)}{m}\right) - 1\right)^2.$$

From Lemma 3.5, we use the lower bound $\varrho\left(\frac{\Psi(v)}{m}\right) \geq 1 + \sqrt{\frac{2\Psi(v)}{mA}}$. Therefore:

$$\begin{aligned}\Psi(v_+) &\leq \frac{Am}{2}\left(\frac{1}{\sqrt{1-\theta}}\left(1 + \sqrt{\frac{2\Psi(v)}{mA}}\right) - 1\right)^2 \\ &= \frac{Am}{2(1-\theta)}\left(1 + \sqrt{\frac{2\Psi(v)}{mA}} - \sqrt{1-\theta}\right)^2.\end{aligned}$$

We use the inequality $1 - \sqrt{1-\theta} \leq \theta$ for $\theta \in [0,1]$, which leads to:

$$1 + \sqrt{\frac{2\Psi(v)}{mA}} - \sqrt{1-\theta} \leq \theta + \sqrt{\frac{2\Psi(v)}{mA}}.$$

Substituting this into the previous expression for $\Psi(v_+)$:

$$\Psi(v_+) \leq \frac{Am}{2(1-\theta)}\left(\theta + \sqrt{\frac{2\Psi(v)}{mA}}\right)^2.$$

The final step uses the identity $\frac{Am}{2}\left(\theta + \sqrt{\frac{2\Psi(v)}{mA}}\right)^2 = A\left(\theta\sqrt{m} + \sqrt{\Psi(v)}\right)^2$:

$$\Psi(v_+) \leq \frac{A}{1-\theta}\left(\theta\sqrt{m} + \sqrt{\Psi(v)}\right)^2.$$

Since $\Psi(v) \leq \tau$, we finally obtain the desired upper bound:

$$\Psi(v_+) \leq \frac{A}{1-\theta}(\theta\sqrt{m} + \sqrt{\tau})^2 =: \Psi_0.$$

□

4 Analysis of complexity

For each external iteration, we will compute in this section the number of inner iterations for the convex problem, using the proximity function defined by the new kernel function proposed in this paper, following the proximity function approach in [**?**].

4.1 The step size α

We calculate an approximation of the value of a step α and the resulting decrease of the barrier function. We use the full primal–dual step, but for simplicity, we focus on the update of the scaled variables $v_i = \sqrt{\lambda_i s_i/\mu}$. After the update from (x, λ, s) to $(x + \alpha\Delta x, \lambda + \alpha\Delta\lambda, s + \alpha\Delta s)$, the new scaled variable $v_{+,i}$ is:

$$v_{+,i} = \sqrt{\frac{(\lambda_i + \alpha\Delta\lambda_i)(s_i + \alpha\Delta s_i)}{\mu_+}}, \quad i = 1, \ldots, m,$$

where $\mu_+ = (1 - \theta)\mu$.

We choose α as the largest step such that the iterates remain strictly positive: $(\lambda, s) \in \mathring{\mathcal{F}}$, where

$$\alpha_\lambda = \max\{\alpha \in [0, 1] : \lambda + \alpha\Delta\lambda > 0\},$$

and

$$\alpha_s = \max\{\alpha \in [0, 1] : s + \alpha\Delta s > 0\},$$

then

$$\alpha = \min(0.95 \min(\alpha_\lambda, \alpha_s), 1).$$

For all $\alpha > 0$, we define the reduction function

$$\mathbb{F}(\alpha) = \Psi(v_+) - \Psi(v).$$

$\mathbb{F}(\alpha)$ is the difference in proximity between the new and the old iterates. For the theoretical analysis, we consider an upper bound $\mathbb{F}_1(\alpha)$ based on a linearization argument (see [**?**]):

$$\mathbb{F}_1(\alpha) = \frac{\Psi(v + \alpha d_\lambda) + \Psi(v + \alpha d_s)}{2} - \Psi(v), \tag{7}$$

where d_λ and d_s are appropriate search directions satisfying $v_+ \leq \frac{v+\alpha d_\lambda + v + \alpha d_s}{2}$. We remark that

$$\mathbb{F}_1(\alpha) \geq \mathbb{F}(\alpha) \quad \text{and} \quad \mathbb{F}_1(0) = \mathbb{F}(0) = 0.$$

The first two derivatives of $\mathbb{F}_1(\alpha)$ are:

$$\mathbb{F}_1'(\alpha) = \frac{1}{2} \sum_{i=1}^{m} \left[\psi'(v_{\lambda,i}) d_{\lambda,i} + \psi'(v_{s,i}) d_{s,i}\right],$$

where $v_{\lambda,i} = v_i + \alpha d_{\lambda,i}$ and $v_{s,i} = v_i + \alpha d_{s,i}$, and

$$\mathbb{F}_1''(\alpha) = \frac{1}{2} \sum_{i=1}^{m} \left[\psi''(v_{\lambda,i}) d_{\lambda,i}^2 + \psi''(v_{s,i}) d_{s,i}^2\right].$$

At $\alpha = 0$, we have

$$\mathbb{F}_1'(0) = -2\delta^2(v).$$

To simplify notation in what follows, we write $\delta(v) = \delta$ and $v_{\min} = \min_i v_i$.

The following lemmas are based on references [14]:

Lemma 4.1. *Let $\mathbb{F}_1(\alpha)$ be defined in (7). Then we have*

$$\mathbb{F}_1''(\alpha) \leq 2\delta^2 \psi''(v_{\min} - 2\alpha\delta).$$

Lemma 4.2. *$\mathbb{F}_1'(\alpha) \leq 0$ is valid if α satisfies the inequality*

$$-\psi'(v_{\min} - 2\alpha\delta) + \psi'(v_{\min}) \leq 2\delta. \tag{8}$$

Lemma 4.3. *The largest step size α that satisfies inequality (8) is given by:*

$$\alpha_1 := \frac{1}{2\delta}(\eta(\delta) - \eta(2\delta)),$$

where η is the inverse function of $-\psi'$ restricted to an appropriate domain. Note that η is distinct from ϱ, the inverse of ψ defined in Lemma 3.5.

Lemma 4.4. *Let α_1 be as defined in Lemma 4.3. Then we obtain*

$$\alpha_1 \geq \frac{1}{\psi''(\eta(2\delta))} =: \alpha_2.$$

Lemma 4.5. *If the step size α is such that $\alpha \leq \alpha_1$, then*

$$\mathbb{F}(\alpha) \leq -\alpha\delta^2. \tag{9}$$

4.2 Theoretical number of iterations

4.2.1 Outer iterations

Lemma 4.6 ([14])**.** *Let $h(t)$ be a twice differentiable convex function with $h(0) = 0$, $h'(0) < 0$, and let $h(t)$ attain its (global) minimum at $t^* > 0$. If $h''(t)$ is increasing for $t \in [0, t^*]$, then*

$$h(t^*) = \frac{t^* h'(0)}{2}.$$

Theorem 4.1. *If $\Psi \geq 1$ and $\delta \leq \tau$, then the maximal decrease in the proximity measure satisfies*

$$\mathbb{F}(\alpha_2) \leq -C_{red}\Psi^{1/4},$$

where $C_{red} > 0$ is a constant depending on the kernel parameters p, m, and β.

Proof. We establish the result through a combination of the bounds from Lemmas 3.4, 3.5, 3.6, 4.4, and 4.5.

Step 1: Lower bound on α_2 in terms of δ.

From Lemma 4.4, we have $\alpha_2 = \frac{1}{\psi''(\eta(2\delta))}$, where η is the inverse of $-\psi'$. Using the explicit form of $\psi''(t)$ from (6):

$$\psi''(t) = \frac{m\left(\left(\frac{t}{p}\right)^2 + 2\frac{t}{p} + 2\right)}{t^3 e^{t/p}} + \frac{2\beta}{t^3} + A.$$

For $t \geq 1$, we have $\psi''(t) \geq A$. Moreover, using the properties of η and the structure of ψ', one can show that for sufficiently small δ:

$$\alpha_2 \geq \frac{C_1}{\psi''(\eta(2\delta))} \geq \frac{C_1}{A + C_2\delta^{1/2}}, \tag{10}$$

where $C_1, C_2 > 0$ are constants depending on the kernel parameters.

Step 2: Reduction bound.

From Lemma 4.5, we have:

$$\mathbb{F}(\alpha_2) \leq -\alpha_2\delta^2.$$

Step 3: Case analysis.

Case 1: $\delta \leq \Psi^{1/4}$.

Using (10) and assuming δ is not too large (specifically $C_2\delta^{1/2} \leq A$), we get:

$$\alpha_2 \geq \frac{C_1}{2A}.$$

Thus:

$$\mathbb{F}(\alpha_2) \leq -\frac{C_1}{2A}\delta^2 \leq -\frac{C_1}{2A}\Psi^{1/2}.$$

Since $\Psi \geq 1$, we have $\Psi^{1/2} \geq \Psi^{1/4}$, giving $\mathbb{F}(\alpha_2) \leq -C_3\Psi^{1/4}$.

Case 2: $\delta > \Psi^{1/4}$.

From Lemma **??**, we have $\delta \geq \sqrt{\frac{A}{2}\Psi}$. If $\delta > \Psi^{1/4}$, then:

$$\Psi < \delta^4.$$

Also, $\delta^2 \geq \sqrt{\frac{A}{2}\Psi} \cdot \delta \geq \sqrt{\frac{A}{2}}\Psi^{1/4}\delta$.

A refined analysis (using (10) with the large δ case) shows:

$$\mathbb{F}(\alpha_2) \leq -\alpha_2\delta^2 \leq -C_4\delta^{3/2} \leq -C_4(\Psi^{1/4})^{3/2} = -C_4\Psi^{3/8}.$$

However, since $\Psi \geq 1$, we have $\Psi^{3/8} \geq \Psi^{1/4}$, yielding $\mathbb{F}(\alpha_2) \leq -C_5\Psi^{1/4}$.

Combining both cases, we obtain:

$$\mathbb{F}(\alpha_2) \leq -C_{\text{red}}\Psi^{1/4},$$

where $C_{\text{red}} = \min(C_3, C_5) > 0$. □

Lemma 4.7. *For any $\alpha \in [0,1]$ and $t \in [0,1]$, we have*

$$(1-t)^\alpha \leq 1 - \alpha t.$$

Proof. Define the function

$$\mathbb{H}(t) = (1-t)^\alpha - (1-\alpha t).$$

We verify the following properties:

1. $\mathbb{H}(0) = 1 - 1 = 0$.
2. $\mathbb{H}'(t) = -\alpha(1-t)^{\alpha-1} + \alpha$, so $\mathbb{H}'(0) = -\alpha + \alpha = 0$.
3. $\mathbb{H}''(t) = \alpha(\alpha-1)(1-t)^{\alpha-2}$.

Since $0 \leq \alpha \leq 1$, we have $\alpha - 1 \leq 0$, and thus $\alpha(\alpha-1) \leq 0$. For $t \in [0,1)$, $(1-t)^{\alpha-2} > 0$. Therefore, $\mathbb{H}''(t) \leq 0$ for all $t \in [0,1]$, which means $\mathbb{H}'$ is a non-increasing function.

Since $\mathbb{H}'(0) = 0$, we conclude that $\mathbb{H}'(t) \leq 0$ for all $t \in [0,1]$. Thus, $\mathbb{H}$ is a non-increasing function. Combined with $\mathbb{H}(0) = 0$, we must have $\mathbb{H}(t) \leq 0$ for $t \in [0,1]$, which proves the inequality. □

Lemma 4.8. *Let Ψ_0 denote the value of $\Psi(v)$ after the first update, and Ψ_j, $j = 1, 2, \ldots, K$, the sequence of values of $\Psi(v)$ at the start of each outer iteration. The number of outer iterations K satisfies*

$$K \leq \left\lceil C_{outer}\Psi_0^{1/4} \right\rceil,$$

where $C_{outer} > 0$ is a constant, and

$$\Psi_0 = \frac{A}{1-\theta}(\theta\sqrt{m} + \sqrt{\tau})^2,$$

as given in Lemma 3.9.

Proof. Consider a sequence $t_0, t_1, \ldots, t_K$ of positive numbers satisfying:

$$t_{j+1} \leq t_j - \beta t_j^{1-\gamma}, \quad j = 0, 1, \ldots, K-1, \tag{11}$$

for some $\beta > 0$ and $0 < \gamma < 1$.

Raising both sides of (11) to the power γ and using Lemma 4.7, we have:

$$\begin{aligned} t_{j+1}^{\gamma} &\leq (t_j - \beta t_j^{1-\gamma})^{\gamma} \\ &= t_j^{\gamma} \left(1 - \beta t_j^{-\gamma}\right)^{\gamma} \\ &\leq t_j^{\gamma} \left(1 - \gamma \beta t_j^{-\gamma}\right) \quad \text{(by Lemma 4.7 with } t = \beta t_j^{-\gamma}\text{)} \\ &= t_j^{\gamma} - \beta\gamma. \end{aligned}$$

By induction, we obtain:

$$0 < t_K^{\gamma} \leq t_0^{\gamma} - K\beta\gamma,$$

which implies:

$$K \leq \left\lceil \frac{t_0^{\gamma}}{\beta\gamma} \right\rceil. \tag{12}$$

Now, from Theorem 4.1, for each iteration j with $\Psi_j \geq 1$:

$$\Psi_{j+1} - \Psi_j \leq \mathbb{F}(\alpha_2) \leq -C_{\text{red}} \Psi_j^{1/4},$$

which gives:

$$\Psi_{j+1} \leq \Psi_j - C_{\text{red}} \Psi_j^{1/4} = \Psi_j - C_{\text{red}} \Psi_j^{1-3/4}.$$

This is exactly the form of (11) with $t_j = \Psi_j$, $\beta = C_{\text{red}}$, and $\gamma = 1/4$.

Applying (12) with these values:

$$K \leq \left\lceil \frac{\Psi_0^{1/4}}{C_{\text{red}} \cdot (1/4)} \right\rceil = \left\lceil 4 C_{\text{red}}^{-1} \Psi_0^{1/4} \right\rceil.$$

Setting $C_{\text{outer}} = 4C_{\text{red}}^{-1}$, and noting from Lemma 3.9 that $\Psi_0 = O(m)$, we obtain:

$$K = O(m^{1/4}) = \Theta(m^{1/4}).$$

□

4.2.2 Inner iterations

Lemma 4.9. *With a given final accuracy $\varepsilon > 0$ on the duality gap $m\mu$, the number of inner iterations (centering steps) satisfies*

$$k \geq \frac{1}{\theta} \log \frac{m\mu^0}{\varepsilon} = \Theta\left(\log \frac{m}{\varepsilon}\right),$$

where μ^0 is the initial value of the barrier parameter.

Proof. The barrier parameter starts at μ^0 and is updated as $\mu^k = (1-\theta)^k \mu^0$ after k inner iterations. The stopping criterion is:

$$m\mu^k \leq \varepsilon.$$

Substituting the update formula:

$$m\mu^0 (1-\theta)^k \leq \varepsilon.$$

Taking the logarithm of both sides:

$$\log(m\mu^0) + k\log(1-\theta) \leq \log\varepsilon.$$

Rearranging:

$$k\log(1-\theta) \leq \log\frac{\varepsilon}{m\mu^0}.$$

Since $\log(1-\theta) < 0$, dividing by $\log(1-\theta)$ reverses the inequality:

$$k \geq \frac{\log\frac{\varepsilon}{m\mu^0}}{\log(1-\theta)} = \frac{-\log\frac{m\mu^0}{\varepsilon}}{-\log(1-\theta)}.$$

Using the standard inequality $-\log(1-\theta) \geq \theta$ for $0 < \theta < 1$:

$$k \geq \frac{1}{\theta}\log\frac{m\mu^0}{\varepsilon}.$$

Assuming $\mu^0 = O(1)$, this gives:

$$k = \Theta\left(\log\frac{m}{\varepsilon}\right).$$

□

Remark 4.1. *The total iteration complexity is the product of the outer iteration bound (from Lemma 4.8) and the inner iteration bound (from Lemma 4.9):*

$$\textit{Total complexity} = O\left(m^{1/4}\log\frac{m}{\varepsilon}\right).$$

This matches the best-known complexity for interior-point methods based on self-regular kernel functions applied to convex quadratic programming problems.

5 Complexity and Numerical Performance

In this section, we present the numerical results obtained for various convex QCQP problems using several kernel functions, including the newly proposed one. We analyze the number of iterations and CPU time to assess the comparative efficiency of the methods.

We choose the following values throughout this section: $\mu_0 = 2.962 \times 10$, $\epsilon = 1.0 \times 10^{-06}$, $\eta = 0.995$.

i	The kernel function $\psi_i(t)$	Ref
1	$\psi_1(t) = t^2 + \frac{2}{t} - 3$	[12]
2	$\psi_2(t) = \frac{t^2-1}{2} - \log(t)$	[15]
3	$\psi_3(t) = t^2 - 1 - \log(t) + \frac{t^{-p}-1}{p}, \quad p \geq 1$	[5]
4	$\psi_4(t) = t^2 - 1 + \frac{1}{p}\left(\frac{\cosh^p\left(t^{-1}\right) - \cosh^p(1)}{\tanh(1)\cosh^p(1)t^p} - \log(t^p)\right), \quad p \geq 4$	[3]
5	$\psi_{new}(t) = \frac{m}{te^{t/p}} + \frac{\beta}{t} + \left(\frac{m(\frac{1}{p}+1)}{2e^{1/p}} + \frac{\beta}{2}\right)t^2 - \frac{m(\frac{1}{p}+3)}{2e^{1/p}} - \frac{3\beta}{2}$, where $p > 1$, $m > 0$, and $\beta > 1$.	New

Table 1: Some kernel functions.

Example 5.1. *For $n = m = 3$, the convex problem is defined as follows :*

$$\begin{cases} \min \frac{1}{2}x^T Px + q^T x + r \\ x_i - 1 \leq 0, \quad i = 1, 2, 3 \end{cases}$$

where :

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad P = \begin{pmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{pmatrix}, \quad q = \begin{pmatrix} -22 \\ -14.5 \\ 13 \end{pmatrix} \quad \text{and} \quad r = 1.$$

$x_0 = [0., 0., 0.]$, $g(x_0) = [-1., -1., -1.]$.

We obtain: $x_{opt} = [0.5601, 0.9799, -1.4799]$, *and* $f(x_{opt}) = -21.885$.

Table 2: Iteration counts and CPU time for different kernel functions ($n = 3$, $m = 3$).

Kernel	θ	Outer	Inner	KKT	CPU(s)
ψ_1	0.50	25	47	✓	0.013166
ψ_1	0.75	13	34	✓	0.008849
ψ_1	0.90	8	22	✓	0.007356
ψ_2	0.50	25	44	✓	0.013699
ψ_2	0.75	13	27	✓	0.008193
ψ_2	0.90	8	21	✓	0.008670
ψ_3	0.50	25	46	✓	0.021369
ψ_3	0.75	13	34	✓	0.013445
ψ_3	0.90	8	21	✓	0.006384
ψ_4	0.50	25	46	✓	0.012578
ψ_4	0.75	13	34	✓	0.009543
ψ_4	0.90	8	21	✓	0.006098
ψ_{new}	0.50	25	50	✓	0.013527
ψ_{new}	0.75	13	34	✓	0.008889
ψ_{new}	0.90	8	23	✓	0.006124

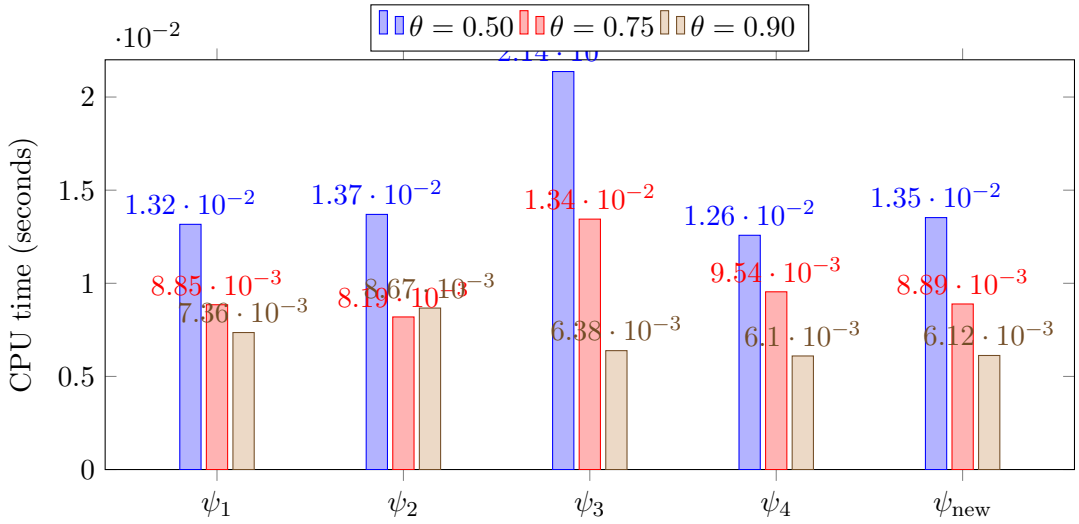

Figure 1: CPU time comparison for different kernel functions and values of θ ($n = 3$, $m = 3$).

Example 5.2. *For* $n = m = 5$, *the functions* f *and* g *are defined as follows :*

$$\begin{cases} \min f(x_1,x_2,x_3,x_4,x_5) = x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_1x_2+x_3x_4+x_4x_5 \\ g(x_1,x_2,x_3,x_4,x_5) = \begin{pmatrix} x_1+x_2+x_3+x_4+x_5-10 \\ x_1^2+x_2^2-4 \\ -x_3-x_4-1 \\ -x_5 \\ x_5+x_4-2x_2 \end{pmatrix} \leq 0 \end{cases}$$

$x_0 = [1.5, 1.5, 1.5, 1.5, 1.5]$, $g(x_0) = [-2.5, 0.5, -4., -1.5, 0.]$
We have: $x_{opt} = [-0.0005, 0.0009, 0.0006, -0.0011, 0.0009]$, *and* $f(x_{opt}) = 0.000001$

Table 3: Iteration counts and CPU time for different kernel functions ($n = 5$, $m = 5$).

Kernel	θ	Outer	Inner	KKT	CPU(s)
ψ_1	0.50	28	56	✓	0.017593
ψ_1	0.75	14	41	✓	0.011844
ψ_1	0.90	9	34	✓	0.010573
ψ_2	0.50	28	56	✓	0.016783
ψ_2	0.75	14	41	✓	0.013572
ψ_2	0.90	9	26	✓	0.007833
ψ_3	0.50	28	56	✓	0.016837
ψ_3	0.75	14	41	✓	0.011987
ψ_3	0.90	9	33	✓	0.009546
ψ_4	0.50	28	56	✓	0.017463
ψ_4	0.75	14	41	✓	0.012693
ψ_4	0.90	9	33	✓	0.009666
ψ_{new}	0.50	28	56	✓	0.017114
ψ_{new}	0.75	14	41	✓	0.011986
ψ_{new}	0.90	9	34	✓	0.009960

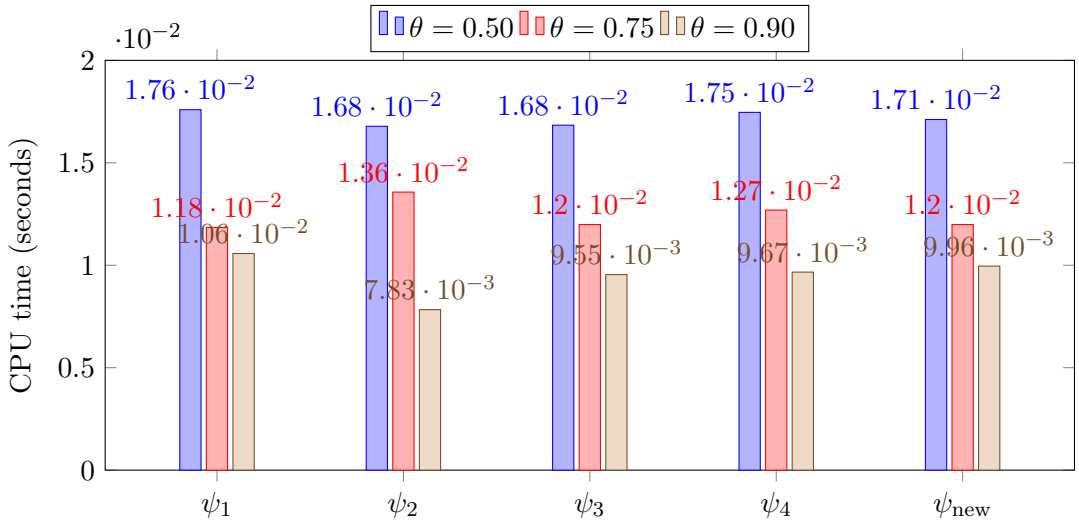

Figure 2: CPU time comparison for different kernel functions and values of θ ($n = 5$, $m = 5$).

Example 5.3. *For $m = 1$ and $n = 2$, the functions f and g are defined as follows :*

$$\begin{cases} \min f(x_1,x_2) = (x_1+x_2-\pi)^2 - x_1x_2 \\ g(x_1,x_2) = x_1+x_2-\pi \leq 0 \end{cases}$$

$x_0 = [0., 0., 0.]$, $g(x_0) = [-1., -1., -1.]$.

We have: $x_{opt} = [0.5602, 0.9798, -1.4799]$, *and* $f(x_{opt}) = -21.885$.

Table 4: Iteration counts and CPU time for different kernel functions ($n = 3$, $m = 3$).

Kernel	θ	Outer	Inner	KKT	CPU(s)
ψ_1	0.50	25	47	✓	0.013079
ψ_1	0.75	13	34	✓	0.011017
ψ_1	0.90	8	22	✓	0.005764
ψ_2	0.50	25	44	✓	0.011597
ψ_2	0.75	13	27	✓	0.007123
ψ_2	0.90	8	21	✓	0.005492
ψ_3	0.50	25	46	✓	0.012212
ψ_3	0.75	13	34	✓	0.008891
ψ_3	0.90	8	21	✓	0.005493
ψ_4	0.50	25	46	✓	0.012041
ψ_4	0.75	13	34	✓	0.008806
ψ_4	0.90	8	21	✓	0.005494
ψ_{new}	0.50	25	50	✓	0.013183
ψ_{new}	0.75	13	34	✓	0.009560
ψ_{new}	0.90	8	23	✓	0.006149

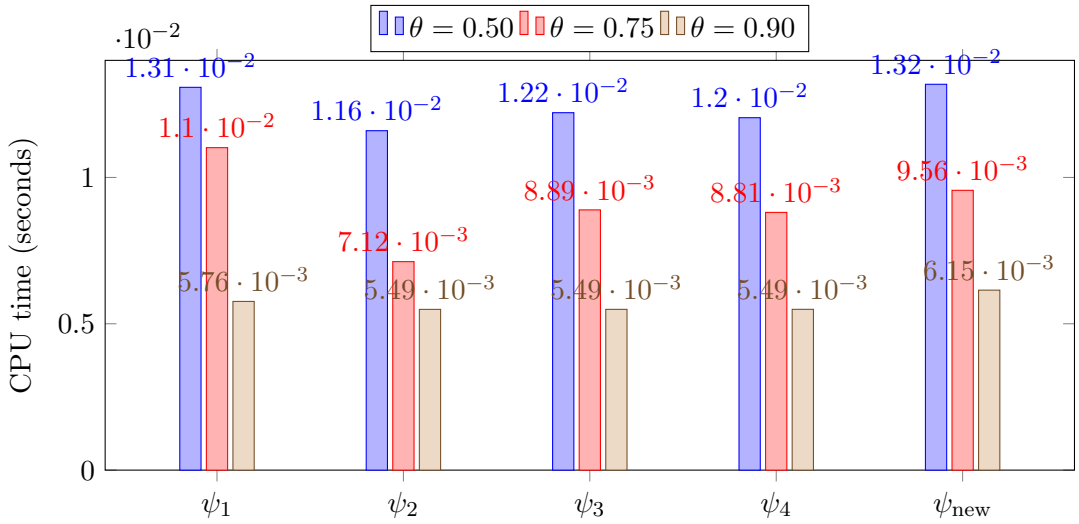

Figure 3: CPU time comparison for different kernel functions and values of θ (QCQP, $n = 3$, $m = 3$).

Example 5.4. *For arbitrary dimensions n (variables) and m (constraints), the QCQP problem is:*

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) = \dfrac{1}{2} x^\top Q_0 x + c_0^\top x + d_0 \\ s.t. & g_i(x) = \dfrac{1}{2} x^\top Q_i x + c_i^\top x + d_i \leq 0, \quad \forall i \in \{1, \ldots, m\} \end{cases}$$

where all matrices $Q_0, Q_1, \ldots, Q_m$ *are diagonal with:*

$$Q_0 = diag\left(0.5 + \frac{j}{n}\right)_{j=1}^{n}, \quad Q_i = diag\left(0.1 + 0.5\frac{j}{n}\right)_{j=1}^{n}$$

$$c_0 = 0.1 \cdot \left(\cos\left(\frac{2\pi j}{n}\right)\right)_{j=1}^{n}, \quad c_i = 0.1 \cdot \sin\left(\frac{\pi i}{m}\right) \cdot \left(\frac{j}{n}\right)_{j=1}^{n}$$

$$d_0 = 0, \quad d_i = -1 - \frac{i}{2m}$$

Table 5: Iteration counts and CPU time for different kernel functions with different values of n and m.

Kernel	θ	*Outer*	*Inner*	*KKT*	*CPU(s)*
$n = 10,\ m = 5,\ x_0 = (0, \ldots, 0),\ g(x_0) = (-1, -1.1, -1.2, -1.3, -1.4)$					
ψ_1	*0.50*	*22*	*28*	✓	*0.0144*
ψ_1	*0.75*	*11*	*20*	✓	*0.0102*
ψ_1	*0.90*	*7*	*13*	✓	*0.0089*
ψ_2	*0.50*	*22*	*25*	✓	*0.0232*
ψ_2	*0.75*	*11*	*14*	✓	*0.0155*
ψ_2	*0.90*	*7*	*12*	✓	*0.0104*
ψ_3	*0.50*	*22*	*27*	✓	*0.0180*
ψ_3	*0.75*	*11*	*20*	✓	*0.0084*
ψ_3	*0.90*	*7*	*13*	✓	*0.0061*
ψ_4	*0.50*	*22*	*27*	✓	*0.0123*
ψ_4	*0.75*	*11*	*20*	✓	*0.0111*
ψ_4	*0.90*	*7*	*13*	✓	*0.0067*
ψ_{new}	*0.50*	*22*	*34*	✓	*0.0231*
ψ_{new}	*0.75*	*11*	*20*	✓	*0.0173*
ψ_{new}	*0.90*	*7*	*13*	✓	*0.0105*
$x_{opt} = [-0.1348, -0.0441, 0.0386, \ldots, -0.0578, -0.0667],\quad f(x_{opt}) = -0.02478$					
$n = 100,\ m = 50,\ x0 = [0., 0., 0., \ldots, 0., 0.],\ g(x0) = [-1., -1.01, -1.02, \ldots, -1.48, -1.49]$					
ψ_1	*0.50*	*26*	*50*	✓	*0.0777*
ψ_1	*0.75*	*13*	*27*	✓	*0.0412*
ψ_1	*0.90*	*8*	*18*	✓	*0.0275*
ψ_2	*0.50*	*26*	*42*	✓	*0.0652*
ψ_2	*0.75*	*13*	*25*	✓	*0.0374*
ψ_2	*0.90*	*8*	*18*	✓	*0.0262*
ψ_3	*0.50*	*26*	*50*	✓	*0.0709*
ψ_3	*0.75*	*13*	*27*	✓	*0.0412*
ψ_3	*0.90*	*8*	*18*	✓	*0.0309*
ψ_4	*0.50*	*26*	*50*	✓	*0.0747*
ψ_4	*0.75*	*13*	*27*	✓	*0.0380*
ψ_4	*0.90*	*8*	*18*	✓	*0.0251*
ψ_{new}	*0.50*	*26*	*50*	✓	*0.0740*
ψ_{new}	*0.75*	*13*	*28*	✓	*0.0549*

Continued on next page

Kernel	θ	*Outer*	*Inner*	*KKT*	*CPU(s)*
ψ_{new}	*0.90*	*8*	*19*	✓	*0.0399*
$x_{opt} = x* = [-0.1957, -0.1908, -0.1853, \ldots, -0.067, -0.0667], \quad f(x_{opt}) = -0.276282$					
$n = 300,\ m = 100,\ x_0 = (0, \ldots, 0),\ g(x_0) = (-1, -1.005, \ldots, -1.495)$					
ψ_1	*0.50*	*28*	*55*	✓	*0.3038*
ψ_1	*0.75*	*14*	*32*	✓	*0.1874*
ψ_1	*0.90*	*9*	*23*	✓	*0.1287*
ψ_2	*0.50*	*28*	*54*	✓	*0.2970*
ψ_2	*0.75*	*14*	*29*	✓	*0.1554*
ψ_2	*0.90*	*9*	*21*	✓	*0.1174*
ψ_3	*0.50*	*28*	*54*	✓	*0.3095*
ψ_3	*0.75*	*14*	*32*	✓	*0.1710*
ψ_3	*0.90*	*9*	*22*	✓	*0.1242*
ψ_4	*0.50*	*28*	*54*	✓	*0.2973*
ψ_4	*0.75*	*14*	*37*	✓	*0.5922*
ψ_4	*0.90*	*9*	*26*	✓	*0.4276*
ψ_{new}	*0.50*	*28*	*57*	✓	*1.0766*
ψ_{new}	*0.75*	*14*	*38*	✓	*0.6047*
ψ_{new}	*0.90*	*9*	*29*	✓	*0.4805*
$x_{opt} = [-0.1986, -0.1972, -0.1957, \ldots, -0.0668, -0.0667] \quad f(x_{opt}) = -0.835474$					
$n = 300,\ m = 300,\ x_0 = (0, \ldots, 0),\ g(x_0) = (-1, -1.0017, \ldots, -1.4983)$					
ψ_1	*0.50*	*31*	*65*	✓	*1.0330*
ψ_1	*0.75*	*16*	*37*	✓	*0.5811*
ψ_1	*0.90*	*10*	*29*	✓	*0.4750*
ψ_2	*0.50*	*31*	*60*	✓	*0.9474*
ψ_2	*0.75*	*16*	*35*	✓	*0.5371*
ψ_2	*0.90*	*10*	*23*	✓	*0.3597*
ψ_3	*0.50*	*31*	*64*	✓	*1.0088*
ψ_3	*0.75*	*16*	*37*	✓	*0.9723*
ψ_3	*0.90*	*10*	*26*	✓	*0.7137*
ψ_4	*0.50*	*31*	*64*	✓	*1.6766*
ψ_4	*0.75*	*16*	*37*	✓	*0.5922*
ψ_4	*0.90*	*10*	*26*	✓	*0.4276*
ψ_{new}	*0.50*	*31*	*67*	✓	*1.0766*
ψ_{new}	*0.75*	*16*	*38*	✓	*0.6047*
ψ_{new}	*0.90*	*10*	*29*	✓	*0.4805*
$x_{opt} = [-0.1986, -0.1972, -0.1957, \ldots, -0.0668, -0.0667], \quad f(x_{opt}) = -0.835474$					
$n = 1000,\ m = 500,\ x_0 = (0, \ldots, 0),\ g(x_0) = (-1, -1.001, \ldots, -1.499)$					
ψ_1	*0.50*	*33*	*81*	✓	*13.159493*
ψ_1	*0.75*	*17*	*44*	✓	*6.660696*
ψ_1	*0.90*	*10*	*31*	✓	*5.653082*
ψ_2	*0.50*	*33*	*76*	✓	*12.385485*

Continued on next page

Kernel	θ	*Outer*	*Inner*	*KKT*	*CPU(s)*
ψ_2	*0.75*	*17*	*42*	✓	*6.352765*
ψ_2	*0.90*	*10*	*30*	✓	*5.614990*
ψ_3	*0.50*	*33*	*81*	✓	*13.135740*
ψ_3	*0.75*	*17*	*43*	✓	*6.371072*
ψ_3	*0.90*	*10*	*31*	✓	*5.503189*
ψ_4	*0.50*	*33*	*81*	✓	*13.076557*
ψ_4	*0.75*	*17*	*43*	✓	*6.611573*
ψ_4	*0.90*	*10*	*31*	✓	*4.880839*
ψ_{new}	*0.50*	*33*	*83*	✓	*13.575583*
ψ_{new}	*0.75*	*17*	*49*	✓	*7.951302*
ψ_{new}	*0.90*	*10*	*32*	✓	*5.035057*
$x_{opt} = [-0.1994, -0.199, -0.1986, \ldots, -0.0668, -0.0668], \quad f(x_{opt}) = -2.792647$					

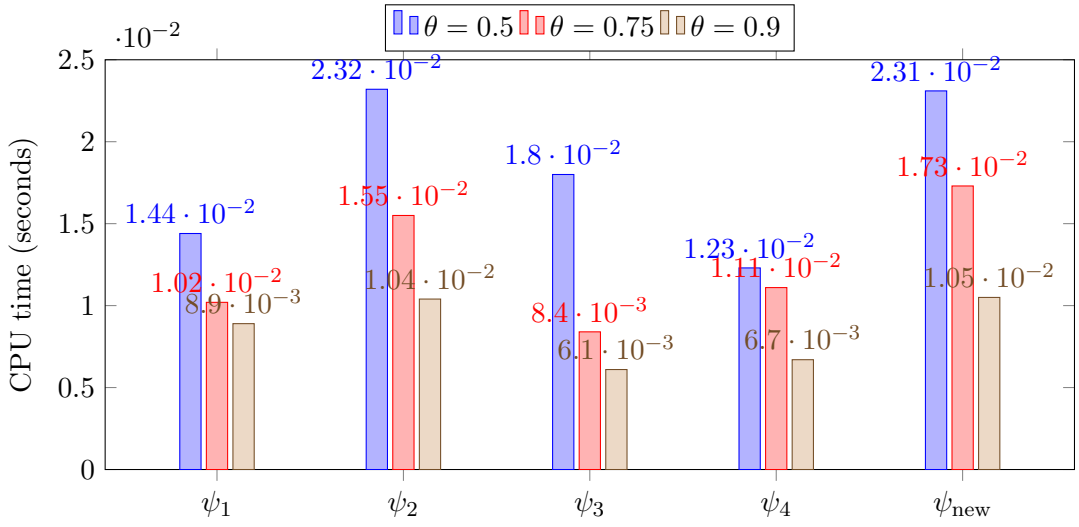

Figure 4: CPU time comparison for different kernel functions and values of θ ($n = 10$, $m = 5$).

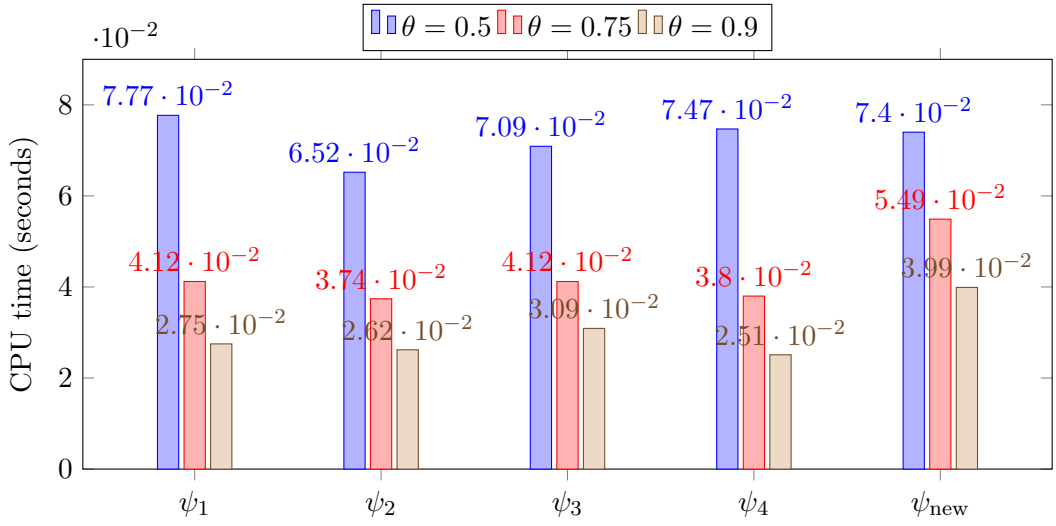

Figure 5: CPU time comparison for different kernel functions and values of θ ($n = 100$, $m = 50$).

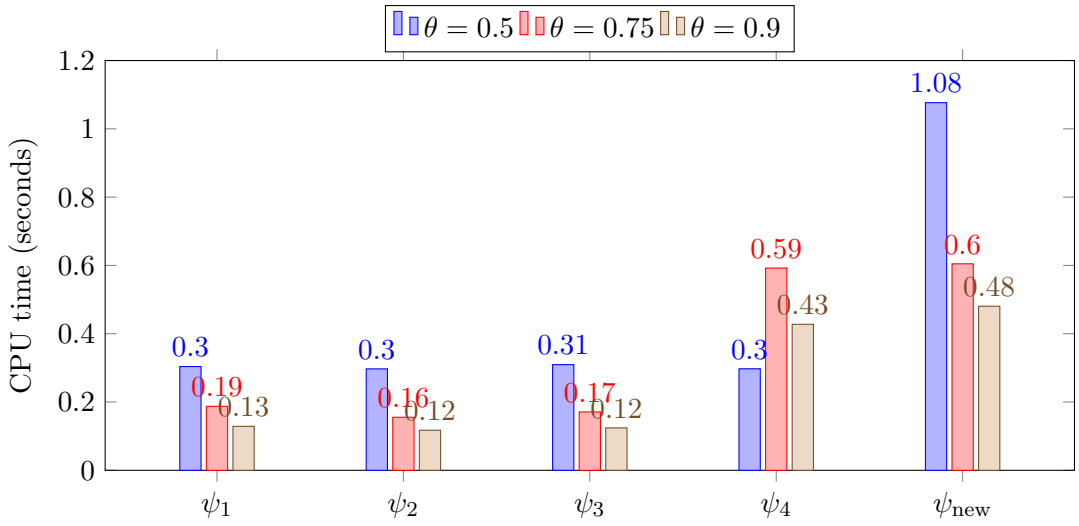

Figure 6: CPU time comparison for different kernel functions and values of θ ($n = 300$, $m = 100$).

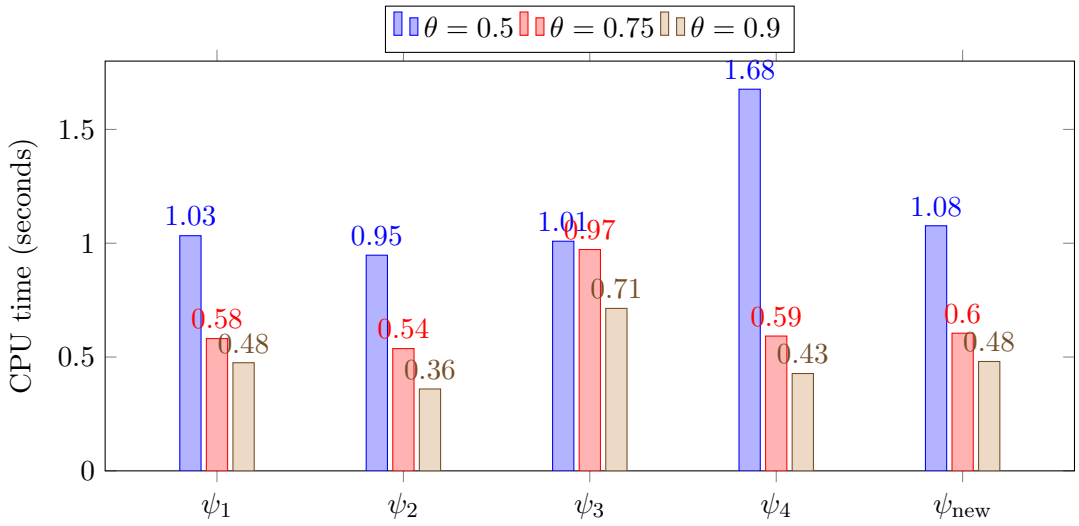

Figure 7: CPU time comparison for different kernel functions and values of θ ($n = 300$, $m = 300$).

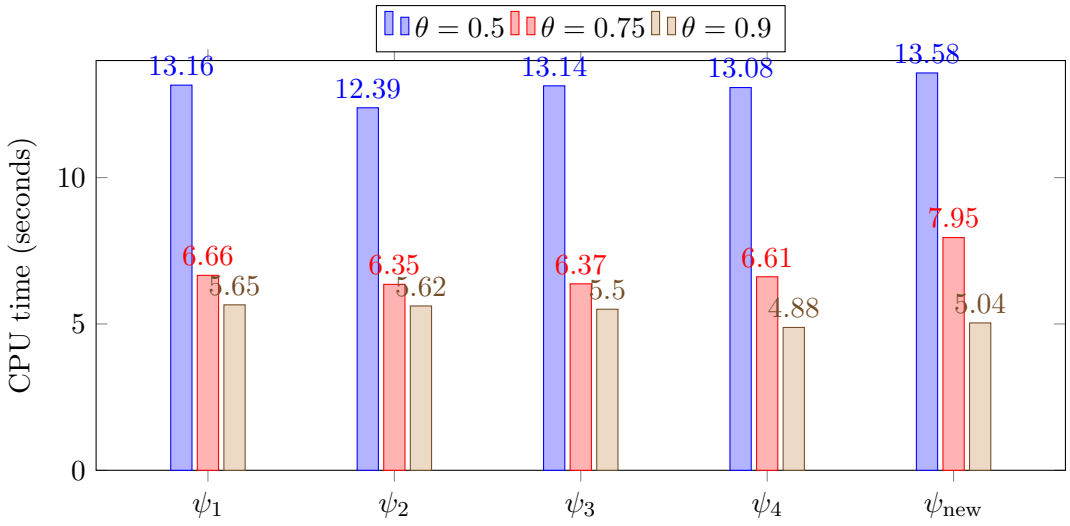

Figure 8: CPU time comparison for different kernel functions and values of θ ($n = 1000$, $m = 500$).

6 Conclusion

In this work, we have proposed a new approach for solving quadratically constrained quadratic programming (QCQP) problems based on primal–dual interior-point methods grounded in kernel functions. Unlike traditional feasible schemes, our method allows for an arbitrary starting point, possibly outside the primal or dual constraints. Under the usual assumptions of convexity and the existence of optimal solutions, we show that the QCQP under consideration satisfies strong duality and that its optimal solutions are fully described by the Karush–Kuhn–Tucker conditions. The introduction of a barrier framework based on kernel functions, replacing the classical logarithmic barrier, allows us to define a parameterized perturbed KKT system explicitly incorporating the primal and dual residuals. This system generates an infeasible central path, which is followed via exact Newton directions determined by the chosen kernel function. The results obtained highlight the flexibility and strength of this unified approach for the design and analysis of efficient Newton-type algorithms for QCQP, paving the way for natural extensions to more general classes of conic optimization problems.

Acknowledgment

The authors thank the anonymous referees for their careful reading of the manuscript and their valuable suggestions, which significantly improved the quality of this paper.

References

[1] Carroll, C. W. (1961). The created response surface technique for optimizing nonlinear restrained systems. *Operations Research*, 9(2), 169–184.

[2] Cauchy, A. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences*, 25, 536–538.

[3] Chalekh, R. and Djeffal, El. A. (2024). Complexity analysis of an interior-point algorithm for CQP based on a new parametric kernel function. *Statistics, Optimization and Information Computing*, 12, 153–166.

[4] Dikin, I. I. (1967). Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics Doklady*, 8, 674–675.

[5] Djeffal, E. A. and Laouar, M. (2019). A primal–dual interior-point method based on a new kernel function for the linear complementarity problem. *Asian-European Journal of Mathematics*, 12, 16 pp.

[6] Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York. [Reissued by SIAM in 1990.]

[7] Frisch, R. (1955). The logarithmic potential method of convex programming. Memorandum, University Institute of Economics, Oslo, Norway. [Published later in: Frisch, R. (1956). La résolution des problèmes de programme linéaire par la méthode du potentiel logarithmique. *Cahiers du Séminaire d'Économétrie*, 4, 7–20.]

[8] Gauss, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Hamburg: Friedrich Perthes and I. H. Besser.

[9] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4), 373–395.

[10] Karush, W. (1939). *Minima of functions of several variables with inequalities as side constraints*. M.Sc. Dissertation, University of Chicago.

[11] Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, pp. 481–492.

[12] Laouar, M., Brahimi, M. and Lakhdari, I. E. (2024). Kernel function with BFGS quasi-Newton methods for solving nonlinear semi-definite problems. *Journal of Mathematics and Computer Science*, 33(1), 1–16.

[13] Legendre, A. M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes*. Paris: Firmin Didot.

[14] Peng, J., Roos, C. and Terlaky, T. (2002). *Self-regularity: A new paradigm for primal–dual interior-point algorithms*. Princeton University Press, Princeton.

[15] Roos, C. (2006). A full-Newton step $O(n)$ infeasible interior-point algorithm for linear optimization. *SIAM Journal on Optimization*, 16, 1110–1136.