Received: xxx **Accepted:** xxx **Published:** xxx.
DOI. xxxxxxx
Volume xxx, Issue xxx, p.p. 1-31: xxx
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

A Multi-Parameter Kernel Function for Primal–Dual Interior-Point Methods in Linear Optimization: Complexity Analysis and Numerical Performance

Abderrahim Guemmaz

Laboratory of Science for Mathematics, Computer Science and Engineering Applications, Department of Mathematics, University Center of Barika, Amdoukal Road, Barika, 05001, Algeria

Correspondence:
Abderrahim Guemmaz
E-mail:
abderrahim.guemmaz@cu-barika.dz

How to Cite
Guemmaz, A. (2026). "A Multi-Parameter Kernel Function for Primal–Dual Interior-Point Methods in Linear Optimization: Complexity Analysis and Numerical Performance". *Control and Optimization in Applied Mathematics*, X(x), 1-31. https://doi.org/10.30473/xxxxxxxxxxxxxx

Abstract. This paper introduces a novel parameterized kernel function for primal–dual interior-point methods (IPMs) applied to the linear optimization (LO) problem. The proposed kernel, governed by two parameters — a base $u > 1$ and a term-count $\ell \in \mathbb{N} \setminus \{0\}$ — employs a multi-exponent power-sum structure (i.e., a sum of power terms with geometrically spaced exponents u, $u^2, \ldots, u^\ell$) that decouples local curvature near the central path from asymptotic growth far from it. This decoupling is impossible with single-parameter kernels such as self-regular or trigonometric variants. Under mild, verifiable conditions, the resulting IPM attains a worst-case iteration complexity of $\mathcal{O}\left(u^\ell\, n^{\frac{u^\ell+1}{2u^\ell}} \log\left(\frac{n}{\epsilon}\right)\right)$ for large-update strategies, and $\mathcal{O}\left(u^{2\ell}\sqrt{n}\,\log\left(\frac{n}{\epsilon}\right)\right)$ for small-update strategies. Choosing $u = \left(\frac{\log n}{2}\right)^{1/\ell}$ recovers the best-known large-update bound $\mathcal{O}\left(\sqrt{n}\log(n)\log\left(\frac{n}{\epsilon}\right)\right)$. Numerical experiments on benchmark LP instances with dimensions up to $n = 1000$ confirm that the $\ell = 1, u = 3.5$ variant consistently outperforms the classical self-regular kernel of Peng et al. in both iteration count and CPU time. The study demonstrates that self-regularity is sufficient but not necessary for optimal complexity, broadening the theoretical foundations of kernel-based interior-point frameworks.

Keywords. Primal-dual interior-point algorithms, Large and small-update methods, Parameterized kernel function, Central path, Complexity bounds, Linear optimization.

MSC. 90C05; 90C51; 65K05.

1 Introduction

The quest for polynomial-time algorithms for linear programming took a decisive turn with Karmarkar's interior-point method [15], which offered a fundamentally different perspective from the simplex method. This breakthrough was quickly followed by the independent development of primal-dual schemes by Kojima et al. [16] and Megiddo [19], laying the foundations of what would become a broad and continuously evolving research area. Since then, the theory of interior-point methods (IPMs) has progressed steadily, with research efforts aiming to reconcile sharp theoretical complexity bounds with strong practical performance on large-scale problems.

Barrier-function-based interior-point techniques have been studied extensively in the optimization literature. Among the most influential contributions is the introduction of *self-concordant* barriers by Nesterov and Nemirovskii [20], which extended the reach of IPMs well beyond linear programming, into nonlinear complementarity problems (NCP), second-order cone programming (SOCP), and semidefinite optimization (SDO). Within this general framework, the best known worst-case iteration bound for linear optimization (LO) in a wide neighborhood of the central path is $\mathcal{O}(n \log(n/\epsilon))$.

A significant refinement of this picture was obtained by Peng, Roos, and Terlaky [22], who introduced the notion of *self-regular* (SR) proximity functions. Building on this concept, they devised primal-dual IPMs not only for LO but also for SDO, SOCP, and NCP, establishing the improved iteration bound $\mathcal{O}(\sqrt{n} \log n \log(n/\epsilon))$, which remains among the strongest results known for large-update (L-UP) methods. This achievement spurred a search for alternative kernel functions (K-F) — including non-SR ones — capable of matching or even improving upon SR-based performance; see, e.g., [4, 5, 12].

Beyond classical logarithmic and trigonometric constructions, recent work has explored a variety of parametric kernel families. Early studies by El Ghami et al. [12] examined logarithmic kernel functions and derived complexity bounds for large-update methods. Later, El Ghami and co-authors [13] were the first to incorporate a trigonometric kernel into a feasible primal-dual IPM, showing that the resulting method solves LO problems with worst-case complexity $\mathcal{O}(n^{3/4} \log(n/\epsilon))$. This line of work inspired numerous follow-up studies in which trigonometric kernels were refined and tested, several of which attained state-of-the-art large-update complexity. More recently, Bouhenache et al. [7] (2025) proposed an interior-point algorithm for linear complementarity problems (LCPs) based on a parameterized hyperbolic kernel function, attaining the best-known bound $\mathcal{O}(\sqrt{n} \log n \log(n/\epsilon))$ for large-update methods. In the same period, Ibtissam et al. [14] extended parametric kernel functions to the class of $P_*(\kappa)$-horizontal linear complementarity problems (HLCP), reaching the same optimal bound. Lee and Cho [17] developed an infeasible-interior-point method with full-Newton steps based on a new family of kernel functions, demonstrating that kernel-based approaches can handle in-

feasible starting points while retaining optimal theoretical guarantees. Most recently, in 2025, Amrane and Fateh [2] introduced an efficient parametric kernel function for linear optimization, again achieving the same best-known complexity bound.

Motivated by these developments, the present work proposes a new large-update (L-UP) primal-dual interior-point method (IPM) for LO, based on a novel class of kernel functions (N-K-F) defined as

$$\varphi_\ell(x) = \frac{\ell\left(x^2-1\right)}{2} - \sum_{j=1}^{\ell} \frac{x^{1-u^j}-1}{1-u^j}, \quad u>1, \quad \ell \in \mathbb{N}\setminus\{0\}. \tag{1}$$

This family generalizes the kernel function of Peng et al. [22]. For it we establish the iteration-complexity bounds

$$\mathcal{O}\left(u^\ell\, n^{\frac{u^\ell+1}{2u^\ell}} \log\left(\frac{n}{\epsilon}\right)\right) \qquad \text{and} \qquad \mathcal{O}\left(u^{2\ell}\sqrt{n}\,\log\left(\frac{n}{\epsilon}\right)\right)$$

for the large-update (L-UP) and small-update (S-UP) methods, respectively.

Despite the success of kernel-based IPMs, existing kernel functions share a common structural limitation: a single parameter is typically responsible for both the local curvature near the central path and the global growth far from it. The self-regular kernel of Peng et al. [22],

$$\varphi_p(x) = \frac{x^2-1}{2} + \frac{x^{1-p}-1}{p-1}, \tag{2}$$

attains the best-known complexity $\mathcal{O}(\sqrt{n}\log n\log(n/\epsilon))$ for L-UP methods, but its steep growth for $x \gg 1$ forces conservative step-size choices and can limit practical performance. Trigonometric kernels, in contrast, grow more mildly but at the expense of reduced curvature, which can weaken the local convergence rate.

Our kernel is designed explicitly to overcome this trade-off. By summing power terms with *different* exponents $u, u^2, \ldots, u^\ell$, it decouples local and global behavior:

- the term with the largest exponent u^ℓ dominates as $x \to \infty$ and governs the overall growth rate;
- the lower-order exponents u^j ($j < \ell$) shape the intermediate region and contribute curvature without inducing excessive steepness.

This multi-exponent construction preserves the essential eligibility properties (non-negativity, strict convexity, exponential decay) and, with a suitable choice of parameters (e.g., $u = (\log n/2)^{1/\ell}$), recovers the best-known $\mathcal{O}(\sqrt{n}\log n\log(n/\epsilon))$ bound for L-UP methods. Theoretically, this shows that self-regularity, while sufficient, is not a necessary condition for optimal complexity. Practically, the added flexibility opens the possibility of tuning ℓ and u to problem

characteristics, potentially yielding larger step sizes and improved computational performance on specific problem classes.

The remainder of the paper is organized as follows. Section 2 reviews the basic concepts and presents a generic IPM framework for LO that is used throughout the paper. Section 3 introduces the proposed kernel function, discusses its properties, and establishes its key technical bounds. Section 4 addresses step-size estimation. Section 5 develops the LO algorithm based on this kernel and derives its complexity bound. Section 6 presents a comparative study against existing algorithms, and Section 7 reports numerical experiments. Section 8 concludes the paper.

Throughout the paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and Euclidean norm $\| \cdot \|$. The sets $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ denote the nonnegative and strictly positive orthants, respectively. For $z, w \in \mathbb{R}^n$, $z_{\min}$ denotes the smallest component of z, and zw denotes the componentwise (Hadamard) product. The matrix $Z = \operatorname{diag}(z)$ is the diagonal matrix whose entries are the components of z, and e denotes the all-ones vector. For functions $f(z), g(z) : \mathbb{R}^n_{++} \to \mathbb{R}^n_{++}$, we write $f(z) = O(g(z))$ if $f(z) \leq C_1\, g(z)$ for some constant $C_1 > 0$, and $f(z) = \Theta(g(z))$ if there exist constants $C_2, C_3 > 0$ such that

$$C_2\, g(z) \leq f(z) \leq C_3\, g(z).$$

2 Preliminaries

We consider a new primal-dual IPM for solving the LO problem in standard form

$$\min \{ \langle c, t \rangle : Mt = b,\ t \geq 0 \}, \tag{Pr}$$

together with its dual

$$\max \{ \langle b, r \rangle : M^T r + s = c,\ s \geq 0 \}, \tag{Du}$$

where $t, c, s \in \mathbb{R}^n$, $r, b \in \mathbb{R}^m$, and $M \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(M) = m$.

Polynomial-time IPMs for LO were pioneered by Karmarkar [15], and the methodology has since been substantially developed, becoming a standard tool for linear optimization and its variants [11]. For a comprehensive treatment, see [4, 5, 22, 24, 26].

The existence of a strictly feasible starting point (t^0, r^0, s^0) for problems (Pr) and (Du) is equivalent to the so-called interior-point condition (IPC), i.e., the existence of (t^0, r^0, s^0) such that

$$Mt^0 = b, \quad t^0 > 0, \quad M^T r^0 + s^0 = c, \quad s^0 > 0. \tag{3}$$

Finding optimal solutions of (Pr) and (Du) is equivalent to solving the system

$$Mt = b, \quad t \geq 0, \quad M^T r + s = c, \quad s \geq 0, \quad ts = 0. \tag{4}$$

The central idea underlying primal-dual IPMs for LO is to replace the complementarity condition — the third equation in system (4), common to both (Pr) and (Du) — by its parameterized counterpart $ts = \mu e$ for a parameter $\mu > 0$. This leads to the perturbed system

$$Mt = b, \quad t \geq 0, \quad M^T r + s = c, \quad s \geq 0, \quad ts = \mu e. \tag{5}$$

The central path is the foundational concept underlying IPMs: it is the continuous trajectory of points $(t(\mu), r(\mu), s(\mu))$, each being the unique solution of (5) for a given $\mu > 0$, with existence and uniqueness guaranteed by the IPC. The components $t(\mu)$ and $(r(\mu), s(\mu))$ are called the μ-centers of the primal (Pr) and dual (Du) problems, respectively.

The significance of this path for LO was first highlighted by Megiddo [19] and Sonnevend [25]. Its key property is that, as $\mu \to 0$, the path converges to a point satisfying the complementarity condition, thereby yielding an optimal solution of (Pr) and (Du).

From a theoretical standpoint, the IPC can be assumed without loss of generality, e.g., by initializing at $t^0 = s^0 = e$. In practice, problems lacking such a point can be embedded into a homogeneous self-dual formulation, a standard technique [24] that introduces auxiliary variables and constraints to guarantee feasibility.

The task of an IPM is to numerically trace the central path. Starting from a known point — e.g., $(t(1), r(1), s(1))$ with $t(1) = s(1) = e$, corresponding to $\mu = 1$ — the barrier parameter is reduced by a factor $(1 - \theta)$, $\theta \in (0, 1)$, to advance along the path. The corresponding search direction is obtained by solving the Newton system

$$M\Delta t = 0, \quad M^T \Delta r + \Delta s = 0, \quad s\Delta t + t\Delta s = \mu e - ts. \tag{6}$$

This system has a unique solution $(\Delta t, \Delta r, \Delta s)$, and a line search determines a step size α that keeps the new iterate feasible. Repeating this Newton step (the inner loop) continues until the iterate sufficiently approximates the target μ-center $(t(\mu), r(\mu), s(\mu))$.

Once the current μ-center is adequately approximated, the outer loop reduces the barrier parameter via $\mu \leftarrow (1 - \theta)\mu$, and a new Newton system is solved toward the smaller target. This cycle of reducing μ and re-centering continues until a termination criterion such as $n\mu \leq \epsilon$ is met, at which point the final iterate is an ϵ-approximate optimal solution of (Pr) and (Du). The update after a Newton step of size α is

$$t_+ = t + \alpha\Delta t, \quad s_+ = s + \alpha\Delta s, \quad r_+ = r + \alpha\Delta r, \tag{7}$$

with $0 < \alpha \leq 1$.

To simplify the system, we introduce the scaled vector v and the corresponding scaled directions d_t and d_s:

$$v = \sqrt{\frac{ts}{\mu}}, \qquad d_t = \frac{v\Delta t}{t}, \qquad d_s = \frac{v\Delta s}{s}. \tag{8}$$

System (6) can then be rewritten as

$$\overline{M}d_t = 0, \qquad \overline{M}^T\Delta r + d_s = 0, \qquad d_t + d_s = v^{-1} - v, \tag{9}$$

where

$$\overline{M} = \frac{1}{\mu}MV^{-1}T, \qquad V = \text{diag}(v), \qquad T = \text{diag}(t). \tag{10}$$

The right-hand side of the last equation in (9) is precisely the negative gradient of the logarithmic barrier function $\Phi(v)$, i.e., $d_t + d_s = -\nabla\Phi(v)$. This allows system (9) to be written more compactly as

$$\overline{M}d_t = 0, \qquad \overline{M}^T\Delta r + d_s = 0, \qquad d_t + d_s = -\nabla\Phi(v). \tag{11}$$

Here the barrier function $\Phi(v) : \mathbb{R}^n_{++} \to \mathbb{R}_+$ acts as a proximity function, measuring the distance between the current iterate and the target μ-center:

$$\Phi(v) = \Phi(t, s; \mu) = \sum_{i=1}^{n} \varphi(v_i), \tag{12}$$

with kernel function

$$\varphi(v_i) = \frac{v_i^2 - 1}{2} - \log v_i. \tag{13}$$

We also define the norm-based proximity measure $\sigma(v) : \mathbb{R}^n_{++} \to \mathbb{R}_+$ by

$$\sigma(v) = \frac{1}{2}\|\nabla\Phi(v)\| = \frac{1}{2}\|d_t + d_s\|. \tag{14}$$

The kernel $\varphi(x)$ in (13) corresponds to the classical logarithmic barrier. In this work, we introduce a new kernel function $\varphi_\ell(x)$, giving rise to a new barrier function $\Phi_\ell(v)$; both are formally defined in Section 3. The pair (t, s) coincides with the μ-center $(t(\mu), s(\mu))$ if and only if $v = e$, so $\Phi_\ell(v)$ measures how close (t, s) is to the μ-center. A threshold $\tau > 0$ governs the algorithm's main loop: if $\Phi_\ell(v) \le \tau$, the iterate is considered sufficiently close to the μ-center, and an outer iteration is performed by reducing μ by the factor $(1 - \theta)$, $\theta \in (0, 1)$, targeting the new μ-center. Otherwise, if $\Phi_\ell(v) > \tau$, one or more inner (Newton) iterations are performed via (7) until the proximity condition $\Phi_\ell(v) \le \tau$ is restored. This process continues until the termination criterion $n\mu < \epsilon$ is satisfied, yielding an ϵ-approximate solution of the LO problem.

The parameters τ, θ, and the step size α should be chosen so as to minimize the total number of iterations. The choice of the barrier-update parameter θ is particularly important, both theoretically and practically: when θ is a constant independent of the dimension n (e.g., $\theta = 1/2$),

the algorithm is classified as a large-update (L-UP), or long-step, method; when θ depends on n (e.g., $\theta = 1/\sqrt{n}$), it is classified as a small-update (S-UP) method. The generic algorithm is summarized in Algorithm 1. For L-UP methods, the best result obtained to date is $\mathcal{O}(\sqrt{n}\log n\log(n/\epsilon))$, long the best known bound for this class [3]; for S-UP methods, the best known complexity is $\mathcal{O}(\sqrt{n}\log(n/\epsilon))$ [21].

In this work, we introduce a new kernel function (N-K-F) for primal-dual IPMs that yields a complexity bound for L-UP methods improving on previous results based on logarithmic kernel functions. Specifically, the resulting algorithm achieves complexity $\mathcal{O}\big(u^{\ell}n^{\frac{u^{\ell}+1}{2u^{\ell}}}\log(n/\epsilon)\big)$ for L-UP methods and $\mathcal{O}\big(u^{2\ell}\sqrt{n}\log(n/\epsilon)\big)$ for S-UP methods. A more compelling strategy is to choose u as a function of the dimension n and of ℓ so as to minimize the iteration bound. In particular, taking $u = \left(\frac{\log n}{2}\right)^{1/\ell}$ yields the bound $\mathcal{O}(\sqrt{n}\log n\log(n/\epsilon))$, the best known complexity for large-update methods, and a notable improvement over the bound previously established by El Ghami et al. [12] for L-UP methods based on a logarithmic kernel function.

Algorithm 1 Generic Primal–Dual Interior-Point Method for Linear Optimization

Input: Proximity function $\Phi_\ell(v)$; accuracy parameter $\epsilon > 0$; proximity parameter $\tau > 1$; update parameter $\theta \in (0, 1)$.

1. Initialization. Set

$$t^0 = e, \quad s^0 = e, \quad \mu^0 = 1, \quad v^0 = e.$$

2. Outer iteration. While $n\mu \geq \epsilon$, execute the following steps.

a. Update the barrier parameter:

$$\mu \leftarrow (1-\theta)\,\mu.$$

b. Update the scaling vector:

$$v \leftarrow \sqrt{\frac{t\cdot s}{\mu}}.$$

c. **Inner iteration. While** $\Phi_\ell(t, s; \mu) > \tau$, execute the following steps.

i. Solve system (11) with $\Phi(v)$ replaced by $\Phi_\ell(v)$ to compute the Newton direction $(\Delta t,\, \Delta r,\, \Delta s)$.

ii. Determine the appropriate step size $\alpha > 0$.

iii. Update the primal–dual variables:

$$t \leftarrow t + \alpha\,\Delta t, \qquad r \leftarrow r + \alpha\,\Delta r, \qquad s \leftarrow s + \alpha\,\Delta s.$$

3. Return the approximate optimal solution $(t,\, r,\, s)$.

Interpretation of Algorithmic Parameters θ and τ

Role of θ (barrier update parameter). *The parameter $\theta \in (0,1)$ determines how aggressively the barrier parameter μ is reduced at each outer iteration, $\mu \leftarrow (1-\theta)\mu$. A* small θ *(e.g.,* 0.01*) corresponds to a* short-step *method: the iterate remains close to the central path, many outer iterations are needed, but each inner loop requires only a few Newton steps. A* large θ *(e.g.,* 0.99*) yields a* long-step *(*large-update*) method: μ drops sharply, the iterate may leave a narrow neighborhood of the central path, and more inner iterations are needed to restore proximity. The two extreme values* 0.01 *and* 0.99 *are used in Tables 5 and 6 precisely to illustrate these two qualitatively different regimes.*

Role of τ (proximity threshold). *The scalar $\tau > 0$ defines the width of the neighborhood of the μ-center. When the proximity measure $\Phi_\ell(v)$ exceeds τ, inner (Newton) iterations are performed; once $\Phi_\ell(v) \leq \tau$, the current iterate is regarded as sufficiently centered and μ is reduced. A larger τ allows the algorithm to operate in a wider neighborhood, which often reduces the number of outer iterations but may increase the risk of instability or require smaller step sizes. Following standard practice, we set $\tau = \mathcal{O}(n)$ for large-update methods and $\tau = \mathcal{O}(1)$ for short-step methods; this choice directly affects the complexity bounds derived in Section 5.*

3 The Properties of the (N-K-F)

This section presents the new kernel function (N-K-F) introduced in (1) and establishes its fundamental properties, together with the technical results needed for the complexity analysis developed in the subsequent sections.

The kernel function (1) generalizes and extends the kernel function proposed by Peng, Roos, and Terlaky [22]. The motivation for this particular form is to allow finer control over the behavior of the barrier both near and far from the central path. When x is close to 1 (near the central path), the behavior of $\varphi_\ell(x)$ is shaped jointly by all the terms in the sum: the quadratic term provides the basic curvature, while the terms with exponents u^j further adjust the curvature around the μ-center. When x is large (far from the central path), the terms x^{1-u^j} become negligible and the quadratic term $\frac{\ell x^2}{2}$ dominates; however, the largest exponent u^ℓ still governs the speed of this transition. This means that u and ℓ can be chosen so as to avoid an excessively steep kernel — which would otherwise force smaller step sizes in the algorithm — and this added flexibility is expected to translate into improved complexity behavior.

Relation to Existing Kernel Functions and Non-Triviality of the Generalization

The proposed kernel $\varphi_\ell(x)$ generalizes the self-regular kernel

$$\varphi_p(x) = \frac{x^2-1}{2} + \frac{x^{1-p}-1}{p-1},$$

of Peng et al. [22] by replacing its single power term with a sum of ℓ terms. This generalization is non-trivial: in a single-parameter kernel, the same parameter simultaneously controls the curvature near $x = 1$ and the asymptotic growth as $x \to \infty$. In our multi-exponent sum $\sum_{j=1}^{\ell} x^{1-u^j}$, by contrast, the lower exponents u^j ($j < \ell$) shape the barrier near the central path, while the largest exponent u^ℓ dominates the growth far from it — a decoupling of local and global behavior that is impossible for any single-term kernel. Indeed, setting $p = 2\ell$ in $\varphi_p(x)$ gives

$$\varphi_{2\ell}(x) = \frac{x^2-1}{2} + \frac{x^{1-2\ell}-1}{2\ell-1},$$

which still contains a single power term, whereas our kernel contains ℓ distinct power terms with coefficients $\frac{1}{1-u^j}$, producing an interaction effect that no single-term kernel can replicate.

For $\ell = 1$, the kernel reduces to $\varphi_1(x) = \frac{1}{2}(x^2-1) - \log x$, the classical quadratic barrier. For $\ell \geq 2$, the additional terms increase the second derivative at $x = 1$ to

$$\varphi''_\ell(1) = \ell + \sum_{j=1}^{\ell} u^j,$$

making the barrier "stiffer" near the central path. This added stiffness allows larger reductions of the barrier parameter μ per outer iteration, leading to faster overall convergence. In our numerical experiments (Table 5), the variant with $\ell = 2$, $u = 2$ required approximately 19.5% fewer iterations than the $\ell = 1$ variant, while incurring a comparable per-iteration cost. A similar intent of balancing local curvature against asymptotic growth underlies the kernel of Amini and Haseli [1]. A systematic classification of eligible kernel functions is given by Lesaja and Roos [18]; our kernel satisfies all four eligibility conditions (C1)–(C4), and for $\ell \geq 2$ its barrier degree $q = \ell + 1/2$ places it in the class that attains the best-known complexity bound $\mathcal{O}(\sqrt{n}\log n \log(n/\epsilon))$.

It is straightforward to verify that $\varphi_\ell(x) \to \infty$ as $x \to 0^+$ or $x \to \infty$, confirming that $\varphi_\ell(x)$ satisfies the basic requirement of a kernel function. For the analysis that follows, we record its first three derivatives:

$$\varphi'_\ell(x) = \ell x - \sum_{j=1}^{\ell} x^{-u^j}, \tag{15}$$

$$\varphi''_\ell(x) = \ell + \sum_{j=1}^{\ell} u^j x^{-u^j-1}, \tag{16}$$

$$\varphi_\ell'''(x) = -\sum_{j=1}^{\ell} u^j \left(u^j + 1\right) x^{-u^j - 2}. \tag{17}$$

3.1 Eligibility of the (N-K-F)

The following lemma establishes the eligibility of the kernel function (1).

Lemma 1. Let $\varphi_\ell(x)$ be as defined in (1) and let $x > 0$. Then

$$\varphi_\ell''(x) > \ell, \tag{18}$$
$$\varphi_\ell'''(x) < 0, \tag{19}$$
$$x\varphi_\ell''(x) - \varphi_\ell'(x) > 0, \tag{20}$$
$$x\varphi_\ell''(x) + \varphi_\ell'(x) > 0. \tag{21}$$

Proof. Properties (18) and (19) follow immediately from (16) and (17), respectively. For (20) and (21), expressions (15) and (16) give

$$x\varphi_\ell''(x) - \varphi_\ell'(x) = \sum_{j=1}^{\ell} \left(1 + u^j\right) x^{-u^j} > 0,$$

which proves (20), and

$$x\varphi_\ell''(x) + \varphi_\ell'(x) = 2\ell x + \sum_{j=1}^{\ell} \left(u^j - 1\right) x^{-u^j},$$

whose right-hand side is positive, which proves (21). □

Property (21) in Lemma 1 is equivalent to the convexity of the composite function $x \mapsto \varphi_\ell(e^x)$, i.e., to the condition

$$\varphi_\ell(\sqrt{x_1 x_2}) \leq \frac{1}{2}\left(\varphi_\ell(x_1) + \varphi_\ell(x_2)\right), \qquad \text{for all } x_1, x_2 \geq 0. \tag{22}$$

This property is well documented in the literature; see, e.g., [13, 15, 26]. Our contribution here lies in establishing it, together with the eligibility properties above, for the proposed kernel function (N-K-F).

Interpretation of Kernel Parameters u and ℓ

Role of u ***and*** ℓ***.*** *The kernel $\varphi_\ell(x)$ depends on two parameters: the base $u > 1$ and the number of terms $\ell \in \mathbb{N} \setminus \{0\}$. These parameters jointly control the asymptotic growth and the local curvature of the barrier:*

- ***Asymptotic growth*** ***($x \to \infty$).*** *For large x, the dominant term in $\varphi_\ell(x)$ is $\frac{\ell x^2}{2}$, while the sum $\sum_j x^{1-u^j}/(u^j-1)$ decays rapidly. The rate at which the derivative $\varphi'_\ell(x) = \ell x - \sum_j x^{-u^j}$ approaches the linear function ℓx is governed by the largest exponent u^ℓ. A larger u^ℓ makes the kernel steeper far from the central path, which forces smaller default step sizes but, when tuned optimally, still yields the best-known complexity bounds.*

- ***Curvature near*** $x = 1$***.*** *The second derivative at $x = 1$ is*

$$\varphi''_\ell(1) = \ell + \sum_{j=1}^{\ell} u^j = \ell + \frac{u(u^\ell - 1)}{u-1}.$$

This value measures how quickly the proximity grows as we move away from the μ-center. Larger ℓ or larger u increase $\varphi''_\ell(1)$, making the kernel "narrower" around the central path; this typically allows larger reductions of the barrier parameter per outer iteration, but may call for more conservative step sizes.

- ***Flexibility.*** *The use of a* sum *of power terms with* distinct *exponents $u, u^2, \ldots, u^\ell$ provides a flexible mechanism for shaping the kernel between the local and asymptotic regimes. The lower exponents (u^j, $j < \ell$) contribute curvature without causing excessive steepness, effectively decoupling the two regimes — a decoupling that is impossible with single-parameter kernels (e.g., self-regular or trigonometric kernels).*

In the numerical experiments we adopted $u = 2$ and $\ell = 2$ as a compromise: $u = 2$ gives moderate growth, and $\ell = 2$ adds one extra degree of freedom without excessive computational overhead. These values illustrate the behavior of the new kernel while keeping the implementation simple.

We now present a series of technical results required for the convergence analysis.

Lemma 2. For $\varphi_\ell(x)$ we have

$$\frac{\ell}{2}(x-1)^2 \le \varphi_\ell(x) \le \frac{1}{2}\left[\varphi'_\ell(x)\right]^2, \qquad x > 0, \tag{23}$$

$$\varphi_\ell(x) \le \frac{1}{2}\left[\frac{(u-1)\ell + u\left(u^\ell - 1\right)}{u-1}\right](x-1)^2, \qquad x > 1. \tag{24}$$

Proof. For (23), using (18), we have

$$\varphi_\ell(x) = \int_1^x \int_1^t \varphi''_\ell(y)\, dy\, dt \ge \ell \int_1^x \int_1^t dy\, dt = \frac{\ell}{2}(x-1)^2.$$

Also,

$$\begin{aligned}\varphi_\ell(x) &= \int_1^x \int_1^t \varphi_\ell''(y)\,dy\,dt \le \int_1^x \int_1^t \varphi_\ell''(y)\varphi_\ell''(t)\,dy\,dt \\ &= \int_1^x \varphi_\ell''(t)\varphi_\ell'(t)\,dt \\ &= \int_1^x \varphi_\ell'(t)\,d\varphi_\ell'(t) = \frac{1}{2}\left[\varphi_\ell'(x)\right]^2.\end{aligned}$$

For (24), since $\varphi_\ell(1) = \varphi_\ell'(1) = 0$, $\varphi_\ell'''(x) < 0$, and $\varphi_\ell''(1) = \ell + \sum_{j=1}^{\ell} u^j = \ell + \frac{u(u^\ell-1)}{u-1}$, Taylor's theorem yields

$$\begin{aligned}\varphi_\ell(x) &= \varphi_\ell(1) + \varphi_\ell'(1)(x-1) + \frac{1}{2}\varphi_\ell''(1)(x-1)^2 + \frac{1}{6}\varphi_\ell'''(\xi)(x-1)^3 \\ &= \frac{1}{2}\varphi_\ell''(1)(x-1)^2 + \frac{1}{6}\varphi_\ell'''(\xi)(x-1)^3 \\ &\le \frac{1}{2}\varphi_\ell''(1)(x-1)^2 \\ &= \frac{1}{2}\left[\frac{(u-1)\ell + u\left(u^\ell-1\right)}{u-1}\right](x-1)^2,\end{aligned}$$

for some ξ with $1 \le \xi \le x$. This concludes the proof. □

We denote by ς the inverse of $\varphi_\ell(x)$ for $x \ge 1$, and by ϱ the inverse of $-\frac{1}{2}\varphi_\ell'(x)$ for $x \in]0,1]$, where $\varsigma : [0,\infty[\to [1,+\infty[$ and $\varrho : [0,\infty[\to]0,1]$. The following lemma makes use of these functions.

Lemma 3. For $\varphi_\ell(x)$ we have

$$1 + \sqrt{\frac{2(u-1)s}{(u-1)\ell + u\left(u^\ell-1\right)}} \le \varsigma(s) \le 1 + \sqrt{\frac{2s}{\ell}}, \qquad s \ge 0, \tag{25}$$

$$\varrho(z) > \left[\frac{\ell}{2z+\ell}\right]^{\frac{1}{u^\ell}}, \qquad z > 0. \tag{26}$$

Proof. To prove (25), set $s = \varphi_\ell(x)$, $x \ge 1$, so that $\varsigma(s) = x$, $x \ge 1$. By (23) in Lemma 2, $\varphi_\ell(x) \ge \frac{\ell}{2}(x-1)^2$, hence $s \ge \frac{\ell}{2}(x-1)^2$, which gives $x = \varsigma(s) \le 1 + \sqrt{\frac{2s}{\ell}}$. Similarly, by (24) in Lemma 2,

$$s = \varphi_\ell(x) \le \frac{1}{2}\left[\frac{(u-1)\ell + u\left(u^\ell-1\right)}{u-1}\right](x-1)^2, \qquad x \ge 1,$$

so that

$$x = \varsigma(s) \ge 1 + \sqrt{\frac{2(u-1)s}{(u-1)\ell + u\left(u^\ell-1\right)}}.$$

For (26), let $z = -\frac{1}{2}\varphi_\ell'(x)$, $x \in]0,1]$, i.e., $2z = -\varphi_\ell'(x)$. By the definition of $\varphi_\ell'(x)$,

$$2z = -\ell x + \sum_{j=1}^{\ell} x^{-u^j} > -\ell + \sum_{j=1}^{\ell} x^{-u^\ell} = -\ell + \ell x^{-u^\ell},$$

which implies $x = \varrho(z) > \left[\frac{\ell}{2z+\ell}\right]^{\frac{1}{u^\ell}}$. This concludes the proof. □

Lemma 4. Let $\varsigma : [0, \infty[\to [1, +\infty[$ be the inverse of $\varphi_\ell(x)$ over $x \geq 1$. Then

$$\Phi_\ell(\beta v) \leq n\varphi_\ell\left(\beta\varsigma\left(\frac{\Phi_\ell(v)}{n}\right)\right), \qquad v \in \mathbb{R}^n_{++},\ \beta \geq 1.$$

Proof. The result follows directly by combining properties (19) and (20) with the argument used in Lemma 2.4 of [3]. □

Lemma 5. Let $0 \leq \theta < 1$ and $v_+ = \dfrac{v}{\sqrt{1-\theta}}$. If $\Phi_\ell(v) \leq \tau$, then

$$\Phi_\ell(v_+) \leq \frac{\theta n\ell + 2\tau + 2\sqrt{2\tau n\ell}}{2(1-\theta)}.$$

Proof. Since $\frac{1}{\sqrt{1-\theta}} \geq 1$ and $\varsigma\left(\frac{\Phi_\ell(v)}{n}\right) \geq 1$, we have $\frac{1}{\sqrt{1-\theta}}\varsigma\left(\frac{\Phi_\ell(v)}{n}\right) \geq 1$. For $x \geq 1$ we also have $\varphi_\ell(x) \leq \frac{\ell(x^2-1)}{2}$.

Applying Lemma 4 with $\beta = \frac{1}{\sqrt{1-\theta}}$, then (25), and using $\Phi_\ell(v) \leq \tau$, we obtain

$$\begin{aligned}
\Phi_\ell(v_+) &\leq n\varphi_\ell\left(\frac{1}{\sqrt{1-\theta}}\varsigma\left(\frac{\Phi_\ell(v)}{n}\right)\right) \\
&\leq \frac{\ell n}{2}\left(\left[\frac{\varsigma\left(\frac{\Phi_\ell(v)}{n}\right)}{\sqrt{1-\theta}}\right]^2 - 1\right) = \frac{\ell n}{2(1-\theta)}\left(\left[\varsigma\left(\frac{\Phi_\ell(v)}{n}\right)\right]^2 - (1-\theta)\right) \\
&\leq \frac{\ell n}{2(1-\theta)}\left(\left[1 + \sqrt{\frac{2\Phi_\ell(v)}{\ell n}}\right]^2 - (1-\theta)\right) \\
&\leq \frac{\ell n}{2(1-\theta)}\left(2\sqrt{\frac{2\tau}{\ell n}} + 2\frac{\tau}{\ell n} + \theta\right) = \frac{\theta n\ell + 2\tau + 2\sqrt{2\tau n\ell}}{2(1-\theta)}.
\end{aligned}$$

This proves the result. □

Denote

$$(\Phi_\ell)_0 = \frac{\theta n\ell + 2\tau + 2\sqrt{2\tau n\ell}}{2(1-\theta)} = L_\ell(n, \theta, \tau). \tag{27}$$

The value $(\Phi_\ell)_0$ thus serves as a constant upper bound for the proximity function $\Phi_\ell(v_+)$ throughout the execution of the algorithm.

4 Step Size Estimation

This section addresses two closely related tasks: determining an appropriate step size α, and analyzing the resulting decrease of the proximity (barrier) function. A damped Newton step produces the new iterate via

$$t_+ = t + \alpha\Delta t, \qquad r_+ = r + \alpha\Delta r, \qquad s_+ = s + \alpha\Delta s.$$

Using (8), this can be rewritten as

$$t_+ = t\left(e + \alpha\frac{\Delta t}{t}\right) = t\left(e + \alpha\frac{d_t}{v}\right) = \frac{t}{v}\left(v + \alpha d_t\right),$$
$$s_+ = s\left(e + \alpha\frac{\Delta s}{s}\right) = s\left(e + \alpha\frac{d_s}{v}\right) = \frac{s}{v}\left(v + \alpha d_s\right).$$

Hence $v_+ = \sqrt{\frac{t_+ s_+}{\mu}} = \sqrt{\left(v + \alpha d_t\right)\left(v + \alpha d_s\right)}$. For $\alpha \geq 0$, define $f(\alpha) = \Phi_\ell(v_+) - \Phi_\ell(v)$; for a fixed μ, $f(\alpha)$ represents the change in proximity between the current iterate and the new one. From (21) it follows that

$$\Phi_\ell(v_+) = \Phi_\ell\left(\sqrt{\left(v + \alpha d_t\right)\left(v + \alpha d_s\right)}\right) \leq \frac{1}{2}\left(\Phi_\ell(v + \alpha d_t) + \Phi_\ell(v + \alpha d_s)\right),$$

so that $f(\alpha) \leq f_1(\alpha)$, where

$$f_1(\alpha) = \frac{1}{2}\left(\Phi_\ell(v + \alpha d_t) + \Phi_\ell(v + \alpha d_s)\right) - \Phi_\ell(v). \tag{28}$$

Note that $f(0) = f_1(0) = 0$. To examine the behavior of $f_1(\alpha)$ near $\alpha = 0$, we compute its first two derivatives:

$$f_1^{'}(\alpha) = \sum_{i=1}^{n}\left(\varphi_\ell^{'}(v_i + \alpha d_{t_i})d_{t_i} + \varphi_\ell^{'}(v_i + \alpha d_{s_i})d_{s_i}\right),$$
$$f_1^{''}(\alpha) = \sum_{i=1}^{n}\left(\varphi_\ell^{''}(v_i + \alpha d_{t_i})d_{t_i}^2 + \varphi_\ell^{''}(v_i + \alpha d_{s_i})d_{s_i}^2\right).$$

Using (8) together with (14), we obtain

$$f_1^{'}(0) = \frac{1}{2}\left\langle \nabla\Phi_\ell(v),\, d_t + d_s\right\rangle = -\frac{1}{2}\langle \nabla\Phi_\ell(v),\, \nabla\Phi_\ell(v)\rangle = -2\sigma(v)^2.$$

For brevity we write $v_1 = \min(v)$, $\sigma = \sigma(v)$, $\Phi_\ell = \Phi_\ell(v)$.

Lemma 6. Let $\sigma(v)$ be as defined in (14). Then

$$\sigma(v) \geq \sqrt{\frac{\Phi_\ell(v)}{2}}. \tag{29}$$

Proof. Using (23), we have

$$\Phi_\ell(v) = \sum_{i=1}^{n} \varphi_\ell(v_i) \leq \sum_{i=1}^{n} \frac{1}{2}\left[\varphi_\ell^{'}(v_i)\right]^2 = \frac{1}{2}\|\nabla\Phi_\ell(v)\|^2 = 2\sigma(v)^2,$$

so $\sigma(v) \geq \sqrt{\frac{1}{2}\Phi_\ell(v)}$, which completes the proof. □

Remark 1. Suppose $\Phi_\ell(v) \geq \tau$, and recall the standing assumption $\tau \geq 1$. Applying Lemma 6 then immediately gives $\sigma(v) \geq \sqrt{\frac{1}{2}}$.

Following the same line of argument as Lemma 4.1 in [3], Lemma 4.2 in [4], Lemma 5.3 in [23], and Lemma 4.6 in [10], we obtain Lemmas 7–10 below, using the fact that $\varphi_\ell(t)$ is a kernel function and that $\varphi_\ell^{''}(t)$ is monotonically decreasing.

Lemma 7. [3] With $f_1(\alpha)$ and $\sigma(v)$ as in (28) and (14), respectively,

$$f_1^{''}(\alpha) \leq 2\sigma^2\varphi_\ell^{''}(v_1 - 2\alpha\sigma). \tag{30}$$

By the convexity of $f_1(\alpha)$, $f_1^{'}(\alpha) \leq 0$ for every α not exceeding the value at which $f_1(\alpha)$ attains its minimum, and conversely. As an immediate consequence of Lemma 7, we obtain the following three results.

Lemma 8. [4] The condition $f_1^{'}(\alpha) \leq 0$ certainly holds if α satisfies

$$\varphi_\ell^{'}(v_1) - \varphi_\ell^{'}(v_1 - 2\alpha\sigma) \leq 2\sigma. \tag{31}$$

Lemma 9. [23] The maximum step size $\overline{\alpha}$ satisfying (31) is given by

$$\overline{\alpha} = \frac{\varrho(\sigma) - \varrho(2\sigma)}{2\sigma}. \tag{32}$$

Lemma 10. [10] Let $\overline{\alpha}$ be as in Lemma 9. Then

$$\overline{\alpha} \geq \frac{1}{\varphi_\ell^{''}\left(\varrho(2\sigma)\right)}. \tag{33}$$

We can now establish the following lemma.

Lemma 11. Let $\bar{\alpha}$ be as in Lemma 9 (cf. Lemma 10), and let ϱ be the function introduced before Lemma 3. Under the condition $\Phi_\ell(v) \geq \tau$ for some $\tau \geq 1$, we have

$$\overline{\alpha} \geq \frac{1}{\ell + \sum_{j=1}^{\ell} u^j \left[\frac{4\sigma+\ell}{\ell}\right]^{\frac{uj+1}{u^\ell}}}.$$

Proof. Using Lemma 10 together with (16) and (26), we obtain

$$\begin{aligned}\overline{\alpha} &\geq \frac{1}{\varphi_\ell''\left(\varrho(2\sigma)\right)}\\ &= \frac{1}{\ell+\sum_{j=1}^{\ell} u^j\left(\varrho(2\sigma)\right)^{-u^j-1}}\\ &\geq \frac{1}{\ell+\sum_{j=1}^{\ell} u^j\left[\frac{4\sigma+\ell}{\ell}\right]^{\frac{u^j+1}{u^\ell}}}.\end{aligned}$$

This proves the claim. □

Let us denote

$$\widetilde{\alpha} = \frac{1}{\ell+\sum_{j=1}^{\ell} u^j\left[\frac{4\sigma+\ell}{\ell}\right]^{\frac{u^j+1}{u^\ell}}}. \tag{34}$$

Here $\widetilde{\alpha}$ plays the role of the default step size, and by Lemma 11, $\widetilde{\alpha} \leq \overline{\alpha}$.

Lemmas 12 and 13 below follow from Lemma 3.12 of [22] and Lemma 4.5 of [3], respectively.

Lemma 12. [22] Let g be a twice differentiable convex function with $g(0) = 0$ and $g'(0) < 0$, attaining its global minimum at some point $x^* > 0$, and suppose its second derivative $g''(x)$ is nondecreasing on $[0, x^*]$. Then

$$g(x) \leq \frac{xg'(0)}{2}, \qquad 0 \leq x \leq x^*.$$

Lemma 13. [3] Let $\overline{\alpha}$ be the largest step size defined in Lemma 9. If $\alpha \leq \overline{\alpha}$, then

$$f(\alpha) \leq -\alpha\sigma^2.$$

The following result plays a key role in the complexity analysis.

Lemma 14. Assume $\Phi_\ell(v) \geq 1$, and let $\widetilde{\alpha}$ be as in (34). Then

$$f(\widetilde{\alpha}) \leq -\frac{1}{36\sqrt{2}\,\ell\,(1+u^\ell)}\left[\Phi_\ell(v)\right]^{\frac{u^\ell-1}{2u^\ell}}. \tag{35}$$

Proof. Since $\Phi_\ell(v) \geq 1$, (29) gives

$$\sigma \geq \sqrt{\tfrac{1}{2}\Phi_\ell(v)} \geq \sqrt{\tfrac{1}{2}}.$$

Applying Lemma 13 with $\alpha = \widetilde{\alpha}$ and using (34), we obtain

$$f(\widetilde{\alpha}) \leq -\widetilde{\alpha}\sigma^2 = -\frac{\sigma^2}{\ell+\sum_{j=1}^{\ell} u^j\left[\frac{4\sigma+\ell}{\ell}\right]^{\frac{u^j+1}{u^\ell}}}$$

$$
\begin{aligned}
&\leq -\frac{\sigma^2}{\ell(2\sigma)+\sum_{j=1}^{\ell} u^j \left[\frac{2(2\sigma)+\ell(2\sigma)}{\ell}\right]^{\frac{u^j+1}{u^\ell}}} && \text{(since } 2\sigma \geq 1\text{)}\\
&= -\frac{\sigma^2}{\ell(2\sigma)+\sum_{j=1}^{\ell} u^j \left[\frac{2+\ell}{\ell}\right]^{\frac{u^j+1}{u^\ell}} (2\sigma)^{\frac{u^j+1}{u^\ell}}}\\
&\leq -\frac{\sigma^2}{\ell(2\sigma)^{\frac{u^\ell+1}{u^\ell}}+\sum_{j=1}^{\ell} u^j \left[\frac{\ell+2}{\ell}\right]^{\frac{u^j+1}{u^\ell}} (2\sigma)^{\frac{u^\ell+1}{u^\ell}}} && \text{(since } u^j \leq u^\ell,\ 2\sigma \geq 1\text{)}\\
&= -\frac{\sigma^2}{2^{\frac{u^\ell+1}{u^\ell}}\left(\ell+\sum_{j=1}^{\ell} u^j \left[\frac{\ell+2}{\ell}\right]^{\frac{u^j+1}{u^\ell}}\right)\sigma^{\frac{u^\ell+1}{u^\ell}}}\\
&\leq -\frac{\sigma^{2-\frac{u^\ell+1}{u^\ell}}}{2^{\frac{u^\ell+1}{u^\ell}}\left(\ell+\sum_{j=1}^{\ell} u^j \left[\frac{\ell+2}{\ell}\right]^{\frac{u^\ell+1}{u^\ell}}\right)}\\
&\leq -\frac{\sigma^{\frac{u^\ell-1}{u^\ell}}}{2^2\left(\ell+\sum_{j=1}^{\ell} u^j \left[\frac{\ell+2\ell}{\ell}\right]^2\right)} && \text{(since } \tfrac{u^\ell+1}{u^\ell} \leq 2,\ \tfrac{\ell+2}{\ell} \leq \tfrac{3\ell}{\ell} = 3\text{)}\\
&= -\frac{\sigma^{\frac{u^\ell-1}{u^\ell}}}{4\left(\ell+9\sum_{j=1}^{\ell} u^j\right)}\\
&\leq -\frac{\sigma^{\frac{u^\ell-1}{u^\ell}}}{4\left(\ell+9\ell u^\ell\right)} && \text{(since } u^j \leq u^\ell \text{ for } j \leq \ell\text{)}\\
&\leq -\frac{\sigma^{\frac{u^\ell-1}{u^\ell}}}{36\ell\left(1+u^\ell\right)} && \text{(since } 4\ell + 36\ell u^\ell \leq 36\ell + 36\ell u^\ell\text{)}\\
&= -\frac{1}{36\ell\left(1+u^\ell\right)} \cdot \frac{1}{2^{\frac{u^\ell-1}{2u^\ell}}} \left[\Phi_\ell(v)\right]^{\frac{u^\ell-1}{2u^\ell}} && \text{(since } \sigma \geq \sqrt{\Phi_\ell(v)/2}\text{)}\\
&\leq -\frac{1}{36\sqrt{2}\,\ell\left(1+u^\ell\right)} \left[\Phi_\ell(v)\right]^{\frac{u^\ell-1}{2u^\ell}} && \text{(since } 2^{\frac{u^\ell-1}{2u^\ell}} \leq \sqrt{2}\text{).}
\end{aligned}
$$

This concludes the proof. □

5 Complexity Analysis of the Algorithm

Updating μ to $(1-\theta)\mu$ and invoking Lemma 5 gives

$$\Phi_\ell(v_+) \leq (\Phi_\ell)_0 = \frac{\theta n\ell + 2\tau + 2\sqrt{2\tau n\ell}}{2(1-\theta)} = L_\ell\left(n, \theta, \tau\right).$$

To guarantee that $\Phi_\ell(v_+) \leq \tau$ after each reduction of μ, we must bound the number of inner iterations needed. Immediately after updating μ, the value of $\Phi_\ell(v)$ is denoted $(\Phi_\ell)_0$; the values

produced by the subsequent inner iterations within the same outer iteration are denoted $(\Phi_\ell)_k$, $k = 1, 2, \ldots, K$, where K is the total number of inner iterations performed. The decrease achieved at each inner iteration is given by (35). Following [3], we set

$$\kappa = \frac{1}{36\sqrt{2}\ell\,(1+u^\ell)}, \qquad \gamma = 1 - \frac{u^\ell - 1}{2u^\ell} = \frac{u^\ell + 1}{2u^\ell} \in]0, 1].$$

Lemma 15. Let K denote the total number of inner iterations performed during one outer iteration. Then

$$K \leq \left(72\sqrt{2}\ell u^\ell\right) \left[(\Phi_\ell)_0\right]^{\frac{u^\ell+1}{2u^\ell}}.$$

Proof. By Lemma 1.3.2 of [22], $K \leq \dfrac{[(\Phi_\ell)_0]^\gamma}{\kappa\gamma} = \left(72\sqrt{2}\ell u^\ell\right) \left[(\Phi_\ell)_0\right]^{\frac{u^\ell+1}{2u^\ell}}$, which proves the claim. □

By [24, Lemma II.17, p. 116], the number of outer iterations is bounded above by $\dfrac{\log\left(\frac{n}{\epsilon}\right)}{\theta}$. Multiplying this bound by the bound on the number of inner iterations from Lemma 15 yields the following upper bound on the total number of iterations:

$$\left(72\sqrt{2}\ell u^\ell\right) \left[(\Phi_\ell)_0\right]^{\frac{u^\ell+1}{2u^\ell}} \frac{\log\left(\frac{n}{\epsilon}\right)}{\theta}. \tag{36}$$

Large-update methods

For (L-UP) methods, taking $\tau = \mathcal{O}(n)$ and $\theta = \Theta(1)$ in $(\Phi_\ell)_0$ gives $(\Phi_\ell)_0 = \mathcal{O}(n)$, and substituting into (36) yields the iteration-complexity bound

$$\mathcal{O}\left(u^\ell n^{\frac{u^\ell+1}{2u^\ell}} \log\left(\frac{n}{\epsilon}\right)\right).$$

Small-update methods

For (S-UP) methods we take $\tau = \mathcal{O}(1)$ and $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$. Substituting these choices directly into the expression for $(\Phi_\ell)_0$ used above does not give the tightest possible bound; a sharper estimate, which we continue to denote $(\Phi_\ell)_0$ (now specific to this regime), can be obtained as follows.

By (24), $\varphi_\ell(x) \leq \frac{1}{2}\left[\frac{(u-1)\ell + u\left(u^\ell - 1\right)}{u-1}\right](x-1)^2$ for $x > 1$. Combining this with Lemma 4 (taking $\beta = 1/\sqrt{1-\theta}$) gives

$$\Phi_\ell(v_+) \leq n\varphi_\ell\left(\frac{1}{\sqrt{1-\theta}}\varsigma\left(\frac{\Phi_\ell(v)}{n}\right)\right)$$

$$\leq \frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)}\left(\frac{1}{\sqrt{1-\theta}}\varsigma\left(\frac{\Phi_\ell(v)}{n}\right) - 1\right)^2$$
$$= \frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\varsigma\left(\frac{\Phi_\ell(v)}{n}\right) - \sqrt{1-\theta}\right)^2.$$

Using (25), we continue:

$$\frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\varsigma\left(\frac{\Phi_\ell(v)}{n}\right) - \sqrt{1-\theta}\right)^2$$
$$\leq \frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\left(1 + \sqrt{\frac{2\Phi_\ell(v)}{n\ell}}\right) - \sqrt{1-\theta}\right)^2$$
$$= \frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\left(1 - \sqrt{1-\theta}\right) + \sqrt{\frac{2\Phi_\ell(v)}{n\ell}}\right)^2$$
$$\leq \frac{n\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\theta + \sqrt{\frac{2\tau}{n\ell}}\right)^2$$
$$= \frac{\left((u-1)\ell + u\left(u^\ell - 1\right)\right)}{2(u-1)(1-\theta)}\left(\theta\sqrt{n} + \sqrt{\frac{2\tau}{\ell}}\right)^2 = (\Phi_\ell)_0\,,$$

where in the last two steps we used the identity $1 - \sqrt{1-\theta} = \dfrac{\theta}{1+\sqrt{1-\theta}} \leq \theta$ together with $\Phi_\ell(v) \leq \tau$, the latter applied as an upper bound on the initial value $(\Phi_\ell)_0$.

With $\tau = \mathcal{O}(1)$ and $\theta = \Theta(1/\sqrt{n})$, we have $\theta\sqrt{n} = \Theta(1)$ and $\sqrt{2\tau/\ell} = O(1)$, so $(\Phi_\ell)_0 = \mathcal{O}\left(u^\ell\right)$ as $n \to \infty$. Since $\gamma \leq 1$ and $(\Phi_\ell)_0 \geq 1$, we have $[(\Phi_\ell)_0]^\gamma \leq (\Phi_\ell)_0 = \mathcal{O}(u^\ell)$, and substituting into (36) together with $\log(n/\epsilon)/\theta = \Theta(\sqrt{n}\log(n/\epsilon))$ gives the iteration-complexity bound

$$\mathcal{O}\left(u^{2\ell}\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right).$$

Remark 2. For $\ell = 1$, the analysis above remains valid and yields $\mathcal{O}\left(un^{\frac{u+1}{2u}}\log\left(\frac{n}{\epsilon}\right)\right)$ for the (L-UP) case, and $\mathcal{O}\left(u^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$ for the (S-UP) case.

Remark (Optimal Parameter Choice)

We now show how to tune u as a function of n and ℓ so as to recover the best-known complexity bound for (L-UP) methods. Throughout, log denotes the natural logarithm, so that $n^{1/\log n} = e$.

From the L-UP bound derived above,

$$\mathcal{O}\left(u^\ell n^{\frac{u^\ell+1}{2u^\ell}}\log\left(\frac{n}{\epsilon}\right)\right),$$

choose

$$u = \left(\frac{\log n}{2}\right)^{\frac{1}{\ell}}, \qquad \text{so that} \qquad u^{\ell} = \frac{\log n}{2}.$$

Then

$$n^{\frac{\frac{\log n}{2}+1}{\log n}} = n^{\frac{1}{2}+\frac{1}{\log n}} = \sqrt{n} \cdot n^{1/\log n} = \sqrt{n} \cdot e,$$

and substituting into the bound gives

$$u^{\ell} n^{\frac{u^{\ell}+1}{2u^{\ell}}} = \frac{\log n}{2} \cdot \sqrt{n} \cdot e.$$

Hence the complexity bound becomes

$$\mathcal{O}\left(\frac{\log n}{2} \cdot \sqrt{n} \cdot e \cdot \log\left(\frac{n}{\epsilon}\right)\right) = \mathcal{O}\left(\sqrt{n} \log n \log\left(\frac{n}{\epsilon}\right)\right),$$

which is the best complexity bound currently known for (L-UP) interior-point methods.

Table 1: The properties (19), (20), and (22).

Kernel functions	$\varphi_p(x)$	$\varphi_\ell(x)$
$\varphi_i^{'}(x)$	$x - x^{-p}$	$\ell x - \sum_{j=1}^{\ell} x^{-u^j}$
$\varphi_i^{''}(x)$	$1 + px^{-p-1}$	$\ell + \sum_{j=1}^{\ell} u^j x^{-u^j-1}$
$\varphi_i^{'''}(x)$	$-p(p+1)x^{-p-2}$	$-\sum_{j=1}^{\ell} u^j \left(u^j+1\right) x^{-u^j-2}$
$x\varphi_i^{''}(x) + \varphi_i^{'}(x)$	$2x + (p-1)x^{-p}$	$2\ell x + \sum_{j=1}^{\ell} \left(u^j - 1\right) x^{-u^j}$
$x\varphi_i^{''}(x) - \varphi_i^{'}(x)$	$(p-1)x^{-p}$	$\sum_{j=1}^{\ell} \left(1+u^j\right) x^{-u^j}$

6 Theoretical Comparison of Kernel Properties

To assess the practical relevance of the proposed (N-K-F) and its effect on algorithmic behavior, we compare it with the self-regular kernel of Peng et al. [22] and the resulting performance of the corresponding IPM schemes.

1. $\varphi_p(x)$, the kernel function introduced by Peng et al. [22]:

$$\varphi_p(x) = \frac{x^2-1}{2} + \frac{x^{1-p}-1}{p-1}, \qquad p > 1.$$

2. $\varphi_\ell(x)$, our (N-K-F), as defined in (1):

$$\varphi_\ell(x) = \frac{\ell\left(x^2-1\right)}{2} - \sum_{j=1}^{\ell} \frac{x^{1-u^j}-1}{1-u^j}, \qquad u > 1, \qquad \ell \in \mathbb{N} \setminus \{0\}.$$

The comparison is summarized in Tables 1, 2, and 3 below. In what follows, the generic subscript i in $\varphi_i'(x), \varphi_i''(x), \varphi_i'''(x)$ stands for either kernel ($i = p$ or $i = \ell$), to avoid repeating the same expressions twice.

Table 2: The estimated bound terms.

(K-F)	$\varphi_p(x)$	$\varphi_\ell(x)$
$\varsigma(s)$ for $s \geq 0$	$(1+2s)^{\frac{1}{2}} \leq \varsigma(s) \leq \left(2s+\frac{p+1}{p-1}\right)^{\frac{1}{2}}$	$1+\sqrt{\frac{2(u-1)s}{(u-1)\ell+u(u^\ell-1)}} \leq \varsigma(s) \leq 1+\sqrt{\frac{2s}{\ell}}$
$\varrho(z)$ for $z \geq 0$	$\varrho(z) \geq \frac{1}{(2z+1)^{\frac{1}{p}}}$	$\varrho(z) > \left[\frac{\ell}{2z+\ell}\right]^{\frac{1}{u^\ell}}$
$L(n,\theta,\tau) \leq$	$\frac{2\tau+\frac{p+1}{p-1}n}{2(1-\theta)}$	$\frac{\theta n\ell+2\tau+2\sqrt{2\tau n\ell}}{2(1-\theta)}$
$\widetilde{\alpha}$	$\frac{1}{1+p(1+4\sigma)^{\frac{1+p}{p}}}$	$\frac{1}{\ell+\sum_{j=1}^{\ell} u^j\left[\frac{4\sigma+\ell}{\ell}\right]^{\frac{u^j+1}{u^\ell}}}$
κ	$\frac{1}{100(p+1)}$	$\frac{1}{36\sqrt{2\ell}(1+u^\ell)}$
γ	$\frac{p+1}{2p}$	$\frac{u^\ell+1}{2u^\ell}$

Table 3: Complexity findings concerning (L-UP) and (S-UP) algorithms.

Complexity bound	$\varphi_p(x)$	$\varphi_\ell(x)$
(L-UP) methods	$\mathcal{O}\left(pn^{\frac{p+1}{2p}}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathcal{O}\left(u^\ell n^{\frac{u^\ell+1}{2u^\ell}}\log\left(\frac{n}{\epsilon}\right)\right)$
(S-UP) methods	$\mathcal{O}\left(p^2\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$	$\mathcal{O}\left(u^{2\ell}\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$

7 Numerical Tests

We compare five kernel functions, including the one proposed in this paper, summarized in Table 4. For completeness, the table also lists the very recent hyperbolic kernel of Bouhenache et al. [7]; this kernel is included only for context and is *not* part of the numerical comparison below, since reproducing it would require implementation details beyond the scope of the present study.

Table 4: Considered kernel functions.

Kernel function	**Complexity**	**References**
$\varphi_p(x)=\frac{x^2-1}{2}+\frac{x^{1-p}-1}{p-1}, p>1$	$\mathcal{O}\left(pn^{\frac{p+1}{2p}}\log\left(\frac{n}{\epsilon}\right)\right)$	[22]
$\varphi_q(x)=\frac{x^2-1-\log x}{2}+\frac{x^{1-q}-1}{2(q-1)}, q>1$	$\mathcal{O}\left(qn^{\frac{q+1}{2q}}\log\frac{n}{\epsilon}\right)$	[6]
$\varphi_r(x)=(r+2)x^2-(r+3)x-\log x+\frac{1}{x^r}, r\geq 2$	$\mathcal{O}\left((r+1)^{\frac{r+2}{2(r+1)}}n^{\frac{r+2}{2(r+1)}}\log\frac{n}{\epsilon}\right)$	[9]
$\varphi_a(x)=\frac{x^2-1}{2}-\int_1^x a^{\left(\frac{1}{t}-1\right)}dt, a\geq e$	/	[8]
$\phi_h(x)$ (hyperbolic kernel, for context only)	$\mathcal{O}(\sqrt{n}\log n\log(n/\epsilon))$	[7]
$\varphi_\ell(x)=\frac{\ell\left(x^2-1\right)}{2}-\sum_{j=1}^{\ell}\frac{x^{1-u^j}-1}{1-u^j}, u>1, \ell\in\mathbb{N}\backslash\{0\}.$	$\mathcal{O}\left(u^\ell n^{\frac{u^\ell+1}{2u^\ell}}\log\left(\frac{n}{\epsilon}\right)\right)$	†

† This kernel function is proposed for the first time in the present work; hence, no external reference applies.

7.1 Experimental Setup

All numerical experiments were performed on a computer with an Intel Core i7-10750H processor (2.60 GHz) and 16 GB of RAM. The stopping criterion is

$$\frac{x^T s}{\mu} \leq 10^{-8},$$

where x and s are the primal and dual variables, respectively, and μ is the barrier parameter. A maximum of 2000 iterations was imposed for problems of size $(m, n) \leq 100$, and 1500 iterations for larger problems, together with a time limit of 3600 seconds (1 hour) per instance. If the algorithm does not converge within this time limit, the corresponding entry in the tables is marked "?", indicating a timeout. No numerical failures or stability issues were observed; all timeouts occurred solely because of the high per-iteration cost for the largest problem instances.

Default Step Sizes in the Numerical Experiments

The step sizes α_p and α_ℓ used in Tables 5 and 6 are *not* free tuning parameters; they are the analytic default step sizes derived from Lemma 11 (for our kernel) and the corresponding lemma for Peng's kernel [22], namely

$$\alpha_p = \frac{1}{1 + p(1+4\sigma)^{\frac{p+1}{p}}}, \qquad \alpha_\ell = \frac{1}{\ell + \sum_{j=1}^{\ell} u^j \left(\frac{4\sigma+\ell}{\ell}\right)^{\frac{u^j+1}{u^\ell}}}.$$

These expressions guarantee a sufficient decrease of the proximity function (Lemma 14) and were used as-is, with no further heuristic adjustment, to obtain reproducible, theory-compliant numerical illustrations.

Test Problem

We consider the following problem with $n = 2m$:

$$M(i,j) = \begin{cases} 0 & \text{if } i \neq j \text{ and } j \neq i+m, \\ 1 & \text{if } i = j \text{ or } j = i+m, \end{cases}$$

with $c(i) = -1$, $c(i+m) = 0$, $b(i) = 2$, and the interior-point starting condition

We note that for several of the larger, successfully solved instances (e.g. the inner-iteration counts of φ_q and φ_r in Tables 5 and 6), the reported totals exceed these nominal iteration caps; this indicates that, in practice, termination was governed by the duality-gap criterion above rather than by the iteration cap, which therefore acted only as a safety bound and was not the binding constraint in those cases.

$$t^0(i) = t^0(i+m) = 1, \quad r^0(i) = -2, \quad s^0(i) = 1, \quad s^0(i+m) = 2, \qquad i = 1, \ldots, m.$$

All computations were carried out in MATLAB, with $\mu^0 = 1$, $\theta \in \{0.01, 0.99\}$, $\epsilon = 10^{-4}$, $\tau = n$, and $u = 2$, $\ell = 2$ for the (N-K-F). The step size α was chosen at each iteration to satisfy $0 < \alpha < \overline{\alpha}$, using the default formulas α_p and α_ℓ given above.

In the result tables, (m, n) denotes the number of constraints and the number of variables, respectively. The column "Iterations (Inr,Outr)" reports the total number of inner (Newton) iterations followed by the number of outer iterations required to reach the optimal solution, while "Time (s)" reports the corresponding CPU time. Note that, since the outer loop depends only on θ, n, ϵ, and μ^0 (not on the kernel), the outer-iteration count is identical across all kernels for a given (m, n) and θ, as confirmed by the tables below. A summary of the experiments is given in Tables 5 and 6.

The symbol "?" indicates that the algorithm did not converge within the 3600-second time limit (timeout).

Note. For the same configuration φ_ℓ ($\ell = 2$, $u = 2$) at problem size $(500, 1000)$, the CPU time differs substantially between $\theta = 0.01$ (Table 5: 290.11 s, total iterations $5155 + 1714 = 6869$) and $\theta = 0.99$ (Table 6: 507.14 s, total iterations $11434 + 4 = 11438$). This θ-dependent difference is consistent with the roles of θ discussed in Section 2: small-update methods ($\theta = 0.01$) require many outer iterations but very few inner iterations per outer step, keeping the iterate close to the central path, whereas large-update methods ($\theta = 0.99$) require very few outer iterations but many more inner iterations to restore proximity after an aggressive reduction of μ, resulting in a higher cumulative cost in this instance.

Comments

The results in Tables 5 and 6 reveal several interesting patterns regarding the performance of the proposed kernel φ_ℓ ($\ell = 2, u = 2$) and its variant φ_ℓ ($\ell = 1, u = 3.5$), compared to existing kernels.

Observed trends. For the smallest instances, $(10, 20)$ and $(20, 40)$, all kernels perform reasonably well with comparable computation times. As the problem dimension grows, however, the differences between kernels become much more pronounced. The variant φ_ℓ ($\ell = 1, u = 3.5$) requires somewhat more inner iterations than the self-regular kernel φ_p ($p = 3$) on the three smallest instances, but this trend reverses from $(50, 100)$ onward, where φ_ℓ ($\ell = 1, u = 3.5$) consistently needs fewer inner iterations and less time than φ_p, the gap widening as the problem size increases. The variant φ_ℓ ($\ell = 2, u = 2$), by contrast, is consistently slower than both φ_p and φ_ℓ ($\ell = 1$) across all instances, reflecting the more conservative step size associated with $\ell = 2$; its main advantage lies elsewhere, namely in its robustness on the largest instances, dis-

Table 5: Comparative analysis of examples for $\theta = 0.01$

(m, n)	Function	Iterations (Inr,Outr)	Time (seconds)
(10,20)	φ_p $(p = 3)$	$(191, 1324)$	0.0362
	φ_q $(q = 2)$	$(9408, 1324)$	0.5240
	φ_r $(r = 1)$	$(1042, 1324)$	0.0949
	φ_a $(a = 3)$	$(1, 1324)$	25.1535
	φ_ℓ $(\ell = 2, u = 2)$	$(355, 1324)$	0.0579
	φ_ℓ $(\ell = 1, u = 3.5)$	$(216, 1324)$	0.0496
(20,40)	φ_p $(p = 3)$	$(289, 1393)$	0.0714
	φ_q $(q = 2)$	$(21045, 1393)$	1.6124
	φ_r $(r = 1)$	$(1751, 1393)$	0.2499
	φ_a $(a = 3)$	$(1, 1393)$	61.2205
	φ_ℓ $(\ell = 2, u = 2)$	$(555, 1393)$	0.1190
	φ_ℓ $(\ell = 1, u = 3.5)$	$(303, 1393)$	0.0947
(25,50)	φ_p $(p = 3)$	$(332, 1415)$	0.1497
	φ_q $(q = 2)$	$(27391, 1415)$	2.9428
	φ_r $(r = 1)$	$(2070, 1415)$	0.3524
	φ_a $(a = 3)$	$(1, 1415)$	81.0503
	φ_ℓ $(\ell = 2, u = 2)$	$(643, 1415)$	0.1865
	φ_ℓ $(\ell = 1, u = 3.5)$	$(340, 1415)$	0.1593
(50,100)	φ_p $(p = 3)$	$(522, 1484)$	0.3614
	φ_q $(q = 2)$	$(62737, 1484)$	17.895
	φ_r $(r = 1)$	$(3479, 1484)$	1.0906
	φ_a $(a = 3)$	$(1, 1484)$	261.07
	φ_ℓ $(\ell = 2, u = 2)$	$(1025, 1484)$	0.5122
	φ_ℓ $(\ell = 1, u = 3.5)$	$(492, 1484)$	0.3262
(100,200)	φ_p $(p = 3)$	$(835, 1553)$	2.1907
	φ_q $(q = 2)$	$(145343, 1553)$	327.63
	φ_r $(r = 1)$	$(5850, 1553)$	12.653
	φ_a $(a = 3)$	$(1, 1553)$	525.35
	φ_ℓ $(\ell = 2, u = 2)$	$(1652, 1553)$	3.7866
	φ_ℓ $(\ell = 1, u = 3.5)$	$(726, 1553)$	1.8244
(250,500)	φ_p $(p = 3)$	$(1590, 1645)$	23.131
	φ_q $(q = 2)$	$(?, ?)$	?
	φ_r $(r = 1)$	$(?, ?)$	?
	φ_a $(a = 3)$	$(1, 1645)$	1334.7
	φ_ℓ $(\ell = 2, u = 2)$	$(3147, 1645)$	46.813
	φ_ℓ $(\ell = 1, u = 3.5)$	$(1243, 1645)$	16.462
(500,1000)	φ_p $(p = 3)$	$(2620, 1714)$	130.07
	φ_q $(q = 2)$	$(?, ?)$	?
	φ_r $(r = 1)$	$(?, ?)$	?
	φ_a $(a = 3)$	$(?, ?)$	?
	φ_ℓ $(\ell = 2, u = 2)$	$(5155, 1714)$	290.11
	φ_ℓ $(\ell = 1, u = 3.5)$	$(1891, 1714)$	92.320

Table 6: Comparative analysis of examples for $\theta = 0.99$

(m, n)	Function	Iterations (Inr,Outr)	Time (seconds)
(10,20)	φ_p $(p = 3)$	$(247, 3)$	0.0188
	φ_q $(q = 2)$	$(11231, 3)$	0.4454
	φ_r $(r = 1)$	$(1912, 3)$	0.1892
	φ_a $(a = 3)$	$(1, 3)$	0.1386
	φ_ℓ $(\ell = 2, u = 2)$	$(529, 3)$	0.0460
	φ_ℓ $(\ell = 1, u = 3.5)$	$(282, 3)$	0.0278
(20,40)	φ_p $(p = 3)$	$(418, 4)$	0.0435
	φ_q $(q = 2)$	$(25363, 4)$	1.5245
	φ_r $(r = 1)$	$(3802, 4)$	0.3845
	φ_a $(a = 3)$	$(1, 4)$	0.3937
	φ_ℓ $(\ell = 2, u = 2)$	$(1030, 4)$	0.2245
	φ_ℓ $(\ell = 1, u = 3.5)$	$(458, 4)$	0.0570
(25,50)	φ_p $(p = 3)$	$(479, 4)$	0.0768
	φ_q $(q = 2)$	$(33000, 4)$	2.8559
	φ_r $(r = 1)$	$(4140, 4)$	0.5617
	φ_a $(a = 3)$	$(1, 4)$	0.5042
	φ_ℓ $(\ell = 2, u = 2)$	$(1217, 4)$	0.2977
	φ_ℓ $(\ell = 1, u = 3.5)$	$(512, 4)$	0.0940
(50,100)	φ_p $(p = 3)$	$(747, 4)$	0.2133
	φ_q $(q = 2)$	$(75556, 4)$	17.208
	φ_r $(r = 1)$	$(6952, 4)$	1.9228
	φ_a $(a = 3)$	$(1, 4)$	2.2999
	φ_ℓ $(\ell = 2, u = 2)$	$(2049, 4)$	0.9675
	φ_ℓ $(\ell = 1, u = 3.5)$	$(740, 4)$	0.3176
(100,200)	φ_p $(p = 3)$	$(1189, 4)$	2.2764
	φ_q $(q = 2)$	$(175052, 4)$	315.46
	φ_r $(r = 1)$	$(12685, 4)$	30.847
	φ_a $(a = 3)$	$(1, 4)$	2.6967
	φ_ℓ $(\ell = 2, u = 2)$	$(3452, 4)$	8.4463
	φ_ℓ $(\ell = 1, u = 3.5)$	$(1091, 4)$	2.3974
(250,500)	φ_p $(p = 3)$	$(2248, 4)$	26.167
	φ_q $(q = 2)$	$(?, ?)$	?
	φ_r $(r = 1)$	$(25155, 4)$	343.45
	φ_a $(a = 3)$	$(1, 4)$	5.0024
	φ_ℓ $(\ell = 2, u = 2)$	$(6840, 4)$	103.43
	φ_ℓ $(\ell = 1, u = 3.5)$	$(1865, 4)$	24.383
(500,1000)	φ_p $(p = 3)$	$(3693, 4)$	161.43
	φ_q $(q = 2)$	$(?, ?)$	?
	φ_r $(r = 1)$	$(?, ?)$	?
	φ_a $(a = 3)$	$(1, 4)$	9.0605
	φ_ℓ $(\ell = 2, u = 2)$	$(11434, 4)$	507.14
	φ_ℓ $(\ell = 1, u = 3.5)$	$(2837, 4)$	140.12

cussed below. The kernel φ_q ($q = 2$) exhibits extremely high iteration counts on the instances it manages to solve (e.g. 175052 inner iterations for $(100, 200)$ at $\theta = 0.99$) and fails to converge within the time limit on the two largest instances, $(250, 500)$ and $(500, 1000)$, indicating poor scalability. The kernel φ_r ($r = 1$) behaves reasonably on small instances but becomes increasingly expensive at scale, reaching 25155 inner iterations for $(250, 500)$ at $\theta = 0.99$, and times out altogether on $(500, 1000)$.

When the proposed method is preferable. Our kernel functions are particularly advantageous on medium-to-large instances. For example, at $(500, 1000)$ with $\theta = 0.99$ (Table 6), φ_ℓ ($\ell = 1, u = 3.5$) completes in 140.12 seconds with 2837 inner iterations, against 161.43 seconds and 3693 inner iterations for φ_p ($p = 3$). The kernel φ_a ($a = 3$) is notable in that it requires only a single inner (Newton) iteration in total across the entire run, regardless of problem size or θ — its outer-iteration count matches that of the other kernels (e.g. 1324 at $\theta = 0.01$, 3 at $\theta = 0.99$ for $(10, 20)$) — yet its computation time still grows steadily with problem size (from 0.1386 s at $(10, 20)$ to 9.0605 s at $(500, 1000)$ for $\theta = 0.99$), suggesting that the per-outer-iteration cost, rather than the iteration count itself, dominates its running time. Our kernel achieves a more balanced trade-off between the number of inner and outer iterations, which translates into competitive overall performance, particularly for the $\ell = 1$ variant.

Cases with limited advantage. On the smallest instances, $(10, 20)$ and $(20, 40)$, the differences between kernels are modest, and the simpler kernel φ_p remains a fully adequate choice; here φ_ℓ ($\ell = 1$) offers no clear iteration-count advantage, and φ_ℓ ($\ell = 2$) is noticeably slower. The added flexibility of the multi-parameter kernel is most useful at larger scale; for small or moderate problems where extreme accuracy is not required, the extra parameter ℓ mainly adds tuning effort without a clear practical benefit over the simpler $\ell = 1$ variant.

Scalability and robustness. A key observation is the robustness of the proposed kernel family: while φ_q fails to converge at $(500, 1000)$ and on parts of $(250, 500)$, and φ_r times out at $(500, 1000)$, both variants of φ_ℓ successfully solve every test instance within the time limit, with computation time growing only moderately with problem size. Among the two, the $\ell = 1$, $u = 3.5$ variant offers the best overall practical performance, consistently outperforming φ_p in both iteration count and computation time on medium-to-large instances, while the $\ell = 2, u = 2$ variant trades additional iterations for the broader theoretical flexibility of the multi-exponent construction.

We note that the kernel φ_r is analytically defined for $r \geq 2$ in [9], whereas the experiments here use $r = 1$; this parameter choice falls outside the range covered by the original analysis and should be interpreted as an empirical comparison point rather than a theoretically guaranteed regime.

8 Conclusion

In this work, we introduced a novel multi-parameter kernel function $\varphi_\ell(x)$ that generates effective search directions and improves the theoretical complexity of primal–dual interior-point methods for linear optimization. By summing power terms with geometrically spaced exponents $u, u^2, \ldots, u^\ell$, the proposed kernel decouples the local curvature near the central path from the asymptotic growth far from it — a structural flexibility unattainable with single-parameter kernels. Both large-update (L-UP) and small-update (S-UP) variants were rigorously analyzed, yielding complexity bounds of $\mathcal{O}\left(u^\ell n^{\frac{u^\ell+1}{2u^\ell}} \log\left(\frac{n}{\epsilon}\right)\right)$ and $\mathcal{O}\left(u^{2\ell}\sqrt{n}\log\left(\frac{n}{\epsilon}\right)\right)$, respectively. With the optimal tuning $u = \left(\frac{\log n}{2}\right)^{\frac{1}{\ell}}$, these bounds recover the best-known $\mathcal{O}\left(\sqrt{n}\log(n)\log\left(\frac{n}{\epsilon}\right)\right)$ result, confirming that self-regularity is sufficient but not necessary for optimal complexity. The numerical experiments support these theoretical findings, showing that the proposed kernel — particularly the $\ell = 1$ variant — is competitive with, and often more efficient than, classical kernels on medium-to-large instances, while remaining robust on problems where some existing kernels fail to converge within the allotted time.

Limitations: The present analysis is restricted to standard-form linear optimization under the interior-point condition (IPC). Extensions to infeasible starting points, degenerate problems, or problems with special structure remain unaddressed. The selection of u and ℓ is currently guided by theoretical bounds rather than by adaptive or data-driven strategies, and the interaction between ℓ and u in determining step-size conservatism — visible, for instance, in the slower performance of the $\ell = 2$ variant relative to $\ell = 1$ in our experiments — warrants further investigation.

Future Directions: Natural extensions include: *(i)* generalization to semidefinite optimization (SDO) and second-order cone programming (SOCP); *(ii)* development of infeasible-start variants compatible with homogeneous self-dual embeddings; *(iii)* adaptive parameter-selection strategies for u and ℓ based on problem structure; *(iv)* integration with preconditioning techniques for large-scale sparse LP; and *(v)* theoretical and empirical analysis for non-symmetric cones.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The author conducted this research without any funding, grants, or support.

Conflict of Interest

The author declares that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Artificial Intelligence Statement

Artificial intelligence (AI) tools, including large language models, were used solely for language editing and improving readability. AI tools were not used for generating ideas, performing analyses, interpreting results, or writing the scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

[1] Amini, K., Haseli, A. (2007). "A new proximity function generating the best known iteration bounds for both large-update and small-update interior-point methods". *ANZIAM Journal*, 49(2), 259–270. https://doi.org/10.1017/S1446181100012827

[2] Amrane, H., Fateh, M. (2025). "An efficient parametric kernel function of IPMs for linear optimization problems". *Results in Control and Optimization*, 18, 100537. https://doi.org/10.1016/j.rico.2025.100537

[3] Bai, Y., El Ghami, M., Roos, K. (2004). "A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization". *SIAM Journal on Optimization*, 15(1), 101–128. https://doi.org/10.1137/S1052623403423114

[4] Bai, Y., El Ghami, M., Roos, K. (2002). "A new efficient large-update primal-dual interior-point method based on a finite barrier". *SIAM Journal on Optimization*, 13(3), 766–782. https://doi.org/10.1137/S1052623401398132

[5] Bai, Y.Q., Roos, C. (2003). "A polynomial-time algorithm for linear optimization based on a new simple kernel function". *Optimization Methods and Software*, 18(6), 631–646. https://doi.org/10.1080/10556780310001639735

[6] Bouafia, M., Benterki, D., Yassine, A. (2018). "An efficient parameterized logarithmic kernel function for linear optimization". *Optimization Letters*, 12(5), 1079–1097. https://doi.org/10.1007/s11590-017-1170-5

[7] Bouhenache, Y., Chikouche, W., Guerdouh, S. (2025). "An interior-point algorithm for LCP based on a parameterized hyperbolic kernel function". *Computational Mathematics and Mathematical Physics*, 65(6), 1181–1194. https://doi.org/10.1134/S096554252570054X

[8] Bounibane, B., Chalekh, R. (2024). "Kernel function-based primal-dual interior-point methods for linear optimization". *Palestine Journal of Mathematics*, 13(4), 261–277. https://api.semanticscholar.org/CorpusID:279239371

[9] Bounibane, B., Guemmaz, A., Djeffal, El. (2022). "Theoretical and numerical result for linear semidefinite programming based on a new kernel function". *Journal of Mathematical and Computational Science*, 12, 201. https://doi.org/10.28919/jmcs/7718

[10] Cho, G.M. (2011). "An interior point algorithm for linear optimization based on a new barrier function". *Applied Mathematics and Computation*, 218(2), 386–395. https://doi.org/10.1016/j.amc.2011.05.075

[11] El Ghami, M. (2005). "New primal-dual interior-point methods based on kernel functions". *Doctoral Dissertation, Delft University of Technology, The Netherlands*. http://resolver.tudelft.nl/uuid:0fa98a15-e579-4542-8c9f-cfecf3b64272

[12] El Ghami, M., Ivanov, I.D., Melissen, J.B.M., Roos, C., Steihaug, T. (2009). "A polynomial-time algorithm for linear optimization based on a new class of kernel functions". *Journal of Computational and Applied Mathematics*, 224(2), 500–513. https://doi.org/10.1016/j.cam.2008.05.027

[13] El Ghami, M., Guennoun, Z.A., Bouali, S., Steihaug, T. (2012). "Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term". *Journal of Computational and Applied Mathematics*, 236(15), 3613–3623. https://doi.org/10.1016/j.cam.2011.05.036

[14] Ibtissam, M., Randa, C., El Amir, D. (2025). "Complexity analysis of large-update interior-point methods for $P_*(\kappa)$-HLCP based on a new parametric kernel function". *Statistics, Optimization and Information Computing*, 14(1), 193–206. https://doi.org/10.19139/soic-2310-5070-2345

[15] Karmarkar, N. (1984). "A new polynomial-time algorithm for linear programming". *Combinatorica*, 4(4), 373–395. https://doi.org/10.1007/BF02579150

[16] Kojima, M., Mizuno, S., Yoshise, A. (1989). "A primal-dual interior point algorithm for linear programming". In: Megiddo, N. (ed.) *Progress in Mathematical Programming: Interior Point and Related Methods*. Springer, New York, 29–47. https://doi.org/10.1007/978-1-4613-9617-8_2

[17] Lee, J., Cho, G.-M. (2024). "A new full-Newton step infeasible interior-point method for linear optimization". *Journal of Nonlinear and Convex Analysis*, 25(12), 2991–3005. http://yokohamapublishers.jp/online2/opjnca/vol25/p2991.html

[18] Lesaja, G., Roos, C. (2010). "Unified analysis of kernel-based interior-point methods for $P_*(\kappa)$-linear complementarity problems". *SIAM Journal on Optimization*, 20(6), 3014–3039. https://doi.org/10.1137/090766735

[19] Megiddo, N. (1989). "Pathways to the optimal set in linear programming". In: Megiddo, N. (ed.) *Progress in Mathematical Programming: Interior Point and Related Methods*. Springer, New York, 131–158. https://doi.org/10.1007/978-1-4613-9617-8_8

[20] Nesterov, Y.E., Nemirovskii, A.S. (1994). *Interior Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, vol. 13. SIAM, Philadelphia. https://doi.org/10.1137/1036175

[21] Peng, J., Roos, C., Terlaky, T. (2001). "A new and efficient large-update interior-point method for linear optimization". *Journal of Computational Technologies*, 6(4), 61–80. https://optimization-online.org/?p=8564

[22] Peng, J., Roos, C., Terlaky, T. (2002). *Self-Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms*. Princeton University Press. https://doi.org/10.1515/9781400825134

[23] Peyghami, M.R., Hafshejani, S.F. (2014). "Complexity analysis of an interior-point algorithm for linear optimization based on a new proximity function". *Numerical Algorithms*, 67(1), 33–48. https://doi.org/10.1007/s11075-013-9772-1

[24] Roos, C., Terlaky, T., Vial, J.P. (1997). *Theory and Algorithms for Linear Optimization: An Interior Point Approach*. Wiley, Chichester.

[25] Sonnevend, G. (1986). "An 'analytic center' for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming". *Lecture Notes in Control and Information Science*, 84, 866–876. Springer, Berlin. https://doi.org/10.1007/BFb0043914

[26] Ye, Y. (1997). *Interior point algorithms: Theory and analysis*. John Wiley and Sons, New York. https://doi.org/10.1002/9781118032701.scard

Author Bio-sketch

Abderrahim Guemmaz received his Ph.D. in Mathematics from the University of Batna 2, Batna, Algeria, in 2022. He is currently a Lecturer Professor at the University Center of Barika, Barika, Algeria, and a member of the Laboratory of Science for Mathematics, Computer Science and Engineering Applications. His research interests include optimization theory, interior-point methods, linear programming, and computational mathematics. He has published several papers on kernel-based interior-point methods and parametric kernel functions in international journals. His current research focuses on improving complexity bounds for large-update interior-point methods. Email: abderrahim.guemmaz@cu-barika.dz