

École polytechnique de Louvain

Optimization Methods based on Interpolation Conditions

Author: **Anthony RICHELLE**

Supervisors: **François GLINEUR, Julien HENDRICKX**

Readers: **François GLINEUR, Julien HENDRICKX, Geovani NUNES
GRAPIGLIA**

Academic year 2025–2026

Master [120] : ingénieur civil en mathématiques appliquées

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Professor François Glineur and Professor Julien Hendrickx, for their guidance, expertise, and continuous support throughout this work. Their insightful feedback and thoughtful questions pushed me to think deeper and develop as a researcher. It has been an immense pleasure to work under their supervision.

I would also like to extend my thanks to my friends, Robin and Giovanni, who were a constant source of encouragement and moral support throughout the exploration of new ideas and writing of this master thesis. Their patience, good humour, and support made every day more enjoyable. I am grateful to have had them by my side. Finally, I am thankful to all those who, in one way or another, contributed to making this work possible.

Abstract

This thesis investigates smooth convex optimization in the regime where the problem dimension is small relative to the number of oracle evaluations. This setting is largely overlooked by classical complexity theory, whose lower bounds are designed for the opposite regime (when the number of first order oracle evaluations is less than the problem dimension), leaving open the possibility of doing better in any fixed finite-dimensional space. The central question is whether retaining and exploiting the full history of past first order oracle evaluations, rather than discarding it as standard first-order methods do, can yield more effective algorithms when oracle calls are costly. When each evaluation is expensive, the cost of solving a harder subproblem, such as a quadratic program to determine the next query point becomes negligible by comparison, making it worthwhile to build and exploit richer models of the objective.

The investigation is deliberately exploratory: rather than optimizing a single method, it maps the design space by revisiting cutting-plane and bundle methods under smoothness assumptions, and by developing new algorithms grounded in the use of smooth convex interpolation conditions. The proposed approaches are evaluated experimentally on a range of smooth convex problems to assess their relative strengths and limitations. Those preliminary results are encouraging: the proposed methods consistently outperform standard first-order methods in the targeted regime and often exhibit linear convergence, suggesting that aggressive memory exploitation is a productive direction for this class of problems.

Contents

1	Introduction	1
2	Mathematical preliminaries	3
2.1	First-Order Methods	3
2.2	Convexity	3
2.2.1	Coercivity	4
2.3	Smoothness	5
2.4	Smooth-Convex Interpolation	5
2.5	Smooth-Convex Upper Bound	6
2.5.1	Primal Formulation	7
2.5.2	Dual Formulation	7
2.5.3	Linear constraints formulation	7
2.6	Smooth-Convex Lower Bound	8
2.6.1	Primal Formulation	9
2.6.2	Linear constraints formulation	9
2.7	Minimizer localization and extremal values	9
2.8	Stationarity Conditions	10
2.9	First-order lower bounds	11
2.9.1	Convex Lower Bound	11
2.9.2	Smooth-Convex Lower bound	11
2.9.3	Finite dimensional lower bound	12
2.10	Notation Table	14
3	Existing methods	15
3.1	Convex minimization methods	15
3.1.1	Kelley's Method	15
3.1.2	Level Method	16
3.1.3	Proximal Bundle Method	17
3.1.4	Ellipsoid Method	18
3.1.5	Analytic Center Cutting-Plane Method	19
3.1.6	Vaidya's Volumetric Cutting-Plane Method	20
3.1.7	Summary	21
3.2	Smooth convex minimization methods	21
3.2.1	Gradient Method	21
3.2.2	Fast Gradient Method	22
3.2.3	Optimized Gradient Method	23

3.2.4	Gradient Method with Memory	23
3.2.5	Improved Gradient Method with Memory	24
3.2.6	Optimized Gradient Method with Memory	25
3.2.7	Smooth Proximal Bundle Method	26
3.2.8	Summary	28
4	Proposed Methods	29
4.1	Minimizing extremal functions	29
4.1.1	Pessimistic Approach	29
4.1.2	Optimistic Approach	30
4.2	Improved convex techniques	30
4.2.1	Smooth Level Method	31
4.2.2	Smooth Ellipsoid Method	31
4.2.3	Smooth ACCPM	33
4.3	New methods	33
4.3.1	Lower Level Method	34
4.3.2	Bound Mixing Method	34
4.3.3	Information Maximization	36
4.3.4	Lower Bound Method (LBM)	37
5	Refinements and Theoretical Properties of LBM	39
5.1	Invariance	39
5.1.1	Invariance to transformations of functional value	39
5.1.2	Invariance to transformations of space	40
5.1.3	Adaptation to lower dimensional structure	40
5.2	Boundedness	41
5.2.1	Euclidean Norm	41
5.2.2	Infinity norm based boundedness	42
5.3	Convergence Proof for LBM	42
5.4	Warm Start	44
5.4.1	Infeasibility warm-start	44
5.4.2	Trust Region Warm-Start	45
5.4.3	Hyperparameter Free Warm-Start	45
5.4.4	Link with level method	46
5.5	Averaging Techniques	47
5.5.1	Dual Averaging	47
5.5.2	Averaging technique comparison	49
5.6	Pruning Techniques	50
5.6.1	Technique comparison	51
6	Numerical Experiments	54
6.1	Quadratic Functions	55
6.1.1	Baseline benchmark	56
6.1.2	Benchmark with increased dimension	59
6.1.3	Benchmark with increased L	61
6.2	Logistic Regression	62

6.3	Huber Regression	64
6.4	Overall Assessment	65
7	Conclusion	67
	Bibliography	70

Chapter 1

Introduction

Standard first-order methods for smooth convex optimization share a common characteristic: they make limited use of the information gathered during the optimization process. At each iteration, only the current point and its gradient drive the update, and while some methods retain information from previous iterations through averaging or momentum, the full history of function values and gradients is never systematically exploited. This design choice is natural in the regime that classical complexity theory primarily considers, where the number of iterations remains small relative to the problem dimension and little can be done with accumulated history. In a fixed finite-dimensional space, however, the picture changes. As oracle evaluations accumulate, the collected function values and gradients increasingly constrain the set of possible objective functions, and richer models of the function become available. The question is whether these models can be exploited to build more effective algorithms.

This thesis investigates that question in the smooth convex setting. The focus is on problems where oracle evaluations are expensive and constitute the main computational bottleneck. In this context, spending additional effort between evaluations to solve a harder subproblem, such as a quadratic program determining the next query point, is justified if it reduces the total number of oracle calls required. This shifts the design focus toward methods that build and exploit global models of the objective, aggregating all past evaluations rather than discarding them.

The theoretical foundation for this approach lies in smooth convex interpolation conditions, which characterize exactly when a finite set of function-gradient pairs is consistent with some smooth convex function. These conditions make it possible to construct upper and lower bounding models from sampled points, to localize minimizers, and to derive principled criteria for selecting the next evaluation. The use of such conditions as an algorithmic tool, rather than purely as an analysis device, is a central contribution of this work.

This perspective has a natural precedent in nonsmooth optimization, where cutting-plane and bundle methods have long exploited accumulated subgradient information to build global piecewise-linear models of the objective. The smooth setting introduces additional structure that these methods do not capture, and extending their philosophy to smooth problems is one of the main threads of this thesis. Recent work on memory-based and bundle methods for smooth objectives confirms that this line of inquiry is active, and raises unresolved questions about how to construct minimal global models from past evaluations and how to use them to guide the selection of future queries.

The investigation is deliberately exploratory. Rather than proposing a single method with sharp theoretical guarantees, this thesis maps the design space: revisiting methods from nonsmooth optimization under smoothness assumptions, developing new algorithms based on interpolation conditions, and evaluating them experimentally to characterize when and how memory exploitation is beneficial. Chapter 2 introduces the theoretical background, including interpolation conditions and extremal model constructions. Chapter 3 reviews existing methods that incorporate memory or global models. Chapter 4 introduces the new methods developed in this work. Chapter 5 addresses properties and refinements of these methods. Chapter 6 presents experimental results. Chapter 7 concludes and discusses directions for future research.

Chapter 2

Mathematical preliminaries

Before developing the methods and analysis that form the core of this thesis, it is useful to fix the mathematical setting precisely. This chapter covers the essential properties of smooth convex functions and the guarantees one can expect from first-order optimization methods. This chapter is largely based on Nesterov [19].

2.1 First-Order Methods

Definition 1 (First order oracle). *A first-order oracle for a function f is a mapping $\mathbb{O}$ such that, given a point $x \in \mathbb{R}^d$, returns a pair :*

$$\mathbb{O}(x) = (f(x), g(x)), \quad g(x) \in \partial f(x)$$

First order methods are methods which, at each iteration, make a call to a first-order oracle. If the function is non-differentiable but is convex the oracle returns a subgradient instead of the gradient. Finding the exact minimizer of a function proves to be a generally hard problem. First-order methods only aim to find an approximate minimizer.

Definition 2 (ε -approximate solution). *For a function f with a well-defined minimizer x^* , an ε -approximate solution is a point $x \in \text{dom}(f)$ such that :*

$$f(x) - f(x^*) \leq \varepsilon$$

The key metric used here to measure the performance of first order methods is the number of *oracle calls* required to find an ε -approximate solution. Additional assumptions on f are needed in order to study the performance of available methods.

2.2 Convexity

The convexity assumption will always be made throughout this thesis, non-convex optimization methods will not be covered. In the convex case, it is common to consider a $\text{dom}(f) = Q$ where Q is a convex set with non empty interior.

Definition 3 (Convexity). *A function $f : Q \rightarrow \mathbb{R}$ is said to be convex on Q if for all $x, y \in Q$ and all $\lambda \in [0, 1]$,*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Theorem 1 (Sub-differential inequality). *Let $f : Q \rightarrow \mathbb{R}$ be a convex function, and let $x \in \text{dom}(f)$. Then for any $g \in \partial f(x)$ and any $y \in Q$,*

$$f(y) \geq f(x) + \langle g, y - x \rangle.$$

If f is differentiable at x , then $\partial f(x) = \{\nabla f(x)\}$.

At each iteration a first-order method makes a call to the first-order oracle, which returns a pair $(f(x), g(x))$. In order to facilitate notation, let's characterize each oracle call with the triplet $(x, g(x), f(x))$ and call that a sample. Let m be the number of such samples, the samples collected for the points $(x_i)_{i=1}^m$ are denoted as $(x_i, g_i, f_i)_{i=1}^m$.

The two inequalities provided by convexity yield pointwise upper and lower bounds on f from the sample triplets $(x_i, g_i, f_i)_{i=1}^m$. It is possible to construct convex interpolants that are exactly tight at these bounds. For more information about convex interpolation conditions and their construction, see [25].

Definition 4 (Convex interpolable). *A finite set of triples $\{(x_i, g_i, f_i)\}_{i=1}^m$ with $x_i \in Q$, $g_i \in \mathbb{R}^d$, and $f_i \in \mathbb{R}$ is said to be convex interpolable if there exists a convex function $f : Q \rightarrow \mathbb{R}$ such that*

$$f(x_i) = f_i \quad \text{and} \quad g_i \in \partial f(x_i), \quad \text{for all } i = 1, \dots, m.$$

Theorem 2 (Convex interpolation conditions). *A finite set of triples $\{(x_i, g_i, f_i)\}_{i=1}^m$ is convex interpolable if and only if*

$$f_i \geq f_j + g_j^\top (x_i - x_j), \quad \text{for all } i, j = 1, \dots, m.$$

The interpolation conditions of [Theorem 2](#) characterize which set of samples admit a convex interpolant. We now exhibit the tightest such interpolants explicitly: the functions below are convex, consistent with $(x_i, g_i, f_i)_{i=1}^m$, and respectively achieve the pointwise maximum lower bound and minimum upper bound over all convex interpolants. For the following 2 bounds check [16].

Theorem 3 (Tight convex lower bound). *Given the triplets $(x_i, g_i, f_i)_{i=1}^m$ satisfying the conditions of [Theorem 2](#), the function*

$$\hat{f}_m(x) = \max_{i \leq m} \{f_i + \langle g_i, x - x_i \rangle\}$$

is convex, satisfies $\hat{f}_m(x) \leq f(x)$ for any convex interpolant f of $(x_i, g_i, f_i)_{i=1}^m$, and is itself a convex interpolant of $(x_i, g_i, f_i)_{i=1}^m$.

Theorem 4 (Tight convex upper bound). *Given the triplets $(x_i, g_i, f_i)_{i=1}^m$ satisfying the conditions of [Theorem 2](#), the function*

$$h_h(x) = \min_{\lambda \in \Delta_m} \sum_i \lambda_i f_i \quad \text{s.t.} \quad x = \sum_i \lambda_i x_i$$

is convex, satisfies $h_h(x) \geq f(x)$ for any $x \in \text{conv}(\{x_i\})$ (if x is not in the convex hull of the sample points then $h_h(x) = +\infty$) for any convex interpolant f of $(x_i, g_i, f_i)_{i=1}^m$, and is itself a convex interpolant of $(x_i, g_i, f_i)_{i=1}^m$.

2.2.1 Coercivity

Another useful assumption which ensures the existence of a minimizer in the convex case is coercivity.

Definition 5 (Coercive). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$. The function f is said to be coercive if*

$$\|x\| \rightarrow \infty \Rightarrow f(x) \rightarrow +\infty.$$

Theorem 5. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex and coercive function. Then f admits a minimizer, i.e.,

$$\exists x^* \in \mathbb{R}^d \text{ such that } f(x^*) = \inf_{x \in \mathbb{R}^d} f(x).$$

Moreover, the minimum value $\inf_{x \in \mathbb{R}^d} f(x)$ is unique. If f is strictly convex, then the minimizer x^* is also unique.

2.3 Smoothness

Convex functions have no bounds on how their gradients can vary, and can even be non-differentiable. A common assumption in optimization is smoothness : it bounds how similar the gradient of neighboring points must be. For the smooth case, it will always be considered that $\text{dom}(f) = \mathbb{R}^d$ and d will be the dimension of the problem.

Definition 6 (L -smoothness). A differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be L -smooth (or to have Lipschitz continuous gradient with constant $L > 0$) if for all $x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

Theorem 6 (Quadratic upper bound). Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a differentiable L -smooth function. Then for all $x, y \in \mathbb{R}^d$,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

2.4 Smooth-Convex Interpolation

Similarly to convex interpolation conditions, smooth convex interpolation conditions can be derived, more about this approach can be found in [25] (most of what comes in the following sections can be found there). Those interpolation conditions combine both smoothness and convexity into a single set of inequalities.

Definition 7 (Interpolable). A set of triplets $(x_i, g_i, f_i)_{i=1}^m$ is said to be L -smooth convex interpolable if and only if there exists an L -smooth convex function f such that

$$\forall i : \quad f(x_i) = f_i, \quad \nabla f(x_i) = g_i.$$

Theorem 7 (Interpolation conditions). A set of triplets $(x_i, g_i, f_i)_{i=1}^m$ is L -smooth convex interpolable if and only if

$$\forall i, j \in \{1, \dots, m\} : \quad f_j \geq f_i + \langle g_i, x_j - x_i \rangle + \frac{1}{2L}\|g_i - g_j\|^2$$

These conditions are a strengthened version of convexity encapsulating smoothness. In fact they can be reduced to the convex interpolation conditions in a shifted space.

$$\forall i, j \in \{1, \dots, m\} : \quad f_j - \frac{1}{2L}\|g_j\|^2 \geq f_i - \frac{1}{2L}\|g_i\|^2 + \left\langle g_i, \left(x_j - \frac{1}{L}g_j\right) - \left(x_i - \frac{1}{L}g_i\right) \right\rangle$$

If one defines the shifted samples $(u_i, g_i, s_i)_{i=1}^m = (x_i - \frac{1}{L}g_i, g_i, f_i - \frac{1}{2L}\|g_i\|^2)_{i=1}^m$ the smooth convex interpolation conditions become exactly the convex interpolation condition on the shifted samples, highlighting how those conditions strongly relate to convexity.

$$\forall i, j \in \{1, \dots, m\} : \quad s_j \geq s_i + \langle g_i, u_j - u_i \rangle$$

Similarly to the convex case, one can identify extremal smooth-convex interpolants that provide tight pointwise upper and lower bounds. Specifically, if the set of triplets is L -smooth convex interpolable, there exist two L -smooth convex interpolating functions F_l and F_h such that any L -smooth convex interpolant f satisfies

$$F_l(x) \leq f(x) \leq F_h(x), \quad \forall x \in \mathbb{R}^d.$$

The construction of F_l and F_h proceeds by reduction to the convex (non-smooth) case via convex conjugacy as described in [11]. If F is L -smooth convex, its convex conjugate F^* is $\frac{1}{L}$ -strongly convex and interpolates the transformed triplets $\{(g_i, x_i, \langle g_i, x_i \rangle - f_i)\}_{i=1}^m$. Defining

$$\tilde{h}(y) = F^*(y) - \frac{1}{2L}\|y\|^2 \quad \text{interpolates} \quad \left(g_i, x_i - \frac{1}{L}g_i, \langle g_i, x_i \rangle - f_i - \frac{1}{2L}\|g_i\|^2\right)_{i=1}^m$$

The function $\tilde{h}$ is convex, and its associated tight convex lower and upper bounds are $\tilde{h}_l, \tilde{h}_h$ respectively. Those bounds arise from Theorem 3 and Theorem 4 applied to the transformed sample set and yields $\tilde{h}_l(y) \leq \tilde{h}(y) \leq \tilde{h}_h(y)$ for all y . The extremal smooth-convex interpolants are retrieved using conjugacy as follows :

$$F_h(x) = \max_{y \in \mathbb{R}^d} \left\{ \langle y, x \rangle - \tilde{h}_l(y) - \frac{1}{2L}\|y\|^2 \right\},$$

$$F_l(x) = \max_{y \in \mathbb{R}^d} \left\{ \langle y, x \rangle - \tilde{h}_h(y) - \frac{1}{2L}\|y\|^2 \right\}.$$

Both F_l and F_h are L -smooth and convex as they are conjugates of functions of the form $h + \frac{1}{2L}\|\cdot\|^2$, where h is convex, making $h + \frac{1}{2L}\|\cdot\|^2$ a $\frac{1}{L}$ -strongly convex function. The conjugate of a $\frac{1}{L}$ -strongly convex function is L -smooth convex. Since they are interpolants of $(x_i, g_i, f_i)_{i=1}^m$ and the transformations are order reversing/preserving, these functions are L -smooth convex and are upper/lower-bounds of any L -smooth convex functions interpolating $(x_i, g_i, f_i)_{i=1}^m$.

2.5 Smooth-Convex Upper Bound

Multiple formulation for the smooth-convex upper bounding function will be derived. They all are used in different contexts as depending on the number of samples and the dimension of the problem, one or the other can be the most efficient to compute. Since h_l interpolates the samples $(g_i, x_i - \frac{1}{L}g_i, \langle g_i, x_i \rangle - f_i - \frac{1}{2L}\|g_i\|^2)_{i=1}^m$ the definition of the upper-bounding function is as follows :

$$\begin{aligned} \tilde{h}_l(y) &= \max_{1 \leq i \leq m} \left\{ \langle g_i, x_i \rangle - f_i - \frac{1}{2L}\|g_i\|^2 + \langle x_i - \frac{1}{L}g_i, y - g_i \rangle \right\} \\ &= \max_{1 \leq i \leq m} \left\{ -f_i + \frac{1}{2L}\|g_i\|^2 + \langle x_i - \frac{1}{L}g_i, y \rangle \right\} \\ F_h(x) &= \max_{y \in \mathbb{R}^d} \left\{ \langle y, x \rangle - \frac{1}{2L}\|y\|^2 - \tilde{h}_l(y) \right\} \\ &= \max_{y \in \mathbb{R}^d} \min_{1 \leq i \leq m} \left\{ f_i + \left\langle x - \left(x_i - \frac{1}{L}g_i\right), y \right\rangle - \frac{1}{2L}(\|y\|^2 + \|g_i\|^2) \right\} \end{aligned}$$

2.5.1 Primal Formulation

The primal formulation is the most intuitive and easiest to interpret formulation of the smooth-convex upper bounding function. Starting from :

$$\begin{aligned} F_h(x) &= \max_{y \in \mathbb{R}^d} \min_{1 \leq i \leq m} \left\{ f_i + \left\langle x - \left(x_i - \frac{1}{L} g_i \right), y \right\rangle - \frac{1}{2L} (\|y\|^2 + \|g_i\|^2) \right\} \\ &= \max_{y \in \mathbb{R}^d} \min_{1 \leq i \leq m} \left\{ f_i + \langle y, x - x_i \rangle - \frac{1}{2L} \|y - g_i\|^2 \right\} \end{aligned}$$

Using the epigraph technique, this can be rewritten to be the (x, y, t) sample maximizing t under the constraint of the interpolation conditions provided by the samples.

$$F_h(x) = \max_{t, y \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad f_i \geq t + \langle g, x_i - x \rangle + \frac{1}{2L} \|g_i - y\|^2$$

2.5.2 Dual Formulation

Another formulation can be derived from the same from the starting point as the primal formulation. The pointwise minimization can be rewritten over a simplex in order to apply strong duality as the objective is concave in y .

$$\begin{aligned} F_h(x) &= \max_{y \in \mathbb{R}^d} \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \left[f_i + \left\langle x - \left(x_i - \frac{1}{L} g_i \right), y \right\rangle - \frac{1}{2L} (\|y\|^2 + \|g_i\|^2) \right] \right\} \\ &= \min_{\lambda \in \Delta_m} \max_{y \in E^*} \left\{ \sum_i \lambda_i f_i + \left\langle x - \sum_i \lambda_i \left(x_i - \frac{1}{L} g_i \right), y \right\rangle - \frac{1}{2L} \left(\|y\|^2 + \sum_i \lambda_i \|g_i\|^2 \right) \right\} \end{aligned}$$

The maximization problem can be solved using the first order stationarity condition :

$$x - \sum_i \lambda_i \left(x_i - \frac{1}{L} g_i \right) - \frac{1}{L} y^* = 0 \iff y^* = L \left(x - \sum_i \lambda_i \left(x_i - \frac{1}{L} g_i \right) \right)$$

Plugging this expression in the upperbound function this yields :

$$F_h(x) = \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \left(f_i - \frac{1}{2L} \|g_i\|^2 \right) + \frac{L}{2} \left\| x - \sum_i \lambda_i \left(x_i - \frac{1}{L} g_i \right) \right\|^2 \right\}$$

Using the shifted point representation this becomes :

$$F_h(x) = \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i s_i + \frac{L}{2} \left\| x - \sum_i \lambda_i u_i \right\|^2 \right\}$$

2.5.3 Linear constraints formulation

Computing the value of $F_h(x)$ at a given point using the primal formulation might be expensive numerically as it requires m conic constraints and $d + 1$ variables. The dual on the other hand has m variables and $m + 1$ linear constraints with a quadratic (strongly convex) objective. In cases where the number of constraints is large compared to the dimension, the two formulation presented before require significant computational resources. This work derives a new formulation which combines both formulation to

achieve $d + 1$ variables and m linear constraints, and is far more efficient computationally in that regime. Let's start from the dual formulation and set :

$$F_h(x) = \min_{\lambda \in \Delta_m, z \in \mathbb{R}^d} \left\{ \sum_i \lambda_i s_i + \frac{L}{2} \|x - z\|^2 \right\} \quad \text{s.t.} \quad z = \sum_i \lambda_i u_i$$

Let's write this problem using the Lagrangian with the dual variable g of the equality constraint on z .

$$F_h(x) = \max_{g \in \mathbb{R}^d} \min_{\lambda \in \Delta_m, z \in \mathbb{R}^d} \left\{ \sum_i \lambda_i s_i + \frac{L}{2} \|x - z\|^2 + g^\top \left[z - \sum_i \lambda_i u_i \right] \right\}$$

The optimal value for z is given by the KKT conditions and is simply $z = x - \frac{1}{L}g$, yielding :

$$F_h(x) = \max_{g \in \mathbb{R}^d} \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i [s_i + \langle g, x - u_i \rangle] - \frac{1}{2L} \|g\|^2 \right\}$$

Since the dependence on λ is linear, it can either be transformed into a discrete minimization or constraints using the epigraph technique.

$$F_h(x) = \max_{t, g \in \mathbb{R}^{d+1}} \left\{ t - \frac{1}{2L} \|g\|^2 \right\} \quad \text{s.t.} \quad t \leq s_i + \langle g, x - u_i \rangle$$

This reformulation is in practice much faster to compute than the two others when $m \approx d$. It is also sometimes easier to work with as it only has linear inequality constraints.

2.6 Smooth-Convex Lower Bound

The lower interpolant is derived in a very similar manner to the upper interpolant and will also have three very similar formulation that each have their advantages and disadvantages. Starting from the definition of the lower bounding functions, one has :

$$\begin{aligned} \tilde{h}_h(y) &= \min_{\lambda \in \Delta_m} \sum_i \lambda_i \left[-f_i + \langle g_i, x_i \rangle - \frac{1}{2L} \|g_i\|^2 \right] \quad \text{s.t.} \quad y = \sum_i \lambda_i g_i \\ F_l(x) &= \max_{y \in \mathbb{R}^d} \left\{ \langle y, x \rangle - \tilde{h}_h(y) - \frac{1}{2L} \|y\|^2 \right\} \end{aligned}$$

Notice that here, y is entirely determined by λ due to the equality constraint in $\tilde{h}_h(y)$. This yields the dual formulation directly :

$$F_l(x) = \max_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \left(f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \right) - \frac{1}{2L} \left\| \sum_i \lambda_i g_i \right\|^2 \right\}$$

Let us introduce an easy notation for the affine term in the above expression that will significantly improve readability in future sections.

$$b_i = f_i - \langle g_i, x_i \rangle + \frac{1}{2L} \|g_i\|^2 \quad \text{and} \quad \psi_i(x) = b_i + \langle g_i, x \rangle$$

Notice that this quantity b_i it is also the opposite of the function value of the convex Fenchel dual of f at g_i (i.e. $\tilde{h}(g_i)$). With that expression the lower bound becomes :

$$F_l(x) = \max_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \psi_i(x) - \frac{1}{2L} \left\| \sum_i \lambda_i g_i \right\|^2 \right\}$$

2.6.1 Primal Formulation

The primal formulation of the upper bounding function maximized the functional value compatible with the interpolation conditions. Quite intuitively, here the functional value compatible the data will simply be minimized. The derivation uses a dual variable y . Using the Fenchel conjugacy properties ℓ_2 -norm one has :

$$-\frac{1}{2L} \left\| \sum_i \lambda_i g_i \right\| = \min_{y \in \mathbb{R}^d} \left\{ \left\langle y, \sum_i \lambda_i g_i \right\rangle + \frac{1}{2L} \|y\|^2 \right\}$$

Using this identity, the dual formulation becomes the following :

$$F_l(x) = \max_{\lambda \in \Delta_m} \min_{y \in \mathbb{R}^d} \left\{ \sum_i \lambda_i \left(f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \right) + \left\langle y, \sum_i \lambda_i g_i \right\rangle + \frac{L}{2} \|y\|^2 \right\}.$$

Letting $g = -Ly$ and using strong duality leads to :

$$F_l(x) = \min_{g \in \mathbb{R}^d} \max_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \left(f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i - g\|^2 \right) \right\}.$$

Using the epigraph technique on the simplex maximization yields the primal formulation.

$$F_l(x) = \min_{t, g \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad t \geq f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i - g\|^2$$

This formulation is the most intuitive one as it directly minimizes the functional value f of the sample at point $x : (x, g, f)$ under the constraints imposed by the interpolation conditions. Note that this minimization is done for any possible value of the gradient g at x .

2.6.2 Linear constraints formulation

Similarly to the upper-bounding function this work derives a formulation for the lower bounding function with $d + 1$ variables and m linear inequality constraints and offering a more efficient computational approach when $m > d$ by avoiding both the conic constraints and the large number of variables. The derivation starts from the primal formulation, notice that :

$$F_l(x) = \min_{t, g \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad t - \frac{1}{2L} \|g\|^2 \geq f_i + \left\langle g_i, \left(x - \frac{g}{L} \right) - x_i \right\rangle + \frac{1}{2L} \|g_i\|^2$$

Setting $g = L(x - y)$, and adding $\frac{1}{2L} \|g\|^2$, one gets rid of the conic constraints. The problem becomes the following :

$$\begin{aligned} F_l(x) &= \min_{t, y \in \mathbb{R}^{d+1}} t + \frac{L}{2} \|x - y\|^2 \quad \text{s.t.} \quad t \geq f_i + \langle g_i, y - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \\ &= \min_{t, y \in \mathbb{R}^{d+1}} t + \frac{L}{2} \|x - y\|^2 \quad \text{s.t.} \quad t \geq \psi_i(y) \end{aligned}$$

2.7 Minimizer localization and extremal values

Convexity offers upper and lower bounds on the possible values the minimizer could have if it were at a certain point x . Those bounds are found by searching for which values of f^* the sample $(x, 0, f^*)$ is compatible with $(x_i, g_i, f_i)_{i=1}^m$.

$$\begin{aligned} f^* &\geq \tilde{f}_{\min}^* = \max_{i \leq m} \{f_i + \langle g_i, x - x_i \rangle\} \\ f^* &\leq \tilde{f}_{\max}^* = \min_{i \leq m} f_i \end{aligned}$$

Using these, one can define a set of x which are potential minimizer. Indeed, for any $\tilde{x}$ such that $\tilde{f}_{\max}^*(\tilde{x}) < \tilde{f}_{\min}^*(\tilde{x})$ there is no function compatible with the data such that $\nabla f(\tilde{x}) = 0$. On the other hand, if $\tilde{f}_{\max}^*(\tilde{x}) \geq \tilde{f}_{\min}^*(\tilde{x})$ then there exists a function that is compatible with $(x_i, g_i, f_i)_{i=1}^m$ attaining a minimizer at that point. Let's denote that polyhedron $\tilde{\mathcal{X}}_m$ and call it the minimizer localization set

$$\tilde{\mathcal{X}}_m = \left\{ x \text{ s.t. } \forall i = 1, \dots, m : \langle g_i, x_i - x \rangle \geq f_i - \min_j f_j \right\}$$

The interpolation conditions are a strengthened version of convexity, as such they provide stronger bounds. They yield the following bounds :

$$\begin{aligned} f^* &\geq f_{\min}^* = \max_{i \leq m} \psi_i(x) \\ f^* &\leq f_{\max}^* = \min_{i \leq m} s_i \end{aligned}$$

Let $\phi_m(x) = f_{\min}^*(x)$ be this max of affine functions, this notation will prove useful in future sections. Similarly to convexity, these define a polyhedral minimum localization set $\mathcal{X}_m$.

$$\mathcal{X}_m = \left\{ x \text{ s.t. } \forall i = 1, \dots, m : \langle g_i, x_i - x \rangle \geq f_i - f_{\max}^* + \frac{1}{2L} \|g_i\|^2 \right\}$$

Each one of theses inequalities is as a cutting plane that can be rewritten as a half space constraint on x as follows :

$$\forall x \in \mathcal{X}_m : \quad \forall i = 1, \dots, m : \left\langle g_i, \left[x_i - \left(\frac{f_i - f_{\max}^*}{\|g_i\|^2} + \frac{1}{2L} \right) g_i \right] - x \right\rangle \geq 0$$

In this sense, each gradient cuts a half space of possible values for x . Note that adding a constraint does more than just adding a cutting plane, it can also reduce the value of $f_{\max}^*$, tightening the polyhedron by shifting all the other cuts. This last formulation has heavy links with the Polyak step size [23]. Indeed, if f^* is known then no need to use the upper-bound $f_{\max}^*$, the orthogonal projection from point x_i to the hyperplane is then given by the shifted point x_i^+ :

$$x_i^+ = x_i - \left(\frac{f_i - f^*}{\|g_i\|^2} + \frac{1}{2L} \right) g_i$$

The second term that is added to the Polyak step size is a correction that accounts for the smoothness of the function and is a gradient step of size $\frac{1}{2L}$ further than the regular Polyak step.

2.8 Stationarity Conditions

Multiple stationarity conditions can be used when studying first order methods. For smooth convex functions, a clear hierarchy between these conditions can be established. Indeed, assume that $\|x_i - x^*\| \leq \varepsilon$. Then, by the quadratic upper bound,

$$f_i \leq f^* + \frac{L}{2} \|x_i - x^*\|^2 \iff f_i - f^* \leq \frac{L}{2} \|x_i - x^*\|^2$$

Furthermore, smooth convex interpolation yields a lower bound on the gradient norm depending on the functional residual.

$$f_i - f^* \geq \frac{1}{2L} \|g_i\|^2 \iff \|g_i\| \leq \sqrt{2L(f_i - f^*)} \leq L \|x_i - x^*\|$$

Combining these inequalities yields the following hierarchy of stationarity conditions :

$$\|x_i - x^\star\| \leq \varepsilon \implies f_i - f^\star \leq \frac{L\varepsilon^2}{2} \implies \|g_i\| \leq L\varepsilon.$$

Among the three stationarity measures, the distance to the optimum is the strongest: it controls both the functional residual and the gradient norm. The reverse implications do not hold in general, so the hierarchy is strict. Note that all three inequalities are simultaneously tight, as they are all saturated by the quadratic $f(x) = \frac{L}{2}\|x - x^\star\|^2$.

2.9 First-order lower bounds

It is possible to derive lower bounds on the convergence rate in functional residual for first order method. Consider the following problem :

$$\min_{x \in Q} f(x) \tag{2.1}$$

Where f and Q will vary from problem type to problem type. In all cases studied here, f and Q are convex, yielding a convex minimization. Q will also be assumed to have a non-empty interior. These lower bounds will help characterize the method studied by comparing the upper bound on the number of iterations they require to reach a ε -approximate solution to the lower bound. More information about these bounds (notably their proof) can be found in [19].

Definition 8 (Optimal method). *An optimization method is called optimal when the number of oracle calls it requires to find a ε -approximate minimizer is of the same $\mathcal{O}$ order as the lower bound for that class of functions.*

2.9.1 Convex Lower Bound

Theorem 8 (Lower bound). *For any k such that $0 \leq k \leq d - 1$, and any $x_0 \in \mathbb{R}^d$, let R be such that $\|x_0 - x^\star\| \leq R$, there exists an M -Lipschitz convex function f on $B_2(x^\star, R)$ such that :*

$$f(x_k) - f^\star \geq \frac{MR}{2(2 + \sqrt{k+1})}.$$

for any optimization scheme, which generates a sequence (x_k) satisfying the condition

$$x_k \in x_0 + \text{span}\{g(x_0), \dots, g(x_{k-1})\},$$

This bound and its derivation can be found in [19]. For any Q with non-empty interior, the hard instance is supported on a ball which can always be embedded inside Q , so the bound carries over directly. Taking $d \rightarrow \infty$ with $k < d$, gives an asymptotic bound on the number of iterations required to find a ε -approximate solution of $\mathcal{O}(1/\varepsilon^2)$ oracle calls.

2.9.2 Smooth-Convex Lower bound

Theorem 9 (Lower bound). *For any k such that $1 \leq k \leq \frac{1}{2}(d - 1)$, and any $x_0 \in \mathbb{R}^d$, there exists a L -smooth convex function f such that :*

$$x_k \in x_0 + \text{span}\{g_0, \dots, g_{k-1}\}$$

For any first-order method generating iterates (x_k) satisfying the following condition :

$$f(x_k) - f^* \geq \frac{3LR^2}{32(k+1)^2}, \quad \|x_k - x^*\|^2 \geq \frac{R^2}{8}$$

Here, assuming f to be L -smooth allows for a better bound than for convex non-smooth functions, see [19] for the derivation of the bound above. When $d \rightarrow \infty$ this bound shows that the minimum number of oracle calls to reach an ε -approximate minimizer is of order $\mathcal{O}(1/\sqrt{\varepsilon})$. It is interesting to note that, this bound is only valid for the $\frac{d-1}{2}$ first iterations instead of the $d-1$ first iterations (half the number of iterations).

2.9.3 Finite dimensional lower bound

The lower bounds discussed above are meaningful only up to a number of iterations that is small relative to the dimension d . In contrast, when the dimension is low, one can obtain a better convergence regime by explicitly localizing the minimizer inside Q with a number of iterations $k > d$.

The core idea behind finite-dimensional methods is to maintain an approximation of the localization set $\mathcal{X}_k \subseteq E_k \subseteq Q$ that is progressively refined at each iteration using cutting planes derived from convexity. The quantity that is typically controlled to guarantee convergence of these schemes is the distance of the minimizer x^* to the cutting planes defined given by the iterates :

$$v_i = \frac{1}{\|g_i\|} \langle g_i, x_i - x^* \rangle \geq 0$$

Nesterov shows that this quantity yields an upper bound on the functional residual when the function is convex and M -Lipschitz. The result will be shown here again in order to explicit what happens when the function is not M -Lipschitz but is L -smooth and convex (a very similar bound is obtained). Define $\bar{y}$ as follows :

$$\bar{y}_i = x^* + v_i \frac{g_i}{\|g_i\|}, \quad \langle g_i, \bar{y}_i - x_i \rangle = \langle g_i, x^* - x_i \rangle + v_i \|g_i\| = 0$$

This point $\bar{y}_i$ is the closest point to x^* on the cutting plane. As such $\|\bar{y}_i - x^*\| = v_i$. Also notice that $f(\bar{y}_i)$ is also at least as much as f_i as convexity yields the following bound.

$$f(\bar{y}_i) \geq f_i + \langle g_i, \bar{y}_i - x_i \rangle = f_i$$

Now, using either lipschitz continuity or smooth convexity, it is possible to bound the functional residual $f_i - f^*$ using the distance v_i .

$$f_i - f^* \leq f(\bar{y}_i) - f(x^*) \leq \begin{cases} M\|\bar{y}_i - x^*\| & \text{If } f \text{ is } M\text{-Lipschitz} \\ \frac{L}{2}\|\bar{y}_i - x^*\|^2 & \text{If } f \text{ is } L\text{-Smooth Convex} \end{cases}$$

By reducing $\text{Vol}_d(\mathcal{X}_k) \leq \text{Vol}_d(E_k) \rightarrow 0$, one guarantees convergence by combining the previous result with an inequality from Nesterov's book. It yields a bound depending on the functional residual depending on $\text{Vol}_d(\mathcal{X}_k)$.

$$\min_{i \leq k} v_i \leq D \left[\frac{\text{Vol}_d(\mathcal{X}_k)}{\text{Vol}_d(Q)} \right]^{\frac{1}{n}} \implies \min_{i \leq k} f_i - f^* \leq \begin{cases} MD \left[\frac{\text{Vol}_d(\mathcal{X}_k)}{\text{Vol}_d(Q)} \right]^{\frac{1}{n}} & \text{If } f \text{ is } M\text{-Lipschitz} \\ \frac{LD^2}{2} \left[\frac{\text{Vol}_d(\mathcal{X}_k)}{\text{Vol}_d(Q)} \right]^{\frac{2}{n}} & \text{If } f \text{ is } L\text{-Smooth Convex} \end{cases}$$

The more dimension increases, the less this approach is attractive as dividing the volume by c only diminishes the functional residual by a factor $\sim c^{\frac{-1}{d}}$. In any case, a linear convergence rate is achieved if one reduces $\text{Vol}(E_k)$ by at least a constant factor ρ at each iteration so that $\text{Vol}(E_k) \leq \rho^k \text{Vol}(E_0)$.

Similarly to the other kind of problem types, there is also a finite dimensional lower bound on the number of iterations required to reach an ε -approximate solution. This lower bound does not forbid a linear convergence rate. This bound is stated in [19], but it is not derived there.

Theorem 10 (Lower bound in finite dimension). *For any first-order method and any M -Lipschitz convex function f on $B_\infty(x^*, R)$, the number of oracle calls required to produce an ε -approximate solution satisfies*

$$k \geq n \ln \frac{MR}{8\varepsilon}.$$

2.10 Notation Table

General Notations	
x_0	Starting point of the method
$\mathcal{O}$	First-order oracle
m	Number of samples
k	Current iteration
i	Index of a sample
$(x_i, g_i, f_i)_{i=1}^m$	Samples generated by the optimization method
x^*	Minimizer of f
f_i	Value of f at x_i
g_i	Value of ∇f at x_i
d	Dimension of the input of f
R	Upper bound on $\ x_0 - x^*\ $
Q	Closed convex set with non-empty interior
D	Diameter of set Q
$B_2(c, R)$	ℓ_2 -ball of center c and radius R
$B_\infty(c, R)$	ℓ_∞ -ball of center c and radius R
$\text{Vol}(A)$	Volume of set A
Δ_m	Simplex of dimension m
Convex Non-Smooth	
M	Lipschitz constant of f
h_h	Convex upper bound given samples
$\hat{f}_k$	Convex lower bound given samples
$\tilde{f}_{\max}^*$	Upper-bound on f^* given samples
$\tilde{f}_{\min}^*$	Lower-bound on f^* given samples
$\mathcal{X}_m$	Minimum localization set
L -Smooth Convex	
L	Smoothness constant
s_i	Shifted function value : $f_i - \frac{1}{2L}\ g_i\ ^2$
u_i	Gradient step shifted point : $x_i - \frac{1}{L}g_i$
b_i	Opposite of dual h : $f_i - \langle g_i, x_i \rangle + \frac{1}{2L}\ g_i\ ^2$
$\psi_i(x)$	Affine function : $b_i + \langle g_i, x \rangle$
$\phi_m(x)$	Value of $f_{\min}^*$ at x : $\phi_m(x) = \max_{i \leq m} \psi_i(x)$
F_h	Smooth Convex upper-bound given samples
F_l	Smooth Convex lower-bound given samples
$f_{\max}^*$	Upper-bound on f^* given $(x_i, g_i, f_i)_{i=1}^m$
$f_{\min}^*$	Lower-bound on f^* given $(x_i, g_i, f_i)_{i=1}^m$
$\mathcal{X}_m$	Minimum localization set

Table 2.1: Summary of the different mathematical notations used in this work

Chapter 3

Existing methods

3.1 Convex minimization methods

Many methods have been developed for non-smooth convex problems that use partial or full memory of previous oracle calls. Theoretical developement for multiple of those methods can be found in [19] (notably Kelley, the Level Method and the Ellipsoid Method). This section will often use k as the number of samples, indicating that the method has access to all the previous samples $(x_i, g_i, f_i)_{i=0}^k$ at iteration k .

3.1.1 Kelley's Method

Kelley's method is one of the most natural ideas to leverage the inequalities provided by convexity in particular, the piecewise-linear convex lower bounding function $\hat{f}_k(x)$, which is used as a model. Since that model is a convex interpolant, it contains all the available information on f at iteration k . Kelley's method is an optimistic approach: it searches for the lowest minimizer position inside Q consistent with the data by minimizing $\hat{f}_k(x)$. Kelley's method is a pure cutting plane scheme: at each iteration, one adds a new linear constraint and solves the augmented convex program.

Algorithm 1: Kelley's Method

```
Input:  $Q, x_0, \mathbb{O}(x)$   
for  $k = 0, 1, \dots$  do  
     $f_k, g_k = \mathbb{O}(x_k);$   
     $\hat{f}_k(x) = \max_{0 \leq i \leq k} \{f_i + \langle g_i, x - x_i \rangle\};$   
     $x_{k+1} \in \arg \min_{x \in Q} \hat{f}_k(x);$ 
```

While this procedure produces a sequence of increasingly accurate lower models, it suffers from instability. The model $\hat{f}_k$ is piecewise-linear, and its minimizer over Q depends only on a small subset of active planes (at most $d + 1$). An arbitrarily small perturbation of these planes may lead to large changes in the location of the minimizer, which is an inherent property of linear models. This sensitivity can be exploited by a resisting oracle, which returns subgradients designed to maximally destabilize the method.

Nesterov showed how to exploit such a resisting oracle to derive a lower bound on the number of iterations required to reach a ε -approximate solution. While convergence is linear in terms of number of oracle calls, the rate itself is very disappointing in terms of dependence to dimension.

Theorem 11 (Lower bound for Kelley's method). *There exists a convex function $f : Q \rightarrow \mathbb{R}$ and a compact convex set $Q \subset \mathbb{R}^d$ such that the sequence (x_k) generated by Kelley's method satisfies*

$$\min_{i \leq k} f_i - f^* \geq \left[\left(\frac{1}{4} \right)^{\left(\frac{\sqrt{3}}{2} \right)^{d-1}} \right]^k$$

One can still establish asymptotic convergence of the function values [9] and a convergence rate using assumptions on Q , notably assuming that Q is closed and bounded (and thus has a finite diameter $D = \text{diam}(Q) < \infty$). The upper-bound on the number of iterations is derived using a geometric argument. Indeed, if all iterates (x_k) are in Q then they also are in $B_2(c, \frac{D}{2})$ for some center c . This yields a very rough bound but shows asymptotic convergence when the number of iterations goes to infinity.

Theorem 12. *Consider f to be convex and M -Lipschitz continuous and consider the sequence (x_k) generated by Kelley's method. Then :*

$$\min_i f_i - f^* \leq \frac{2MD}{(k^{\frac{1}{d}} - 1)}$$

Which yields an asymptotic rate of $\mathcal{O}(\frac{MD}{k^{1/d}})$ and is not optimal.

3.1.2 Level Method

The main weakness of Kelley's method is the previously mentioned instability. This issue can be mitigated by restricting the next iterate to lie in a compact level set that does not contain the current iterate. It is defined using the minimum value of the model $\hat{f}_k^*$ and the record value f_k^* defined as follows :

$$\hat{f}_k^* = \min_{x \in Q} \left\{ \max_{0 \leq i \leq k} [f_i + \langle g_i, x - x_i \rangle] \right\}, \quad \tilde{f}_{\max}^* = \min_{0 \leq i \leq k} f_i$$

Clearly $\hat{f}_k^* \leq f^* \leq \tilde{f}_{\max}^*$, since $\hat{f}_k^*$ is a lower bound on f and $\tilde{f}_{\max}^*$ is the convex upper bound on the minimizer value. Using these two quantities, and for any $\alpha \in (0, 1)$, define

$$\ell_k(\alpha) = (1 - \alpha)\hat{f}_k^* + \alpha\tilde{f}_{\max}^*, \quad \mathcal{L}_k(\alpha) = \{x \in Q \mid \hat{f}_k(x) \leq \ell_k(\alpha)\}.$$

The set $\mathcal{L}_k(\alpha)$ is convex and compact. Moreover, since $\ell_k(\alpha) < \tilde{f}_{\max}^*$, none of the previous iterates belong to $\mathcal{L}_k(\alpha)$, and the set depends continuously on the samples $(x_i, g_i, f_i)_{i=1}^m$, which makes it stable with respect to small perturbations. The level method consists in computing the next iterate by projecting the current iterate onto $\mathcal{L}_k(\alpha)$, which stabilizes the sequence.

Algorithm 2: Level Method

Input: $Q, x_0, \alpha \in (0, 1), \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $\hat{f}_k(x) = \max_{0 \leq i \leq k} \{f_i + \langle g_i, x - x_i \rangle\};$
 $\hat{f}_k^* = \min_{x \in Q} \hat{f}_k(x), \quad \tilde{f}_{\max}^* = \min_{0 \leq i \leq k} f_i;$
 $\ell_k = (1 - \alpha)\hat{f}_k^* + \alpha\tilde{f}_{\max}^*;$
 $x_{k+1} = \arg \min_{x \in Q} \left\{ \|x - x_k\|^2 : \hat{f}_k(x) \leq \ell_k \right\};$

A natural stopping criterion follows from the bounds :

$$0 \leq \tilde{f}_{\max}^* - f^* \leq \tilde{f}_{\max}^* - \hat{f}_k^* \quad \text{so that} \quad \tilde{f}_{\max}^* - \hat{f}_k^* \leq \varepsilon \implies \tilde{f}_{\max}^* - f^* \leq \varepsilon$$

An analytically convenient value for α is $\frac{1}{2+\sqrt{2}}$ as it yields the tightest upper bound on the number of iterations required by the level method to find a ε -approximate solution. In practice, this may not be the value of α that yields the best results. The bound is uniform with respect to the dimension and is optimal in order of oracle calls for convex minimization. Another way to make Kelley's method optimal in terms of number of oracle calls is described in [4] but will not be covered here.

Theorem 13 (Guarantee for the Level Method). *Assume that f is convex and M -Lipschitz continuous on Q . For $\alpha = \frac{1}{2+\sqrt{2}}$, the iterates (x_k) generated by the level method satisfy*

$$f(x_k) - f^* \leq \frac{2MD}{\sqrt{k}}, \quad \forall k \geq 2.$$

3.1.3 Proximal Bundle Method

Another way to stabilize Kelley's method is to regularize the cutting-plane model by adding a proximal term that penalizes the distance to the current iterate. This modification prevents overly large or unstable steps and ensures better numerical behavior of the subproblem. In addition, the method incorporates a distinction between serious and null steps. After computing a candidate point y_{k+1} by minimizing the model, the actual decrease in the objective is compared to the decrease predicted by the model (the same piecewise-affine model as Kelley's method).

$$\beta(f(x_k) - \hat{f}_k(y_{k+1})) \leq f(x_k) - f(y_{k+1})$$

The parameter $\beta \in (0, 1)$ determines how closely the predicted improvement must match the actual improvement for a step to be accepted. If the above condition is fulfilled, the step is accepted (serious step). Otherwise, the step is rejected (null step), and the current iterate is left unchanged while the model is refined. The full algorithm is described below.

Algorithm 3: Proximal Bundle Method (PBM)

Input: $Q, x_0, \rho > 0, \beta \in (0, 1), \mathbb{O}(x)$
 $f_0, g_0 = \mathbb{O}(x_0);$
for $k = 0, 1, 2, \dots$ **do**
 $\hat{f}_k(x) = \max_{0 \leq i \leq k} \{f_i + \langle g_i, x - x_i \rangle\};$
 $y_{k+1} = \arg \min_{x \in Q} \left\{ \hat{f}_k(x) + \frac{\rho}{2} \|x - x_k\|^2 \right\};$
 $f_{k+1}, g_{k+1} = \mathbb{O}(y_{k+1});$
 $\Delta_k^{\text{pred}} = f_k - \hat{f}_k(y_{k+1});$
 $\Delta_k^{\text{act}} = f_k - f_{k+1};$
 if $\Delta_k^{\text{act}} \geq \beta \Delta_k^{\text{pred}}$ **then**
 | $x_{k+1} = y_{k+1}$
 else
 | $x_{k+1} = x_k$

Multiple complexity results exist for the proximal bundle method. One can either fix ρ or use an adaptive ρ_k , yielding quite different results. Both are derived in [3].

Theorem 14 (Lipschitz case). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex M -Lipschitz objective function, and consider the proximal bundle method with constant stepsize $\rho_k = \rho$, acceptance parameter $\beta \in (0, 1)$ and $\Delta^2 :=$*

$\sup_k \|x_k - x^*\|^2 < \infty$. Then, for any $0 < \varepsilon \leq f(x_0) - f(x^*)$, the number of descent steps before an ε -minimizer is found is at most

$$\frac{2\rho\Delta^2}{\beta\varepsilon} + \left\lceil \log\left(\frac{f(x_0) - f(x^*)}{\rho\Delta^2}\right) - \log\left(1 - \frac{\beta}{2}\right) \right\rceil_+,$$

and the number of null steps is at most

$$\frac{12\rho M^2\Delta^4}{\beta(1-\beta)^2\varepsilon^3} + \frac{8M^2}{\beta(1-\beta)^2\rho^2\Delta^2}$$

The paper also gives bounds on Δ depending on whether the level sets of f are bounded. Note that if we consider a constrained minimization on Q , taking $\Delta = D$ is a valid upper bound. Choosing $\rho = \varepsilon/\Delta^2$ yields an overall complexity bound of $\mathcal{O}\left(\frac{M^2\Delta^2}{\varepsilon^2}\right)$ which is optimal in the convex non-smooth case. Note that this requires knowing Δ in some way (or a bound on it). This second bound is also derived in [3].

Theorem 15 (Smooth case). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex L -smooth objective function, and consider the proximal bundle method with constant stepsize $\rho_k = \rho$ and acceptance parameter $\beta \in (0, 1)$. Then, for any $0 < \varepsilon \leq f(x_0) - f(x^*)$, the number of descent steps before an ε -minimizer is found is at most*

$$\frac{2\rho\Delta^2}{\beta\varepsilon} + \left\lceil \log\left(\frac{f(x_0) - f(x^*)}{\rho\Delta^2}\right) - \log\left(1 - \frac{\beta}{2}\right) \right\rceil_+,$$

and the number of null steps is at most

$$\frac{4(L + \rho)^3}{(1 - \beta)^2\rho^3} \left(\frac{2\rho\Delta^2}{\beta\varepsilon} + \left\lceil \log\left(\frac{f(x_0) - f(x^*)}{\rho\Delta^2}\right) - \log\left(1 - \frac{\beta}{2}\right) \right\rceil_+ + 1 \right),$$

Choosing $\rho = L$ yields the complexity bound $\mathcal{O}\left(\frac{LD^2}{\varepsilon}\right)$, which matches the standard gradient-descent-type rate (and is suboptimal for the smooth convex case). More can be found about this method in [15] and [17]. An optimal version of this scheme for the smooth convex case will be presented in [subsection 3.2.7](#).

3.1.4 Ellipsoid Method

The Ellipsoid Method is a cutting plane scheme that maintains a localization set in the form of an ellipsoid. At each iteration, the current iterate is chosen as the center of the ellipsoid, and the cutting plane provided by the separation oracle is used to construct a strictly smaller ellipsoid containing the feasible half of the current one.

More precisely, given an ellipsoid $\mathcal{E}(M_k, x_k) = \{x \in \mathbb{R}^d \mid (x - x_k)^\top M_k^{-1}(x - x_k) \leq 1\}$ and a cutting direction $g_k \in \mathbb{R}^d$ (given by the subgradient at point x_k), the half-ellipsoid $\mathcal{E}^+ = \{x \in \mathcal{E}(M_k, x_k) \mid \langle g_k, x_k - x \rangle \geq 0\}$ is contained in a new ellipsoid of strictly smaller volume. Using the updates formulas of the ellipsoid method the new ellipsoid $\mathcal{E}(M_{k+1}, x_{k+1})$ has a smaller volume and the ratio of volumes is upper-bounded by a constant that depends only on the dimension d :

$$\frac{\text{vol}_d \mathcal{E}(M_{k+1}, x_{k+1})}{\text{vol}_d \mathcal{E}(M_k, x_k)} \leq \left(1 - \frac{1}{(d+1)^2}\right)^{d/2},$$

Algorithm 4: Ellipsoid Method

Input: $x_0 \in \mathbb{R}^d$, $R > 0$, $\mathbb{O}(x)$

$M_0 = R^2 I_d$;

for $k = 0, 1, \dots$ **do**

$$\left[\begin{array}{l} f_k, g_k = \mathbb{O}(x_k); \\ x_{k+1} = x_k - \frac{1}{d+1} \cdot \frac{M_k g_k}{\sqrt{g_k^\top M_k g_k}}; \\ M_{k+1} = \frac{d^2}{d^2 - 1} \left(M_k - \frac{2}{d+1} \cdot \frac{M_k g_k g_k^\top M_k}{g_k^\top M_k g_k} \right); \end{array} \right.$$

Theorem 16 (Convergence of the Ellipsoid Method). *Let f be M -Lipschitz continuous on $B_2(x^*, R)$. Then for any $k \geq 1$, the iterates of the Ellipsoid Method satisfy*

$$\min_{j \leq k} f(x_j) - f^* \leq MR \cdot \left(1 - \frac{1}{(d+1)^2} \right)^{\frac{k}{2}} \cdot \left[\frac{\text{Vol}_d B_2(x_0, R)}{\text{Vol}_d Q} \right]^{\frac{1}{d}}$$

To produce an ε -solution, the method requires $\mathcal{O}(d^2 \log \varepsilon^{-1})$ oracle calls. While this complexity is not optimal with respect to the dimension in term of oracle calls, the dependence in ε is, as the lower bound is $\mathcal{O}(d \log \varepsilon^{-1})$. The ellipsoid method can be very efficient on low dimensional convex problems and offers an explicit update rule computed in $\mathcal{O}(d^2)$ arithmetic operations.

3.1.5 Analytic Center Cutting-Plane Method

An overview of ACCPM and convergence proofs can be found in [2]. The Analytic Center Cutting-Plane Method uses the analytic center of the localization set as the next query point for the separation oracle, as the name suggests. It starts with a polyhedral localization set P_0 . The analytic center x^* of the polyhedron P_k is defined as the minimizer of the logarithmic barrier $\phi_k(x)$:

$$\phi_k(x) = - \sum_{i=1}^m \log(b_i - a_i^\top x)$$

The analytic center is, in some sense, the point furthest away from boundaries. Since the logarithmic barrier is self-concordant, the analytic center is usually found by minimizing the barrier function using Newton's method, which is very efficient on self-concordant functions. Each cut increases the computational complexity of the Newton's method computations, making it important to restrict the number of cuts used for the computations. ACCPM naturally defines the following irrelevance measure for each constraint :

$$\eta_i = \frac{b_i - a_i^\top x^*}{\sqrt{a_i^\top \nabla^2 \phi_k(x^*)^{-1} a_i}}$$

If $\eta_k > m$ then the constraint is certainly irrelevant. Typically only $\mathcal{O}(d)$ constraints are kept. The η_k are easily derived from the computations required in the inner Newton's method.

The convergence guarantees provided by this algorithm are usually not expressed in terms of functional residual but in terms of the largest inscribed ball that can be inside the polyhedral set.

Theorem 17. *Let $P_0 = \{x \in \mathbb{R}^d : \|x\|_\infty \leq R/2\}$ and let P_k be the localization set after k ACCPM iterations. Then the maximal number of iterations after which P_k can still contain a Euclidean ball of radius $r < R/2$ is bounded by*

$$k \leq \max \left\{ 2d, d^2 \alpha \left(\frac{2r}{R} \right) \right\}, \quad \frac{\alpha(\rho)}{\log(1 + \alpha(\rho))} = \frac{1}{2\rho^4}.$$

Algorithm 5: Analytic Center Cutting-Plane Method

Input: $P_0 = \{x \mid j = 1, \dots, N : c_j^\top x \leq d_j\}, \mathbb{O}(x)$

Initialize $\mathcal{B}_0 = \emptyset$;

for $k = 0, 1, \dots$ **do**

$$\phi_k(x) = \sum_{j=1}^N \log(d_j - c_j^\top x) + \sum_{j \in \mathcal{B}_k} \log(b_j - a_j^\top x);$$

$$x_k = \arg \min_x -\phi_k(x);$$

$$f_k, g_k = \mathbb{O}(x_k);$$

$$f_k^* = \min_i f_i;$$

$$a_k = g_k, \quad b_k = \langle g_k, x_k \rangle - (f_k - f_k^*);$$

$$\mathcal{B}_{k+1} = \mathcal{B}_k \cup \{k\};$$

Remove constraints with high η_j from $\mathcal{B}_{k+1}$;

Equivalently, for any k larger than this bound, P_k cannot contain a Euclidean ball of radius r .

ACCPM also provides a lower bound on the function minimum. This lower bound is built from the dual problem of the inner problem of Kelley's method. Using the analytic center at a specific iteration, it is possible to build a dual solution that provides a lower bound on the function minimum. This bound is not tight but can be used as a stopping criterion. More information about this bound can be found in [2] [18].

3.1.6 Vaidya's Volumetric Cutting-Plane Method

Vaidya's method can be viewed as a refinement of ACCPM. A paper overview of this method is found in [21]. Instead of using the analytic center of the localization set as query point, it uses the volumetric center, which yields improved complexity bounds while still relying on a logarithmic barrier framework. The Hessian of the logarithmic barrier is given by the following :

$$H_k(x) = \nabla^2 \phi_k(x) = \sum_{i=1}^m \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}.$$

The Hessian $H_k(x)$ defines the local Dikin ellipsoid

$$\mathcal{E}_k(x) = \{y \in \mathbb{R}^d \mid (y - x)^\top H_k(x)(y - x) \leq 1\}.$$

This ellipsoid is contained in the polyhedron used to define it. The volumetric center is the point around which that inner ellipsoid has the largest volume. It is found by minimizing the following :

$$V_k(x) = \frac{1}{2} \log \det H_k(x).$$

This quantity better captures the geometry of the localization set than the analytic center alone, yielding even more central query points. Similarly to ACCPM, Vaidya's method provides a metric that quantifies the leverage of each constraint.

$$\sigma_i(x) = \frac{a_i^\top H_k(x)^{-1} a_i}{(b_i - a_i^\top x)^2}, \quad \sum_{i=1}^m \sigma_i(x) = d$$

A constraint is considered redundant when the associated σ_i is smaller than some threshold. Similarly to ACCPM, when the desired number of constraints is reached, one removes the constraint with the

Algorithm 6: Vaidya's Volumetric Cutting-Plane Method

Input: $\mathbb{O}(x)$, P_0
for $k = 0, 1, \dots$ **do**
 $H_k(x) = \sum_{i=1}^m \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2};$
 $V_k(x) = \frac{1}{2} \log \det H_k(x);$
 $x_k = \arg \min_x V_k(x);$
 $f_k, g_k = \mathbb{O}(x_k);$
 $a_{k+1} = g_k;$
 $b_{k+1} = -(f_k^* - f_k + \langle g_k, x_k \rangle);$
 Remove redundant constraints with $\sigma_i(x_k);$

smallest σ_i . The method is described in the following algorithm.

Compared with ACCPM, Vaidya's method achieves better oracle complexity bounds while still maintaining only $\mathcal{O}(d)$ active constraints (see [1]). It is however a lot more expensive computationally and usually requires keeping a large amount of constraints to work well. Multiple approaches showed ways to reduce the oracle complexity. The main interest of Vaidya's method is largely theoretical and in practice the computational cost is very high compared to other methods as it requires applying Newton's method to the log det objective and it requires solving multiple dense linear systems per newton iteration, making ACCPM the preferred choice over Vaidya's method.

3.1.7 Summary

Nesterov's analysis indicates that both Vaidya's method and the ACCPM achieve an oracle complexity of order $\mathcal{O}(d(\log(\varepsilon^{-1}))^p)$, with $p \in \{1, 2\}$, for computing an ε -approximate solution. Another source describing convergence of such methods and the optimality tradeoff between memory and number of oracle calls is [26] (as well as current open problems in that field). However, explicit bounds on the functional residual are not typically provided. In contrast to first-order methods, these results are stated in terms of the number of oracle calls required to solve a feasibility problem, which can be linked to finding an approximate minimizer. Moreover, despite their favorable iteration complexity, both methods involve significantly higher per-iteration computational costs than, for example, the ellipsoid method. In the following, we adopt the standard viewpoint of the cutting-plane literature and report complexity bounds in $\mathcal{O}$ -notation in terms of oracle calls. For the purpose of comparing subproblem costs, we assume that the feasible set Q is polyhedral, so that Kelley's method reduces to solving a linear program at each iteration.

3.2 Smooth convex minimization methods

3.2.1 Gradient Method

The well known gradient descent algorithm is found by minimizing the quadratic upper-bound provided by smoothness.

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^d} \left\{ f(x_k) + \langle g_k, x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 \right\} = x_k - \frac{1}{L} g_k$$

Method Name	Complexity Bound	Subproblem cost
Kelley	$\mathcal{O}\left(\left[\frac{MD}{\varepsilon}\right]^d\right)$	1 LP (growing size, $\sim k$ constraints)
Level	$\mathcal{O}\left(\frac{M^2 D^2}{\varepsilon^2}\right)$	1 LP + 1 QP (size $\sim k$, moderate cost)
PBM	$\mathcal{O}\left(\frac{M^2 D^2}{\varepsilon^2}\right)$	1 strongly convex QP (size $\sim k$, well-conditioned)
Ellipsoid	$\mathcal{O}(d^2 \log(\varepsilon^{-1}))$	Rank-1 matrix update ($\sim d^2$, very cheap)
ACCPM	$\mathcal{O}(d \log(\varepsilon^{-1}))$	Newton solve (log-barrier, dense system $\sim d^3$)
Vaidya	$\mathcal{O}(d \log(\varepsilon^{-1}))$	Newton solve (volumetric barrier, dense system $\sim d^3$)

Table 3.1: Summary of existing convex non-smooth optimization methods: convergence vs practical subproblem cost

There are tight upper bounds on the number of iterations for the gradient method. For smooth convex functions, choosing a step size larger than $\frac{1}{L}$ yields a better bound. The optimal step size is found by solving a performance estimation problem (PEP), see [24] for further details. The following theorem gives

Algorithm 7: Gradient Method (GD)

Input: $x_0, L, \alpha, \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $x_{k+1} = x_k - \frac{\alpha}{L} g_k;$

an analytical bound on function residual, for it's derivation see [5].

Theorem 18 (Guarantee for the gradient method). *Assume that f is a L -smooth convex function. Then for a step size of $\frac{h}{L}$, the iterates (x_k) satisfy :*

$$f(x_k) - f(x^*) \leq \frac{LR^2}{4kh + 2}$$

The lower bound is not optimal in terms of oracle calls since it doesn't match the order of convergence of Theorem 9. The gradient method generally enjoys a convergence rate of $\mathcal{O}\left(\frac{1}{k}\right)$.

3.2.2 Fast Gradient Method

The gradient method achieves a convergence rate of $\mathcal{O}\left(\frac{1}{k}\right)$, which is not optimal in light of the lower bound given in Theorem 9. Nesterov's Fast Gradient Method (FGM) improves upon this by introducing an extrapolation step, allowing the method to achieve the optimal rate of $\mathcal{O}\left(\frac{1}{k^2}\right)$. The method combines standard gradient steps with a momentum term that incorporates information from previous iterates.

Algorithm 8: Fast Gradient Method (FGM)

Input: $x_0, L, \mathbb{O}(x)$
 $y_0 = x_0;$
 $t_0 = 1;$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $y_{k+1} = x_k - \frac{1}{L} g_k;$
 $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2};$
 $x_{k+1} = y_{k+1} + \frac{t_k - 1}{t_{k+1}} (y_{k+1} - y_k);$

The sequence (y_k) corresponds to gradient steps, while (x_k) incorporates an extrapolation term that accelerates convergence.

Theorem 19 (Worst-case bounds for FGM). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be convex and L -smooth, and let (x_k) and (y_k) be generated by the Fast Gradient Method. Then for any $k \geq 1$,*

$$f(y_k) - f^* \leq \frac{LR^2}{2t_{k-1}^2} \leq \frac{2LR^2}{(k+1)^2}, \quad f(x_k) - f^* \leq \frac{LR^2}{2t_k^2} \leq \frac{2LR^2}{(k+2)^2},$$

While this bound achieves the optimal rate of convergence in $\mathcal{O}$ notation, it is still shy of the optimal lower bound in terms of a constant factor.

3.2.3 Optimized Gradient Method

While the Fast Gradient Method achieves the optimal convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$, it does not match the constants of the lower bound. The Optimized Gradient Method (OGM), introduced via performance estimation techniques, improves this constant while preserving the same oracle complexity. An overview of this method can be found in [13]. It can be interpreted as a refinement of FGM where the extrapolation coefficients are chosen optimally with respect to the worst-case performance. OGM combines gradient steps with a more complex extrapolation mechanism.

Algorithm 9: Optimized Gradient Method (OGM)

Input: $x_0, L, \mathbb{O}(x)$

$y_0 = x_0;$

$\theta_0 = 1;$

for $k = 0, \dots, N - 1$ **do**

$f_k, g_k = \mathbb{O}(x_k);$

$y_{k+1} = x_k - \frac{1}{L}g_k;$

$\theta_{k+1} = \begin{cases} \frac{1+\sqrt{1+4\theta_k^2}}{2}, & k \leq N-2, \\ \frac{1+\sqrt{1+8\theta_k^2}}{2}, & k = N-1 \end{cases};$

$x_{k+1} = y_{k+1} + \frac{\theta_k-1}{\theta_{k+1}}(y_{k+1} - y_k) + \frac{\theta_k}{\theta_{k+1}}(y_{k+1} - x_k);$

As in FGM, the sequence (y_k) corresponds to gradient steps. The update of (x_k) includes both the usual extrapolation term and an additional correction term involving (x_k) , which is responsible for the improved worst-case performance.

Theorem 20 (Worst-case bounds for OGM). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be convex and L -smooth, and let (x_k) and (y_k) be generated by OGM. Then for any $1 \leq k \leq N$,*

$$f(y_k) - f^* \leq \frac{LR^2}{4\theta_{k-1}^2} \leq \frac{LR^2}{(k+1)^2},$$

Compared to FGM, OGM achieves the same $\mathcal{O}\left(\frac{1}{k^2}\right)$ convergence rate but improves the constant by a factor of two on the primary sequence (y_k) . This makes OGM optimal near-optimal in terms of worst-case constants among first-order methods for smooth convex minimization. Even better constants can be achieved by methods described in [10].

3.2.4 Gradient Method with Memory

The Gradient Method with Memory (GMM), introduced by Nesterov and Florea in [20], is a first-order method that explicitly exploits both convexity and smoothness by constructing a global model of the

objective from previously computed gradients. In contrast with the gradient method and its accelerated variants, which rely on local information, GMM builds the convexity piecewise-linear model $\hat{f}_k$ and stabilizes the minimization by incorporating a regularization term similar to the quadratic upper-bound.

Algorithm 10: Gradient Method with Memory (GMM)

Input: $x_0, L, \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $\hat{f}_k(x) = \max_{i \leq k} \{f_i + \langle g_i, x - x_i \rangle\};$
 $i_k \leftarrow 0;$
 while true do
 $L \leftarrow 2^{i_k} L_k;$
 $x_k^+ \leftarrow \arg \min_{x \in \mathbb{R}^d} \left\{ \hat{f}_k(x) + \frac{L}{2} \|x - x_k\|^2 \right\};$
 if $f(x_k^+) \leq \hat{f}_k(x_k^+) + \frac{L_k}{2} \|x_k^+ - x_k\|^2$ **then**
 break;
 $i_k \leftarrow i_k + 1;$
 $x_{k+1} \leftarrow x_k^+;$
 $L_{k+1} \leftarrow 2^{i_k-1} L_k;$

The algorithm relies on a line-search approach which is the main upgrade compared to the regular gradient method. Using the full sample model allows for more accurate curvature estimation, yielding lower L_k values than regular line search approaches for the gradient method. It does not offer a better theoretical bound than the gradient method but can be faster in many cases.

The average number of oracle calls required by the line-search procedure is bounded by 2. A warm start is used in order to compute x_k^+ efficiently when L_k changes. For that warm start to be efficient, it solves the problem using the dual formulation rather than primal. The dual formulation of this problem is as follows :

$$\lambda^* = \arg \max_{\lambda \in \Delta_m} \left\{ \sum_{i=1}^m \lambda_i (f_i + \langle g_i, x_k - x_i \rangle) - \frac{1}{2L_k} \left\| \sum_{i=1}^m \lambda_i g_i \right\|^2 \right\}.$$

This is solved using an first order optimization algorithm like Frank-Wolfe and the primal solution is then recovered using :

$$x_k^+ = x_k - \frac{1}{L_k} \sum_{i=1}^m \lambda_i^* g_i.$$

3.2.5 Improved Gradient Method with Memory

A natural extension of GMM replaces the convex piecewise-linear model with the optimal smooth lower bound F_l . This approach was developed in [7] by Nesterov and Florea. The objective becomes the optimal smooth convex lower bound augmented with the same proximal term. Using the dual formulation of $F_l(x)$ this yields :

$$x_{k+1} = \arg \min_{y \in \mathbb{R}^d} \max_{\lambda \in \Delta_m} \left\{ \sum_{i=1}^m \lambda_i \psi_i(y) - \frac{1}{2L} \left\| \sum_{i=1}^m \lambda_i g_i \right\|^2 + \frac{L_k}{2} \|y - x_k\|^2 \right\}.$$

Similarly to GMM, a line-search procedure is used to determine the effective step size, ensuring a descent condition analogous to the quadratic upper bound, which enables convergence guarantees.

Algorithm 11: Improved Gradient Method with Memory (IGMM)

Input: $x_0, L, \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 while true do
 $\lambda^* = \arg \min_{\lambda \in \Delta_m} \left\{ \left(\frac{1}{2L} + \frac{1}{2L_k} \right) \left\| \sum_{i=1}^m \lambda_i g_i \right\|^2 - \sum_{i=1}^m \lambda_i \psi_i(x_k) \right\};$
 $x_k^+ = x_k - \frac{1}{L_k} \sum_{i=1}^m \lambda_i^* g_i;$
 $\omega_k = \sum_{i=1}^m \lambda_i^* \psi_i(x_k) - \left(\frac{1}{2L_k} + \frac{1}{2L} \right) \left\| \sum_{i=1}^m \lambda_i^* g_i \right\|^2;$
 if $L_k \geq L$ **then**
 $L_k = L;$
 $x_{k+1} = x_k - \frac{1}{L} g_k;$
 break;
 if $f(x_k^+) \leq \omega_k$ **then**
 $x_{k+1} = x_k^+;$
 break;
 $L_k = 2 \cdot L_k;$
 $L_{k+1} = \frac{L_k}{2};$

Theorem 21 (Worst-case bounds for IGMM). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be convex and L -smooth, and let (x_k) be generated by IGMM. Then for any $k \geq 1$,*

$$f(x_k) - f(x^*) \leq \frac{R^2}{2 \left(\sum_{i=1}^k \frac{1}{L_i} \right)} \leq \frac{LR^2}{2k},$$

where (L_i) are the effective curvature produced by the line-search.

Although the worst-case rate remains $\mathcal{O}(1/k)$, identical to the standard gradient method, the improved lower model yields significantly better practical performance than regular gradient descent. The tighter approximation of the objective allows the method to take more informed steps, often leading to faster decrease in function values. Since the lower bound is better than GMM the results are as well. Note that the line search requires multiple oracle calls, which is the performance measure being used in this work.

3.2.6 Optimized Gradient Method with Memory

The IGMM method can be accelerated using some of the classical acceleration tricks [7]. Let us directly address the case of the Optimized version, skipping the Accelerated Gradient Method with Memory. The method uses primal dual estimate functions in order to achieve optimized acceleration similar to OGM. This variant with memory also uses a line-search approach in order to leverage the improved model, the line-search is done on A_k here, which is analog to the inverse step size. This line search has the advantage of not using additional oracle calls but leverages the improved model to find a bigger valid step size.

Algorithm 12: Optimized Gradient Method with Memory (OGMM)

Input: $x_0, L, N, \mathbb{O}(x)$
 $v_0 = x_0, A_0 = 0;$
for $k = 0, 1, \dots$ **do**
 $a_{k+1} = \frac{1}{2L} (1 + \sqrt{1 + 4LA_k});$
 $y_{k+1} = \frac{A_k x_k + a_{k+1} v_k}{A_k + a_{k+1}};$
 $f_{k+1}, g_{k+1} = \mathbb{O}(y_{k+1});$
 $x_{k+1} = y_{k+1} - \frac{1}{L} g_{k+1};$
 $e_{k+1} = f_{k+1} - \frac{1}{2L} \|g_{k+1}\|^2;$
 $A_{\text{valid}} = A_k + a_{k+1};$
 $\lambda_{\text{valid}} = \frac{1}{A_k + a_{k+1}} [A_k \ a_{k+1} \ 0 \ \dots \ 0];$
 $A = A_{\text{valid}};$
 for $i = 0, \dots, N$ **do**
 $\lambda_{k+1} = \arg \min_{\lambda \in \Delta_{k+1}} \left\{ \left(\frac{A}{2} + \frac{1}{2L} \left\| \sum_j \lambda_j g_j \right\| \right) - \sum_j \psi_j(x_0) \right\};$
 $\omega^* = \sum_j \psi_j(x_0) - \frac{A_k}{2} \left\| \sum_j \lambda_j g_j \right\|;$
 if $\omega^* \leq e_{k+1}$ **then**
 break;
 $A_{\text{valid}} = A;$
 $\lambda_{\text{valid}} = \lambda;$
 $A = A + 2 \frac{\omega^* - e_{k+1}}{\left\| \sum_j \lambda_j g_j \right\|^2};$
 $A_{k+1} = A_{\text{valid}};$
 $v_{k+1} = x_0 - A_{k+1} \sum_j \lambda_{\text{valid},j} g_j;$

Theorem 22 (Convergence of OGMM). *Assume that f is L -Smooth convex. Let (x_k) be the sequence generated by the Optimized Gradient Method with Memory (OGMM), and let (A_k) be the sequence constructed by the algorithm. Then, for all $k \geq 1$, the iterates satisfy*

$$f(x_k) - f^* \leq \frac{\|x_0 - x^*\|^2}{2A_k}, \quad A_k \geq \frac{k(k+1)}{2L}$$

This theorem show that OGMM achieves the same optimal worst-case rate as regular OGM. Due to the line-search procedure, the values A_k can be strictly larger than their lower bound, yielding improved instance-dependent convergence beyond the worst-case guarantee.

3.2.7 Smooth Proximal Bundle Method

The proximal bundle method (PBM) presented in the convex non-smooth section can be adapted to exploit the interpolation conditions available in the smooth setting. In contrast to the classical PBM, which uses a piecewise linear lower model, the accelerated variant constructs a smooth lower model by solving a convex QCQP. It was developed in [6]. This changes the bundle subproblem into a quadratic program while preserving the proximal bundle structure. At the same time, the resulting subproblem can be interpreted as an inexact proximal step for the smooth objective, where the true function is replaced by its smooth lower interpolating model. This is precisely the point of view adopted in the paper: the accelerated analysis is obtained by relating the serious steps of a-PBM to an accelerated inexact proximal point scheme. In this sense, the algorithm is not treated as fundamentally different from a proximal method, but rather as a proximal method with an approximate oracle and a bundle-based construction of

the proximal subproblem. This interpretation is what makes the acceleration argument possible.

Algorithm 13: Accelerated Proximal Bundle Method (a-PBM)

Input: $x_0, \rho > 0, C, L, \mathbb{O}(x)$

$\zeta_0 = x_0, \quad t_0 = 1;$

$f_0 = f(x_0), \quad g_0 = \nabla f(x_0);$

for $k = 0, 1, 2, \dots$ **do**

$$\hat{f}_k(x) = \max_{0 \leq i \leq k} \left\{ f_i + \langle g_i, x - x_i \rangle + \frac{\rho^2}{2L} \left\| x - x_k + \frac{1}{\rho} g_i \right\|^2 \right\}$$

$$y_{k+1} = \arg \min_x \left\{ \hat{f}_k(x) + \frac{\rho}{2} \|x - x_k\|^2 \right\};$$

$$f_{k+1}, g_{k+1} = \mathbb{O}(y_{k+1});$$

if $C \|y_{k+1} - y_k\| \leq \|x_k - y_{k+1}\|$ **then**

$$t_{k+1} \leftarrow \frac{1 + \sqrt{1 + 4t_k^2}}{2};$$

$$\zeta_{k+1} \leftarrow y_{k+1} - \frac{1}{\rho} g_{k+1};$$

$$x_{k+1} \leftarrow \zeta_{k+1} + \frac{t_k - 1}{t_{k+1}} (\zeta_{k+1} - \zeta_k); \quad // \text{ serious step}$$

else

$$x_{k+1} \leftarrow x_k;$$

$$\zeta_{k+1} \leftarrow \zeta_k;$$

$$t_{k+1} \leftarrow t_k; \quad // \text{ null step}$$

Compared with the classical PBM, the main differences are twofold. First, the bundle subproblem is no longer a standard linear model with a proximal term, but a convex QCQP that builds a smooth lower interpolating model. Second, the proximal center is updated using a Nesterov-type extrapolation through the auxiliary sequences (ζ_k, t_k) , rather than being updated directly after a serious step. In particular, the point y_{k+1} should be viewed as an approximate proximal point for the smooth objective, in the sense that it is obtained by minimizing a smooth lower model plus a proximal term centered at x_k . The accelerated analysis is then carried out exactly by exploiting this proximal interpretation.

$$y_{k+1} \approx \text{prox}_{f/\rho}(x_k) = \arg \min_x \left\{ f(x) + \frac{\rho}{2} \|x - x_k\|^2 \right\},$$

The paper makes this connection explicit by showing that the serious steps satisfy an inexactness criterion for the associated proximal subproblem. The null-step test is designed so that, after sufficiently many null steps, this criterion holds and the proximal-point acceleration mechanism can be applied. As a consequence, the convergence analysis of a-PBM follows the same accelerated proximal-point logic as inexact Nesterov-type schemes, while retaining the bundle structure of the method.

Theorem 23 (Oracle complexity of a-PBM). *Assume that f is convex and L -smooth. Then a-PBM reaches an ε -optimal solution in at most*

$$k \leq \left(\sqrt{\frac{2\rho \|x_0 - x^*\|^2}{\varepsilon}} + 1 \right) \left(\frac{5 \log(1/\varepsilon) + \log(M/A)}{\log \left(1 + \left(\frac{L}{\rho} + \frac{8L^2}{\rho^2} + \frac{16L^3}{\rho^3} \right)^{-1} \right)} + 1 \right)$$

iterations, where M and A are constants appearing in the proof and are covered in more detail in [6].

In particular, the method attains an accelerated complexity of order $\mathcal{O}(\sqrt{1/\varepsilon} \log(1/\varepsilon))$, and the logarithmic factor comes from the bound on the number of consecutive null steps. The method is not optimal but it is near optimal, being shy of the lower bound rate by only a factor $\log(1/\varepsilon)$.

3.2.8 Summary

Contrary to the nonsmooth convex setting, where several methods are known to attain the optimal finite-dimensional lower complexity bound, no first-order method specifically tailored to the smooth convex case is currently known to achieve such a bound. Existing algorithms, including accelerated gradient methods and their variants, are typically designed to attain optimal worst-case guarantees that are uniform with respect to the problem dimension. As a result, their complexity bounds do not exploit potential structure arising from finite-dimensional interpolation conditions.

Method Name	Complexity Bound	Per iteration cost
GD	$\frac{LR^2}{2k}$	$\mathcal{O}(d)$
FGM	$\frac{2LR^2}{(k+1)^2}$	$\mathcal{O}(d)$
OGM	$\frac{LR^2}{(k+1)^2}$	$\mathcal{O}(d)$
GMM	$\frac{2k}{R^2}$	dual QP, size $\sim k$
IGMM	$2 \left(\sum_{i=1}^k \frac{1}{L_i} \right)$	smooth lower QCQP, size $\sim k$
OGMM	$\frac{\ x_0 - x^*\ ^2}{2A_k}$	primal-dual QP, size $\sim k$
a-PBM	$\mathcal{O}\left(\frac{1}{\sqrt{\varepsilon}} \log \frac{1}{\varepsilon}\right)$	QCQP size $\sim k$

Table 3.2: Summary of existing smooth convex optimization methods and their convergence rates compared with their computational cost.

The main objective of this thesis is to investigate whether the smooth convex interpolation framework can be leveraged to design new methods with improved convergence behavior. In particular, the goal is to explore algorithms that exploit these interpolation conditions directly, with the aim of achieving faster, potentially linear, convergence rates, in contrast with the classical sublinear rates obtained from standard convex inequalities.

Chapter 4

Proposed Methods

The main goal of this work is to explore the possibilities that interpolation conditions offer for memory-based methods. While finding a method that is efficient is interesting, the main contribution is to highlight what ideas are worth exploring further and identify possible dead ends.

4.1 Minimizing extremal functions

A natural way to exploit the interpolating upper and lower bounding function is to use their minimizer as the next iterate. A pessimistic approach would consist using the minimizer of the upper bounding function, as it would guarantee a decrease at each iteration. On the other hand, an optimistic approach would choose the point achieving the lowest possible minimum value compatible with the data.

4.1.1 Pessimistic Approach

Let's first study what happens when minimizing the upper-bounding function. The dual formulation is the easiest approach to work with.

$$F_h(x) = \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i s_i + \frac{L}{2} \left\| x - \sum_i \lambda_i u_i \right\|^2 \right\}, \quad \nabla F_h(x) = L \left[x - \sum_i \lambda_i^* u_i \right]$$

With λ_i^* being the minimizer of the quadratic program used to define $F_h(x)$. Since the minimization over x is unconstrained and the function F_h is convex, it is minimized at the point x_h^* such that $\nabla F_h(x_h^*) = 0$, which yields the following expression (back in usual notation) :

$$x_h^* = \sum_i \lambda_i^*(x) \left(x_i - \frac{1}{L} g_i \right), \quad F_h(x_h^*) = \min_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \left(f_i - \frac{1}{2L} \|g_i\|^2 \right) \right\}$$

Notice that the minimization defining F_h does not depend on x at that point and the term is linear with respect to λ . As λ is defined on a simplex, the minimum is attained on a vertex, or an edge between 2 vertices with equal value. In any case, taking one of the indices achieving the minimum value i^* and setting $\lambda_{i^*} = 1$ and $\forall j \neq i^* : \lambda_j = 0$ yields a valid minimizer. In that case, x_h^* is just a gradient step away from x_{i^*} . If the pessimistic approach is used at each iteration, it is possible to show that the method reduces to gradient descent. Indeed, at the starting point x_0 , the next point will be chosen by gradient

descent and afterwards, the descent lemma shows :

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L} \|\nabla f(x_k)\|^2 \implies f_{k+1} - \frac{1}{2L} \|g_{k+1}\|^2 < f_k - \frac{1}{2L} \|g_k\|^2$$

This in terms shows by induction that the last point always has the lowest possible value of $f_k - \frac{1}{2L} \|g_k\|^2$, making the pessimistic approach, reduces exactly to gradient descent, regardless of the number of previously sampled points. This is a consequence of the structure of F_h : at its minimizer, the optimal weights λ^* concentrate on the single index i^* achieving $\min_i \{f_i - \frac{1}{2L} \|g_i\|^2\}$, and the next iterate is simply a gradient step from x_{i^*} . Moreover, the descent lemma implies that this index is always the most recent iterate x_k , so no memory is exploited. In this sense, gradient descent is the unique method that maximises the guaranteed decrease in function value at each step, independently of past iterates. This also reveals a fundamental limitation of the pessimistic approach : by optimising for the worst-case decrease at each step, it discards all accumulated information and cannot benefit from memory.

4.1.2 Optimistic Approach

The optimistic approach consists in choosing the point providing the lowest possible functional value compatible with the data as next iterate. This point, the lowest possible minimizer, is obtained by minimizing $F_l(x)$. The optimization problem yielding such a point is found directly from the primal formulation of $F_l(x)$ and imposing the gradient value $\nabla F_l(x) = 0$:

$$\min_{x \in \mathbb{R}^d} F_l(x) = \min_{x \in \mathbb{R}^d, t \in \mathbb{R}} t \quad \text{s.t.} \quad \psi_i(x) \leq t, \quad \forall i.$$

From the primal formulation alone, it is hard to know when the minimization is bounded and well defined. Indeed, there is no constraint on directions not in the span of the past gradients. Even in directions lying in this span, the minimization may still be unbounded. A clearer condition is obtained from the dual formulation using weak duality : if the dual is infeasible and the primal is feasible, which is always the case here by taking t sufficiently large for a given x , then the primal is unbounded. Using the Lagrangian associated with the constraints $\psi_k(x) \leq t$. Using $\nabla \psi_i(x) = g_i$ one obtains

$$\mathcal{L}(x, t, \lambda) = t + \sum_i \lambda_i (\psi_i(x) - t), \quad \nabla_x \mathcal{L}(x, t, \lambda) = \sum_i \lambda_i g_i = 0$$

Minimization with respect to t yields the condition $\sum_i \lambda_i = 1$, so that $\lambda \in \Delta_m$. Moreover, the stationarity with respect to x gives $\sum_i \lambda_i g_i = 0$. If no $\lambda \in \Delta_m$ satisfies this relation, then the dual is infeasible and the primal problem is unbounded. Plugging that condition back into the dual formulation or the lagrangian yields a simpler expression for the dual problem :

$$\min_{x \in \mathbb{R}^d} F_l(x) = \max_{\substack{\lambda \in \Delta_m \\ \sum_i \lambda_i g_i = 0}} \lambda^\top b.$$

The point x where this minimum is achieved is the optimal dual multiplier linked to the equality constraint on the gradients. This approach is interesting, but minimizing the lower bound alone is not enough, as it is often unbounded. It can yield promising results and will be studied in further detail in [chapter 5](#).

4.2 Improved convex techniques

Now, let's discuss what can be done by using smoothness as a strengthened convexity cutting plane for classical convex optimization methods.

4.2.1 Smooth Level Method

The level method can be strengthened using the better cutting planes smoothness provide. Those planes provide strengthened bounds on possible minimizer value. The upper level value now becomes $f_{\max}^* = \min_i s_i$ and on the lower bound side, the piecewise-affine model is also modified to account for smoothness, yielding :

$$\phi_k(x) = \max_{0 \leq i \leq k} \left\{ f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \right\}$$

Note that contrary to the convex case, this is piece-wise affine model provides a bound on the minimizer value at a given x and not a lower bound on the function value at a given x . As such, $\phi_k(x^*) \leq f(x^*)$ but $\phi_k(x) \not\leq f(x)$, at least not in general. These coincide at the point minimizing $F_l(x)$ as described in [subsection 4.1.2](#). The level is defined using $f_{\min}^* = \min_{x \in Q} F_l(x)$ as follows :

$$l_k(\alpha) = (1 - \alpha)f_{\min}^* + \alpha f_{\max}^*$$

The algorithm incorporating those changes is described below. Using this more accurate information on the minimizer values could possibly lead to faster convergence, but it is not guaranteed.

Algorithm 14: Smooth Level Method (S-Level)

Input: $x_0, Q, L, \alpha, \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $\phi_k(x) = \max_{0 \leq i \leq k} f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2;$
 $f_{\min}^* = \min_{x \in Q} \phi_k(x);$
 $f_{\max}^* = \min_{0 \leq i \leq k} f_i - \frac{1}{2L} \|g_i\|^2;$
 $l_k = (1 - \alpha)f_{\min}^* + \alpha f_{\max}^*;$
 $x_{k+1} = \arg \min_{x \in Q} \{ \|x - x_k\|^2 : \phi_k(x) \leq l_k \};$

4.2.2 Smooth Ellipsoid Method

The deep cut ellipsoid method is a well known method that leverages stronger cuts than simple separation. That method is initially a convex optimization method and more information can be found in [8]. It uses a slightly different ellipsoid update than the regular method but, contrary to the original version, the enclosing ellipsoid is not the tightest ellipsoid around the feasible set. Let's consider an ellipsoid region of the following form :

$$\mathcal{E}_k = \{x | (x - x_k)^\top M_k^{-1} (x - x_k) \leq 1\}$$

The next iterate x_{k+1} and the next matrix M_{k+1} are build from the shifted center c_k instead of x_k . From the smooth convex cutting planes, the shifted center is :

$$c_k = x_k - \left(\frac{f_k - f_{\max}^*}{\|g_k\|^2} + \frac{1}{2L} \right) g_k$$

From this, one can write the shrinking factor α_k , which is quantifying the gain compared a central cut ($\alpha_k = 0$ for a central cut).

$$\alpha_k = \frac{g_k^\top x_k - g_k^\top c_k}{\sqrt{g_k^\top M_k g_k}} = \frac{f_k - f_{\max}^* + \frac{1}{2L} \|g_k\|^2}{\sqrt{g_k^\top M_k g_k}} \leq 1$$

This factor is always less than 1, otherwise the minimum cannot be inside the ellipsoid, which contradicts the initialization. The deep cut update yields the following algorithm.

Algorithm 15: Smooth Ellipsoid Method (S-Ellipsoid)

Input: $x_0 \in \mathbb{R}^d$, $R > 0$ such that $x^* \in B_2(x_0, R)$, $\mathbb{O}(x)$, L

$M_0 = R^2 I_d$;

for $k = 0, 1, \dots$ **do**

$$\left[\begin{array}{l} f_k, g_k = \mathbb{O}(x_k); \\ f_{\max}^* = \min_i \{f_i - \frac{1}{2L} \|g_i\|^2\}; \\ \alpha_k = \frac{f_k - f_{\max}^* + \frac{1}{2L} \|g_k\|^2}{\sqrt{g_k^\top M_k g_k}}; \\ x_{k+1} = x_k - \frac{1 + d\alpha_k}{d+1} \frac{M_k g_k}{g_k^\top M_k g_k}; \\ M_{k+1} = \frac{d^2}{d^2 - 1} (1 - \alpha_k^2) \left(M_k - \frac{2(1 + d\alpha_k)}{(1 + \alpha_k)(d+1)} \frac{g_k^\top M_k M_k g_k}{g_k^\top M_k g_k} \right); \end{array} \right.$$

The volume of the ellipsoids is equal to $\beta \sqrt{\det M}$, where β is some constant depending only on the dimension. The ratio quantifying the volume reduction is then $\sqrt{\frac{\det M_{k+1}}{\det M_k}}$. Using the rank-1 structure of the update on M_k , the following can be derived :

$$\begin{aligned} \det(M_{k+1}) &= \left(\frac{d^2(1 - \alpha_k^2)}{d^2 - 1} \right)^d \det \left(M_k - \frac{2(1 + d\alpha_k)}{(1 + \alpha_k)(d+1)} \frac{g_k^\top M_k M_k g_k}{g_k^\top M_k g_k} \right) \\ &= \left(\frac{d^2(1 - \alpha_k^2)}{d^2 - 1} \right)^d \left(1 - \frac{2(1 + d\alpha_k)}{(1 + \alpha_k)(d+1)} \right) \det(M_k) \\ &= \left(\frac{d^2(1 - \alpha_k^2)}{d^2 - 1} \right)^d \frac{(1 - \alpha_k)(d-1)}{(1 + \alpha_k)(d+1)} \det(M_k) \\ &= \left[(1 - \alpha_k^2)^d \frac{1 - \alpha_k}{1 + \alpha_k} \right] \left(\frac{d^2}{d^2 - 1} \right)^d \frac{d-1}{d+1} \det(M_k) \end{aligned}$$

This can be used to explicit the volume ratio between the ellipsoid that would have been obtained after a central cut and our deep cut, by comparing the volume reduction with $\alpha = 0$ and $\alpha \neq 0$:

$$\rho(\alpha) = \sqrt{1 - \alpha^2}^d \sqrt{\frac{1 - \alpha}{1 + \alpha}} \leq \sqrt{1 - \alpha^2}^d \exp(-\alpha)$$

This $\exp(-\alpha)$ bound is derived using properties of logarithms. The closer to 0 this ratio is, the better the deep cut method performs. Assuming the initial ellipsoid is given by $M_0 = R^2 I_d$ then one can derive a lower bound on the value of α :

$$\alpha_k = \frac{f_k - f_{\max}^* + \frac{1}{2L} \|g_k\|^2}{\sqrt{g_k^\top M_k g_k}} \geq \frac{1}{L} \frac{\|g_k\|^2}{\sqrt{g_k^\top M_k g_k}} \geq \frac{\|g_k\|}{L \sqrt{\lambda_{\max}(M_k)}} \geq \frac{\|g_k\|}{LR}$$

The largest eigenvalue of M_k is the length of the longest ellipsoid axis squared. While the volume of the ellipsoid decreases at each iteration, this reduction is not uniform across all directions. In particular, there is no guarantee that $\lambda_{\max}(M_k)$ decreases monotonically, it is only uniformly bounded above by its initial value (which is R^2). The lower bound is attained only when the cut is aligned with the longest axis of the ellipsoid, i.e., when g_k is an eigenvector associated with $\lambda_{\max}(M_k)$. This corresponds to the least favorable geometry for the current iteration, but it is also precisely the situation in which the ellipsoid update most strongly contracts that axis. As a result, such worst-case alignments cannot persist, and α_k controlled by

the joint evolution of the gradient norm and the ellipsoid geometry. If one can establish a lower bound on $\alpha_k \geq \delta$ then a better convergence rate can be derived by using the tools established in [subsection 2.9.3](#) :

$$\begin{aligned} \min_{j \leq k} f(x_j) - f^* &\leq MR e^{-\frac{k\delta}{d}} (1 - \delta^2)^{\frac{k}{2}} \cdot \left(1 - \frac{1}{(d+1)^2}\right)^{\frac{k}{2}} \cdot \left[\frac{\text{Vol}_d B_2(x_0, R)}{\text{Vol}_d Q}\right]^{\frac{1}{d}} \\ &\leq MR \cdot \exp\left(-\left[\frac{k}{2(d+1)^2} + \frac{k\delta}{d} + \frac{k\delta^2}{2}\right]\right) \cdot \left[\frac{\text{Vol}_d B_2(x_0, R)}{\text{Vol}_d Q}\right]^{\frac{1}{d}} \end{aligned}$$

The second expression is obtained using $1 - x \leq e^{-x}$. It is unclear whether it is possible to derive such a lower bound on α_k and what that lower bound would depend. In practice, α_k tends to be relatively small and offers only a small improvement compared to the standard ellipsoid method, as will be highlighted in [chapter 6](#).

4.2.3 Smooth ACCPM

The classic ACCPM can be improved using smoothness by leveraging the improved cuts. The algorithm is the same as [algorithm 5](#) except that it uses the smooth cutting planes, which require keeping track of $f_{\max}^*$. The new cutting planes's right hand side is given by :

$$\tilde{b}_k = \langle g_k, x_k \rangle - \left(f_k - f_{\max}^* + \frac{1}{2L}\|g_k\|^2\right) \leq \underbrace{\langle g_k, x_k \rangle - (f_k - \min_k f_k)}_{\text{Regular ACCPM cutting plane}} - \underbrace{\frac{1}{2L}\left(\|g_k\|^2 - \min_j \|g_j\|^2\right)}_{\text{Additional } < 0 \text{ term}}$$

These cutting planes are strictly better than regular ACCPM, assuming the iterates are not the minimizer ($\|g_k\| \neq 0$), and for the same sampling point, will always give a higher volume reduction.

Algorithm 16: Smooth ACCPM (S-ACCPM)

Input: $P_0 = \{x \mid j = 1, \dots, N : c_j^\top x \leq d_j\}$
Initialize $\mathcal{B}_0 = \emptyset$;
for $k = 0, 1, \dots$ **do**
 $\phi_k(x) = \sum_{j=1}^N \log(d_j - c_j^\top x) + \sum_{j \in \mathcal{B}_k} \log(b_j - a_j^\top x)$;
 $x_k = \arg \min_x -\phi_k(x)$;
 $f_k, g_k = \mathbb{O}(x_k)$;
 $f_{\max}^* = \min_i \{f_i - \frac{1}{2L}\|g_i\|^2\}$;
 $a_k = g_k, \quad b_k = \langle g_k, x_k \rangle - (f_k - f_{\max}^* + \frac{1}{2L}\|g_k\|^2)$;
 $\mathcal{B}_{k+1} = \mathcal{B}_k \cup \{k\}$;
 Remove constraints with high η_j from $\mathcal{B}_{k+1}$;

Similarly to the original ACCPM, if $f_{\max}^*$ changes, all the older cutting planes should be updated to account for the new, lower, value. Notably, they always remove the sample point itself from the localization set. These better bounds should, translate to better practical performance. While ACCPM is a very powerful method, it doesn't fully make use of the interpolation conditions. It just leverages the improved cutting planes.

4.3 New methods

The methods presented so far : the Smooth Level Method, the Smooth Ellipsoid Method, and Smooth ACCPM, all follow the same design philosophy: take a classical convex optimization algorithm and strengthen

its cutting planes by incorporating the smoothness interpolation conditions. While this leads to measurable improvements over their non-smooth counterparts, these methods do not fundamentally exploit the structure that smoothness provides. In particular, they make no use of the upper-bounding function or the lower-bounding function as global models of the objective. The methods introduced in this section take a different approach, building iterates directly from the interpolating bounds rather than from cutting planes alone.

4.3.1 Lower Level Method

The level method projects on the cutting planes defining the possible minimum values. In the convex case, those values are also the interpolating lower bound but here, with smoothness, this is not the case anymore. A quite natural idea would be to project on a level defined by the interpolating lower-bounding function $F_l(x)$, in a similar manner to what the regular level method does. Using the same $l_k(\alpha)$ as the smooth level method, the level set is as follows :

$$\mathcal{L}_k(\alpha) = \{x \quad \text{s.t.} \quad F_l(x) \leq l_k(\alpha)\}$$

While projecting on this set seems complicated, it is possible to derive an easy formulation of this using the linear constraint formulation for the lower bound [subsection 2.6.2](#).

$$\begin{aligned} F_l(x) \leq l_k(\alpha) &\iff \left\{ \min_{t,y} t + \frac{L}{2} \|x - y\|^2 \quad \text{s.t.} \quad \phi_k(y) \leq t \right\} \leq l_k(\alpha) \\ &\iff \exists(t, y) \quad \text{s.t.} \quad \begin{cases} t + \frac{L}{2} \|x - y\|^2 \leq l_k(\alpha) \\ \phi_k(y) \leq t \end{cases} \\ &\iff \exists y \quad \text{s.t.} \quad \phi_k(y) + \frac{L}{2} \|x - y\|^2 \leq l_k(\alpha) \end{aligned}$$

This set can be described using a either one conic constraint and k linear constraints. The projection is now tractable and can be used to define a new type of level set method described below.

Algorithm 17: Lower Level Method (LLM)

Input: $x_0, Q, L, \alpha, \mathbb{O}(x)$
for $k = 0, 1, \dots$ **do**
 $f_k, g_k = \mathbb{O}(x_k);$
 $\phi_k(y) = \max_{0 \leq i \leq k} \{f_i + \langle g_i, y - x_i \rangle + \frac{1}{2L} \|g_i\|^2\};$
 $\hat{f}_k^* = \min_{x \in Q} \phi_k(x);$
 $f_k^* = \min_{0 \leq i \leq k} \{f_i - \frac{1}{2L} \|g_i\|^2\};$
 $l_k = (1 - \alpha) \hat{f}_k^* + \alpha f_k^*;$
 $x_{k+1} = \arg \min_{y \in \mathbb{R}^d, x \in Q} \left\{ \frac{1}{2} \|x - x_k\|^2 \quad \text{s.t.} \quad \phi_k(y) + \frac{L}{2} \|x - y\|^2 \leq l_k \right\};$

The solution provided by the smooth level method is a feasible solution of the problem solved by (LLM). Indeed, taking $y = x$ reduces the problem to the one of the Smooth level Method. Despite this appealing formulation, in practice the performance is very close to that of the regular level method and smooth level method as will be demonstrated in [chapter 6](#).

4.3.2 Bound Mixing Method

The lower bound minimization is usually unbounded, as discussed earlier, but it can be stabilized in many ways. Previous work, such as IGMM, use a proximal term in order to guarantee boundedness. This method

can be seen as a pessimistic-optimistic approach, it mixes the upper and lower bounds to ensure the next iterate is well defined. The next iterate is found by minimizing a mixed function F_ρ defined as follows :

$$F_\rho(x) = \rho F_l(x) + (1 - \rho) F_h(x)$$

To derive the expression of $\min_x F_\rho(x)$, let's first derive a joint problem by separating the variables in the dual formulation of F_l and F_h and combine them in a joint minimization as follows.

$$\min_x F_\rho(x) = \min_x \max_{\lambda \in \Delta_m} \min_{\mu \in \Delta_m} \left\{ \sum_i \rho \lambda_i \psi_i(x) - \frac{\rho}{2L} \left\| \sum_i \lambda_i g_i \right\|^2 + (1 - \rho) \left[\sum_i \mu_i s_i + \frac{L}{2} \left\| x - \sum_i \mu_i u_i \right\|^2 \right] \right\}$$

Here strong duality applies and the maximization over λ can be swapped with the minimization over x . The first order optimality condition as writes as follows :

$$\begin{aligned} \rho \sum_i \lambda_i^* g_i + (1 - \rho) L \left[x_\rho^* - \sum_i \mu_i^* u_i \right] &= 0 \\ \iff x_\rho^* &= \sum_i \mu_i^* u_i - \frac{\rho}{1 - \rho} \frac{1}{L} \sum_i \lambda_i^* g_i \end{aligned}$$

Plugging this minimizer inside of the original expression allows for many simplification and removes x from the joint minimization.

$$\begin{aligned} \min_x F_\rho(x) &= \max_{\lambda \in \Delta_m} \min_{\mu \in \Delta_m} \left\{ \sum_i \rho \lambda_i b_i - \frac{\rho}{2L} \left\| \sum_i \lambda_i g_i \right\|^2 + \sum_i \mu_i (1 - \rho) s_i + (1 - \rho) \frac{L}{2} \left\| \frac{\rho}{1 - \rho} \frac{1}{L} \sum_i \lambda_i g_i \right\|^2 \right. \\ &\quad \left. + \rho \sum_i \sum_j \lambda_i \mu_j \langle u_j, g_i \rangle - \frac{\rho^2}{1 - \rho} \frac{1}{L} \sum_i \sum_j \lambda_i \lambda_j \langle g_i, g_j \rangle \right\} \\ &= \max_{\lambda \in \Delta_m} \min_{\mu \in \Delta_m} \left\{ \sum_i \rho \lambda_i b_i - \frac{1}{2L} \frac{\rho}{1 - \rho} \left\| \sum_i \lambda_i g_i \right\|^2 + \sum_i \mu_i \left[(1 - \rho) s_i + \rho \left\langle u_i, \sum_j \lambda_j g_j \right\rangle \right] \right\} \end{aligned}$$

The minimization over μ can be handled via the epigraph technique, reducing the problem to a quadratic objective with $2m + 1$ linear constraints. Once λ^* and μ^* are obtained by solving the above problem, the minimizer is recovered via strong duality. When $\rho \rightarrow 0$, one recovers the minimization of the upper bound both in objective and in minimizer value :

$$\min_x F_0(x) = \max_{\lambda \in \Delta_m} \sum_i \lambda_i s_i = \min_i s_i$$

On the other hand, when $\rho \rightarrow 1$, the feasibility condition of the dual appears naturally, indeed if $\sum_i \lambda_i g_i \neq 0$ then the problem becomes unbounded and the objective becomes $-\infty$. If the problem is feasible then the objective function is finite as it is equal to $F_l(x_l^*)$.

$$\min_x F_1(x) = \max_{\substack{\lambda \in \Delta_m \\ \sum_i \lambda_i g_i = 0}} \left\{ \sum_i \lambda_i b_i + \min_i \left[\left\langle u_i, \sum_j \lambda_j g_j \right\rangle \right] \right\} = \max_{\substack{\lambda \in \Delta_m \\ \sum_i \lambda_i g_i = 0}} \lambda^\top b$$

The expression of the minimizer however, is not well defined and x^* is the dual variable linked to the equality constraint on the gradients, as mentioned earlier. This helps identify a key constraint in the Bound Mixing Method. It will tend to be numerically unstable as one pushes ρ close to 1. Sadly pushing

ρ to 1 tends to yield better performances.

While this approach seems sound on paper, in practice it suffers from numerical issues, high computational cost and does not perform well. The main issue is that, as $\rho \rightarrow 1$, the optimizer heavily suffers from numerical instabilities. On the other hand, the computational cost is quite high, even with warm starting requiring a full QP with many constraints, and the performance simply does not make up for it as it generally is worse than regular first order methods such as OGM.

4.3.3 Information Maximization

If the gap between the upper and lower bounding functions on the minimum localization set could be collapsed, then the values that a point in that set can take would converge to the value of the minimum, and convergence of the method would follow. In that spirit, consider the following quantity, which will be called the information:

$$I(x) = F_h(x) - F_l(x)$$

By construction, $I(x) \geq 0$, and at interpolation points x_k it collapses to

$$I(x_k) = F_h(x_k) - F_l(x_k) = 0.$$

If the value of $I(x)$ on $\mathcal{X}_m$ is bounded by some ε , then sampling at the point x_l^* minimizing F_l yields an ε -approximate solution:

$$f(x_l^*) - f^* \leq f(x_l^*) - F_l(x^*) \leq F_h(x_l^*) - F_l(x_l^*) \leq \varepsilon$$

Since sampling a point collapses its information value to 0, a natural idea is to sample, at each iteration, the point with the maximum information value and add it to the bundle. The hope is that this procedure will collapse information efficiently. The point maximizing the information is easiest to find using the primal formulation of the bounds. First, note that the information can be computed by solving the following problem :

$$\begin{aligned} F_h(x) - F_l(x) = \max_{f_l, f_h, g_l, g_h} \quad & f_h - f_l \\ \text{s.t.} \quad & f_l \geq f_k + \langle g_k, x - x_k \rangle + \frac{1}{2L} \|g_k - g_l\|^2 \quad \forall k \\ & f_h \geq f_k + \langle g_k, x - x_k \rangle + \frac{1}{2L} \|g_k - g_h\|^2 \quad \forall k \\ & f_k \geq f_l + \langle g_l, x_k - x \rangle + \frac{1}{2L} \|g_k - g_l\|^2 \quad \forall k \\ & f_k \geq f_h + \langle g_h, x_k - x \rangle + \frac{1}{2L} \|g_k - g_h\|^2 \quad \forall k \end{aligned}$$

If one were to maximize this value over the entire space, it would generally be unbounded, since the lower bound function grows at most linearly far from sample points, while the upper bounds grow quadratically. Furthermore, information values are only relevant inside the minimum localization set. It therefore seems

reasonable to maximize information under the constraint $x \in \mathcal{X}_m$:

$$\begin{aligned}
\max_x [F_h(x) - F_l(x)] &= \max_{f_l, f_h, g_l, g_h, x} f_h - f_l \\
\text{s.t. } \forall k = 1, \dots, m : \quad &f_l \geq f_k + \langle g_k, x - x_k \rangle + \frac{1}{2L} \|g_k - g_l\|^2 \\
&f_h \geq f_k + \langle g_k, x - x_k \rangle + \frac{1}{2L} \|g_k - g_h\|^2 \\
&f_k \geq f_l + \langle g_l, x_k - x \rangle + \frac{1}{2L} \|g_k - g_l\|^2 \\
&f_k \geq f_h + \langle g_h, x_k - x \rangle + \frac{1}{2L} \|g_k - g_h\|^2 \\
&f_k + \langle g_k, x - x_k \rangle + \frac{1}{2L} \|g_k\|^2 \leq f_{\max}^*
\end{aligned}$$

This non-convex program must be solved at each iteration, which requires a large computational effort. In practice, the main issue is that the point that tends to maximize information is the point furthest away from every other sample. In more than one dimension, this results in exploring the vertices of $\mathcal{X}_m$, which yields **very** slow convergence, if convergence occurs at all. For a method requiring a non-convex program, this is not efficient.

4.3.4 Lower Bound Method (LBM)

A key limitation of the information maximization approach is that it seeks to reduce the gap between upper and lower bounds uniformly over all admissible functions, rather than focusing specifically on functions that actually attain a minimum in $\mathcal{X}_m$. Minimum localization, by contrast, depends only on the bounds on the attainable minimum values. In the preceding sections, such bounds were obtained by imposing $g = 0$ in the interpolation conditions, and yielded $f_{\min}^*(x)$ and $f_{\max}^*(x)$. These quantities characterize the range of possible minimum values directly, and are therefore the relevant objects for minimum localization. In contrast with information maximization, where the pointwise gap $F_h(x) - F_l(x)$ is reduced, the present objective is to reduce the gap between $f_{\min}^*$ and $f_{\max}^*$. When a point is sampled, it is excluded from the minimum localization set (if it is not the minimizer). Since $f_{\max}^*$ does not depend on x , we have

$$\max_{x \in \mathbb{R}^d} [f_{\max}^* - f_{\min}^*(x)] = f_{\max}^* - \min_{x \in \mathbb{R}^d} f_{\min}^*(x).$$

By construction, the smallest attainable minimum value is achieved at the point where the lower envelope $F_l(x)$ reaches its minimum. As such, minimizing $f_{\min}^*(x)$ is equivalent to minimizing $F_l(x)$. Due to this link with the lower envelope, this method is called the Lower Bound Method (LBM). The resulting strategy therefore coincides with the optimistic approach and yield the following problem.

$$x_{k+1} = \arg \min_{x, t} t \quad \text{s.t.} \quad b_i + \langle g_i, x \rangle \leq t \quad i = 1, \dots, m.$$

Since the selected point is the minimizer of a smooth convex function compatible with the data, it must belong to $\mathcal{X}_m$. One may therefore hope it would lie in a reasonably central region of the localization set, since a cut generated at central point should remove a larger portion of the current feasible region and thus produce strong reduction of volume of $\mathcal{X}_m$. The point chosen by LBM will tend to be somewhat central, as it will be far away from the boundaries of the localization set. This intuition can be made more precise by reformulating the problem. If one introduces the tightening parameter $\tau = f_{\max}^* - t$, it can be rewritten as

$$x_{k+1} = \arg \max_{x, \tau} \tau \quad \text{s.t.} \quad b_i - f_{\max}^* + \langle g_i, x \rangle \leq -\tau, \quad i = 1, \dots, m.$$

Remember that the minimum localization set is given $\mathcal{X}_m = \{x \mid g_i^\top x \leq c_i, i = 1, \dots, m\}$ where $c_i = f_{\max}^* - b_i$. The problem can thus be rewritten as follows :

$$\begin{aligned} x_{k+1} &= \arg \max_{x, \tau} \tau \quad \text{s.t.} \quad \langle g_i, x \rangle \leq c_i - \tau, \quad i = 1, \dots, m. \\ &= \arg \max_x \min_i \{c_i - \langle g_i, x \rangle\} \end{aligned}$$

This formulation shows that the point selected by minimizing the lower bound is the point around which the localization set can be uniformly tightened the most. In other words, it is the point maximizing the minimum slack of $\mathcal{X}_m$, which should yield a somewhat central point. Due to this centrality property, the selected point is in a descent direction for all the previous sample points (assuming none of the previous iterates are the minimizer).

$$\forall i = 1, \dots, k : \quad \langle g_i, x_{k+1} - x_i \rangle \leq - \left(f_i - t + \frac{1}{2L} \|g_i\|^2 \right) < 0$$

As, t is the value of the lowest minimizer compatible with the data, it is always less than f_i and thus the above quantity is always negative. In practice, suitable modifications are required to make this formulation implementable and competitive, which will be described in the next chapter.

Chapter 5

Refinements and Theoretical Properties of LBM

Methods presented earlier rely on the interpolation conditions in order to derive the next iterate. These conditions are defined in terms of the ℓ_2 -norm, and are dependent only on inner products and norm-based quantities. This suggests that the methods presented earlier might exhibit invariance properties under transformations of space that preserve the ℓ_2 -norm. Those methods (notably the cutting plane methods like the Ellipsoid method, ACCPM, Vaidya's Method and LBM) are also known to deteriorate as the ambient dimension increases, as a constant volume decrease ρ of $\mathcal{X}_m$ only reduces the functional residual by a factor of $\rho^{1/d}$. This raises the question of whether the method can adapt to lower-dimensional structure when the objective function effectively depends on a reduced number of directions (low-rank structure). The main focus will be put on LBM and ACCPM as those are promising methods. Additionally the performance of LBM can be improved using a warm-start strategy and special pruning techniques. This chapter will cover the invariance properties of the methods using the minimum localization set (which is mainly focused towards LBM but valid for many different methods), the boundedness of the inner problems, the adaptation to lower-dimensional structure and the warm-start and pruning techniques for LBM.

5.1 Invariance

An optimization method is said to be invariant under a transformation if the sequence of iterates produced in the transformed space corresponds to the image of the iterates generated in the original space. As all the optimizers presented rely on the interpolation conditions in order to produce their next iterate, studying the invariance properties of these conditions can provide insights into the invariance properties of the methods themselves. In this particular setting, let's consider how the minimum localization set is affected by those transformations.

5.1.1 Invariance to transformations of functional value

Consider an affine transformation of the function value as follows :

$$h(x) = af(x) + b, \quad \nabla h(x) = a\nabla f(x), \quad L_h = aL_f.$$

Where $a, b \in \mathbb{R}$ with $a > 0$. The corresponding samples become $(x_i, g_i, a \cdot f_i + b)$. The cutting plane inequalities then read :

$$\begin{aligned} \forall i : a f_i + b + a \langle g_i, x - x_i \rangle + \frac{1}{2aL} \|a g_i\|^2 &\leq \min_j \left\{ a f_j + b - \frac{1}{2aL} \|a g_j\|^2 \right\} \\ \iff \forall i : a \left(f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \right) + b &\leq a \min_j \left\{ f_j - \frac{1}{2L} \|g_j\|^2 \right\} + b \end{aligned}$$

This shows that an affine transformation of the function value does not affect the geometry of the localization set, as the inequalities defining $\mathcal{X}_m$ are preserved. This means that the methods relying on the geometry of the localization set are invariant under affine transformations of the function value, which is a desirable property as it implies that their behavior is unaffected by changes in scale or shifts of the objective function. However, scaling f changes L and thus the link between the volume of $\mathcal{X}_m$ and the functional residual.

5.1.2 Invariance to transformations of space

Let's consider transformations of space of type $x \rightarrow \alpha Q x + \beta$, where Q is orthogonal, i.e. $Q^\top Q = I$. Such transformations include uniform scalings, orthogonal transformations, changes of coordinates and translation. Consider the following transformed function :

$$h(x) = f(\alpha Q x + \beta), \quad \nabla h(x) = \alpha Q^\top \nabla f(\alpha Q x + \beta), \quad L_h = \alpha^2 L_f.$$

Define the point $z_i = Q^\top(x_i - \beta)/\alpha$. The transformed samples are then (z_i, v_i, h_i) where $h_i = f_i$, $v_i = \alpha Q^\top g_i$ and $z_i = Q^\top(x_i - \beta)/\alpha$. The cutting plane inequalities for the transformed function read :

$$\begin{aligned} \forall i : h_i + \langle v_i, z - z_i \rangle + \frac{1}{2L_h} \|v_i\|^2 &\leq \min_j \left\{ h_j - \frac{1}{2L_h} \|v_j\|^2 \right\} \\ \iff \forall i : f_i + \langle \alpha Q^\top g_i, z - z_i \rangle + \frac{1}{2\alpha^2 L} \|\alpha Q^\top g_i\|^2 &\leq \min_j \left\{ f_j - \frac{1}{2\alpha^2 L} \|Q^\top g_j\|^2 \right\} \\ \iff \forall i : f_i + \left\langle g_i, \frac{Q}{\alpha}(z - z_i) \right\rangle + \frac{1}{2L} \|g_i\|^2 &\leq \min_j \left\{ f_j - \frac{1}{2L} \|g_j\|^2 \right\} \end{aligned}$$

This shows that transformations of space of the form $x \rightarrow \alpha Q x + \beta$ preserve the geometry of the localization set, as the inequalities defining $\mathcal{X}_m$ are preserved under such transformations. The set is still getting stretched by a factor of α in all directions, but this does not affect the relative geometry of the set. That shows that methods relying on the geometry of the localization set are invariant under these transformations of space and will generate iterates that are the image of the iterates generated in original space. Those invariance properties will be leveraged in the generation of adequate test functions in the [chapter 6](#), as they allow testing for a large variety of functions using a single base function.

The only difference with general affine transformation is these transformations do not include anisotropic scaling, which is to be expected as those anisotropic transformations change the underlying spectral properties of the problem, which cannot be captured by a single number L . They also affect the performance of the methods by changing those spectral properties by creating flat or very steep directions.

5.1.3 Adaptation to lower dimensional structure

The proposed method relies on cutting-plane constructions, which are known to deteriorate as the ambient dimension d increases. It is therefore of interest to understand whether the method can exploit a lower-dimensional structure when the objective function effectively depends on a reduced number of directions.

Consider a transformation defined by a matrix $U \in \mathbb{R}^{r \times d}$, with $r \leq d$, and define

$$h(z) = f(Uz), \quad \nabla h(z) = U^\top \nabla f(Uz).$$

Define the transformed samples $z_i = Ux_i$, $h_i = f_i$ and $v_i = U^\top g_i$. Assuming the lipschitz constant of h to be L_h , the localization set is then transformed as follows :

$$\begin{aligned} \forall i : h_i + \langle v_i, z - z_i \rangle + \frac{1}{2L_h} \|v_i\|^2 &\leq \min_j \left\{ h_j - \frac{1}{2L_h} \|v_j\|^2 \right\} \\ \iff \forall i : f_i + \langle U^\top g_i, z - z_i \rangle + \frac{1}{2L_h} \|U^\top g_i\|^2 &\leq \min_j \left\{ f_j - \frac{1}{2L_h} \|U^\top g_j\|^2 \right\} \\ \iff \forall i : f_i + \langle g_i, U(z - z_i) \rangle + \frac{1}{2L_h} \|U^\top g_i\|^2 &\leq \min_j \left\{ f_j - \frac{1}{2L_h} \|U^\top g_j\|^2 \right\} \end{aligned}$$

This formulation shows that the constraints depend on z only through Uz . As a consequence, the localization set is not constrained in the directions of the kernel of U . Depending on how the method use the geometry of the localization set and the dimension d , they can potentially adapt to this lower-dimensional structure without prior knowledge. The ellipsoid method for instance, cannot adapt to this structure as it relies on the dimension explicitly. On the other hand, ACCPM, LBM and Vaidya's method can adapt to this structure, as the directions in the kernel of U do not affect the objective. Note that this can however lead to unboundedness issues, as the localization set is not bounded in the directions of the kernel of U .

5.2 Boundedness

This section will focus mainly on LBM. The localization set is usually unbounded, which leads to unboundedness of the optimization problems. In LBM, the primal is bounded as soon as $\exists \lambda \in \Delta_m : \sum_i \lambda_i g_i = 0$, which is a weaker condition than requiring $\mathcal{X}_m$ to be bounded. However, while the value of the primal is bounded, it can still be unstable in practice due to some directions not being constrained.

5.2.1 Euclidean Norm

A natural choice of to ensure boundedness is to use a ball of the same norm as the one the interpolation conditions are formulated in. Here, the interpolation conditions were considered using the ℓ_2 -norm, so it is only natural to consider a ℓ_2 -norm ball as it will preserve invariance under orthogonal transformations. Using the ball center z and radius R , the inner problem becomes :

$$\begin{aligned} \min_{x, t \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad & \psi_i(x) \leq t \quad \forall i \\ & \|x - z\|_2^2 \leq R^2 \end{aligned}$$

Denote the arg min in x of that problem as x^* , This condition adds interesting properties to the optimization problem, notably, it constrains directions not in the span of gradients in a quite natural manner. Defining λ_i as the dual variable linked to the inequalities on the ψ_i and μ as the variable linked to boundedness, the stationarity condition is as follows :

$$\sum_{i=1}^m \lambda_i g_i + 2\mu(x^* - z) = 0 \iff x^* = z - \frac{1}{2\mu} \sum_{i=1}^m \lambda_i g_i$$

Usually one sets $z = x_0$. If the ball constraint is active then μ can be non zero and x^* lies in $z + \text{span}\{g_0, \dots, g_k\}$. If the ball constraint is inactive, the solution is not constrained in the directions of

the kernel of the span of the gradients but the solution point still is bounded, solving any numerical issue that could arise. In practice, it can be interesting to force $x^* \in z + \text{span} \{g_0, \dots, g_k\}$ by altering slightly to have better numerical properties. This requires to keep track of a basis of $\text{span} \{g_0, \dots, g_k\}$ (which can be done using progressive QR factorization). Denoting Q the matrix whose columns are an orthonormal basis of $\text{span} \{g_0, \dots, g_k\}$, the inner problem can be reformulated using $x = z + Q\alpha$ as follows :

$$\min_{\alpha, t \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad \forall i : \psi_i(z + Q\alpha) \leq t \\ \|\alpha\|_2^2 \leq R^2$$

Indeed, Q is orthogonal so $\|Q\alpha\| = \|\alpha\|$. In practice, this formulation can help better exploit natural symmetries of the problem. Notably, $f(x) = \frac{1}{2}\|x\|^2$ is treated as a one-dimensional problem as the gradients are all colinear, and the method converges much faster, while the initial formulation might choose points that are badly conditioned in the directions of the kernel of the span of the gradients.

When the ball constraint is active the piecewise-affine model $\phi_m(x) = \max_i \psi_i(x)$ has a non-zero subgradient, as the selected point is not the minimizer of the model (unless $\mu = 0$ and the solution lies exactly at the boundary, which is a degenerate case). The gradient of each $\psi_i(x) = g_i$, yielding a clean formula for one of the subdifferentials of ϕ_m at x^* . Using Danskin's theorem :

$$\partial\phi_m(x^*) = \text{conv} \{g_i : \psi_i(x^*) = t\} \implies p_m = \sum_i \lambda_i g_i \in \partial\phi_m(x^*)$$

This result is valid whether the ball constraint is active or not, p_m is always a valid subgradient of $\phi_m(x^*)$. If the ball constraint is active, it is usually non zero, if the constraint is inactive, it is zero, certifying that x^* is a minimizer of the piecewise-affine model.

5.2.2 Infinity norm based boundedness

The previous formulation required to solve a quadratic program, which can be more computationally expensive than a linear program. Using an ∞ -norm ball instead of an ℓ_2 -norm ball, the inner problem becomes :

$$\min_{x, t \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad \forall i : \psi_i(x) \leq t \\ \|x - x_0\|_\infty \leq R$$

One might think that this formulation is more computationally efficient but since the quadratic program only uses a simple conic constraint the practical execution time is similar. A huge drawback of this formulation however is that the initial volume can be much larger depending on the dimension. Indeed, for a given radius R , the volume ratio between the ℓ_2 -norm ball and the ∞ -norm ball is :

$$\text{Vol}(B_2(R)) / \text{Vol}(B_\infty(R)) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2} + 1)2^d} \approx \left(\frac{\pi e}{2d}\right)^{\frac{d}{2}}$$

This means that the initial volume is much larger, which is quite detrimental to the convergence of the method as the functional residual depends on the volume of the localization set, creating a fixed cost. Moreover, the ∞ -norm ball does not preserve any invariance by orthogonal transformation the initial problem might have, which can lead to very bad performance in some cases.

5.3 Convergence Proof for LBM

The aim of this section is to provide an asymptotic convergence proof for LBM. The proof will be quite rough in terms of guarantees on convergence rate as the main goal is to show that the method eventually

converges. The proof technique comes from [9], and is very similar to the one used for Kelley's method. Indeed, LBM is essentially a smooth variant of Kelley's method. Now that boundedness problems have been addressed, the LBM algorithm is well defined, and can be written as follows :

Algorithm 18: Lower Bound Method (LBM)

Input: x_0, R, L

for $k = 0, 1, \dots$ **do**

$f_k = f(x_k);$
 $g_k = \nabla f(x_k);$
 $\phi_k(x) = \max_{0 \leq i \leq k} \{f_i + \langle g_i, x - x_i \rangle + \frac{1}{2L} \|g_i\|^2\};$
 $x_{k+1} = \arg \min_{\|x - x_0\| \leq R} \phi_k(x);$

Consider the iterates (x_k) generated by the LBM, we will show that it takes finitely many steps in order to reach a ε -approximate solution in function residual. First define the set of iterates with a functional residual larger than ε :

$$K_\varepsilon = \{k \quad \text{s.t.} \quad f_k - f^* > \varepsilon\}$$

Consider iterates $k_1, k_2 \in K_\varepsilon$ with $k_1 < k_2$ and evaluate the cut provided by k_1 at x_{k_2} and compare it to the lower model $\phi_{k_1}(x)$:

$$f_{k_1} + \langle g_{k_1}, x_{k_2} - x_{k_1} \rangle + \frac{1}{2L} \|g_{k_1}\|^2 \leq \phi_{k_1}(x_{k_2})$$

Since a cut is added at each iteration the model can only become tighter : for all $j \geq k_1$ we have $\phi_{k_1}(x) \leq \phi_j(x)$. The model ϕ_{k_2-1} provides a lower bound on f^* at x^* as :

$$\forall i = 0, \dots, k_2 - 1 : f^* \geq f_i + \langle g_i, x^* - x_i \rangle + \frac{1}{2L} \|g_i\|^2 \implies f^* \geq \phi_{k_2-1}(x^*)$$

Since x^* is a feasible point for the inner minimization (as $\|x^* - x_0\| \leq R$), the minimum of ϕ_{k_2-1} will provide a lower bound on f^* . Using the assumption that $k_2 \in K_\varepsilon$, this yields the following :

$$\begin{aligned} f_{k_1} + \langle g_{k_1}, x_{k_2} - x_{k_1} \rangle + \frac{1}{2L} \|g_{k_1}\|^2 &\leq \phi_{k_1}(x_{k_2}) \leq \phi_{k_2-1}(x_{k_2}) \leq \phi_{k_2-1}(x^*) \leq f^* < f_{k_2} - \varepsilon \\ \iff \varepsilon &< f_{k_2} - f_{k_1} - \langle g_{k_1}, x_{k_2} - x_{k_1} \rangle - \frac{1}{2L} \|g_{k_1}\|^2 \end{aligned}$$

Now using interpolation conditions, one can eliminate f_{k_1} and f_{k_2} from the expression as :

$$\begin{aligned} f_{k_1} &\geq f_{k_2} + \langle g_{k_2}, x_{k_1} - x_{k_2} \rangle + \frac{1}{2L} \|g_{k_1} - g_{k_2}\|^2 \\ \implies \varepsilon &< \langle g_{k_2} - g_{k_1}, x_{k_2} - x_{k_1} \rangle - \frac{1}{2L} (\|g_{k_1}\|^2 + \|g_{k_1} - g_{k_2}\|^2) \end{aligned}$$

Here, the gradient terms provide additional information about convergence but they are not required to show asymptotic convergence. Let's derive a loose bound on ε using smoothness :

$$\varepsilon < \|g_{k_2} - g_{k_1}\| \|x_{k_2} - x_{k_1}\| \leq L \|x_{k_2} - x_{k_1}\|^2 \iff \|x_{k_2} - x_{k_1}\| > 2\delta = \sqrt{\frac{\varepsilon}{L}}$$

Remember that $\forall k : x_k \in B(x_0, R)$ by definition. This provides interesting insight : the family of points $(x_k)_{k \in K_\varepsilon}$ is a set of points in a ball of radius R , which are at least a distance 2δ apart from each-other. Each iterate defines a ball of radius δ which are disjoint from one another and every ball is contained in

$B(x_0, R + \delta)$ (note that we can even consider all the points to be in the half ball since the first cut will always remove half of the ball from the localization set). Using volume comparison, one can infer that :

$$|K_\varepsilon| \text{Vol}(B(0, \delta)) \leq \text{Vol}(0, R + \delta) \implies |K_\varepsilon| \leq \left(\frac{R + \delta}{\delta} \right)^d \leq \left(2R \sqrt{\frac{L}{\varepsilon}} + 1 \right)^d$$

While this bound is very loose as iterates cannot be in the parts of the ball that have been removed by the different cutting planes, it provides a proof that K_ε is finite for every $\varepsilon > 0$ and thus proves asymptotic convergence in function value of the algorithm. It also shows that after k iterations, the guarantee is at least :

$$\min_{i \leq k} f_i - f^* \leq \frac{4R^2 L}{(k^{\frac{1}{d}} - 1)^2}$$

While this upper bound is quite disappointing, the following sections will cover methods used to transform LBM into a practically usable method with performances on test problems that usually outperforms classical local methods.

5.4 Warm Start

Since R is quite large, the initial localization set is large, which leads to a large initial function residual and requires LBM to use a large number of iterations to be on the same level as classical local methods, as it spends a large amount of time building a good bundle. In order to mitigate this issue, a warm start can be used to progressively build an initial bundle and avoid the largest part of the fixed cost. This is done by choosing a small starting radius R_0 around a center z_k and progressively increasing it.

5.4.1 Infeasibility warm-start

One way to control the radius is to increase it only when the inner problem certifies that no minimizer can lie in the current localization set. This is better done for a fixed center z . For a given radius R , the inner problem provides a necessary and sufficient condition—based solely on the available samples—for the existence of a smooth convex function whose minimizer lies inside the ball $B_2(z, R)$. More precisely, consider the min-max formulation, the condition

$$\min_{\|x-z\|_2 \leq R} \max_i \psi_i(x) \leq \min_j s_j$$

is equivalent to the existence of a function f that is L -smooth and convex, interpolates the observed data (x_i, f_i, g_i) , and admits at least one minimizer $x^* \in B_2(z, R)$. Importantly, this condition does not assert that the true underlying function has its minimizer inside the ball. Rather, it guarantees that there exists at least one function consistent with all past observations for which a minimizer lies in the localization set. In that sense, it is the strongest possible statement that can be made given the available information: if the condition fails, then no L -smooth convex function interpolating the data can have a minimizer within $B_2(z, R)$, and the radius must therefore be increased. This condition is thus stronger than the minimal radius one derived earlier. While this approach is more principled, there is no clear bound on the number of iterations needed in to detect if the current localization ball contains the minimizer. In practice, the number of iterations required can be quite large, leading to poor performance.

5.4.2 Trust Region Warm-Start

Another approach is to choose a small R_0 around x_0 and progressively increasing it and, instead of keeping the center constant, shifting the ball by setting $z_k = x_k$, similarly to a trust region method. Let us define a growth rate γ_k and set $R_k = \gamma_k R_{k-1}$. The inner problem becomes :

$$\min_{x, t \in \mathbb{R}^{d+1}} t \quad \text{s.t.} \quad \forall i : \psi_i(x) \leq t \\ \|x - x_k\|_2^2 \leq R_k^2$$

Making the right choice of R_0 and γ_k can help mitigate the fixed cost of the method. The interpolation conditions provide some heuristics on how to tune those hyperparameters. Indeed, one can lower bound the distance between an iterate (x_k) and the minimizer x^* as follows :

$$f_{\max}^* \geq f_k + \langle g_k, x^* - x_k \rangle + \frac{1}{2L} \|g_k\|^2 \implies \|x_k - x^*\|_2 \geq R_{\min}^k = \left[\frac{f_k - f_{\max}^*}{\|g_k\|} + \frac{1}{2L} \|g_k\| \right] \geq \frac{1}{L} \|g_k\|$$

This condition on the radius is less strict than the one derived in the previous section, but it does not requires to solve the LP to have an estimation of the minimum radius. A natural choice of R_0 is thus $R_0 = \frac{1}{L} \|g_0\|$, which ensures that the first iterate is not too far from x_0 and can potentially be a minimizer. Choosing the right growth rate γ_k is difficult and no theoretically based heuristic is known. In practice, a constant growth rate $\gamma_k = \gamma$ can be used by tuning γ empirically.

5.4.3 Hyperparameter Free Warm-Start

Another interesting effect of using $z = x_k$ is that, if the ball constraint is active then the step can be seen as a normalized proximal subgradient step of step size R_k as we have :

$$x^* = x_k - R_k \frac{\sum_i \lambda_i g_i}{\|\sum_i \lambda_i g_i\|} = x_k - R_k \frac{p_k}{\|p_k\|} \quad \text{with} \quad p_k \in \partial \phi_m(x^*)$$

This insight can help us design a hyperparameter free warm start strategy by setting R_{k+1} to be non decreasing, at least $R_{\min}^k$ and increasing when the model is steep at x^* as follows :

$$R_{k+1} = \max \left(R_k, R_{\min}^k + \frac{\|p_k\|}{L} \right)$$

The idea behind this choice is that if the model is steep at x^* (which is signified by a large $\|p_k\|$), then the radius should be increased to allow the iterates to move in a more significant manner. The norm of p_k is normalized by L which is the length scale of our problem and shifts the measure from gradient space to x -space. This heuristic is quite effective in practice and does not require any hyperparameter tuning, which is a big advantage. Better results can be achieved by carefully tuning the growth rate, but this is a good default choice and will be used in the practical experiments.

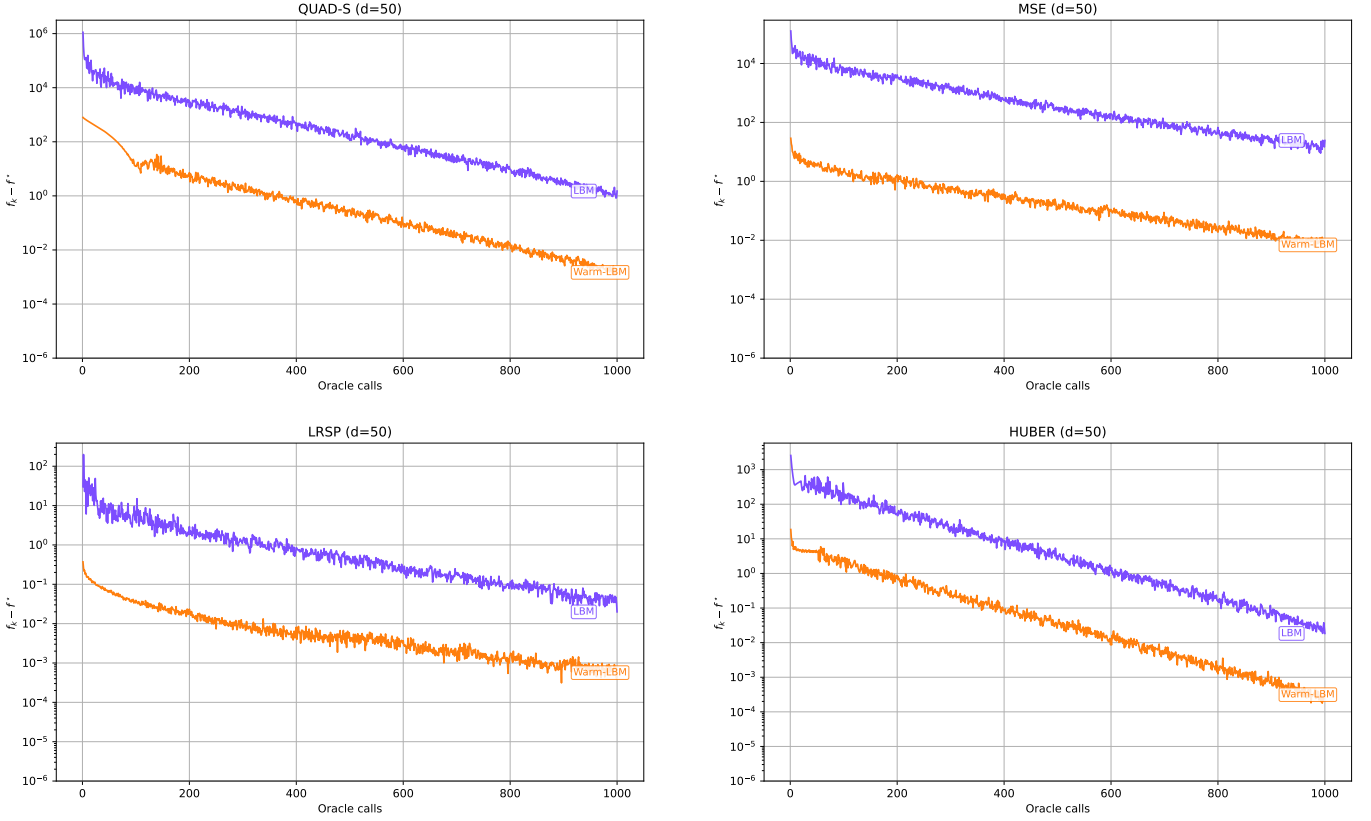


Figure 5.1: Hyperparameter free warm start on multiple problems. (A description of the test problems can be found in [chapter 6](#))

Hyperparameter free showcases (Figure 5.1) : shows the hyperparameter free warm start on multiple problems that will be described in the benchmark chapter [chapter 6](#). This shows that this hyperparameter free warm start effectively negates any fixed cost the LBM might incur while keeping the asymptotic convergence properties of the method.

5.4.4 Link with level method

The warm start strategies described above can be linked to the smooth level method in a quite direct manner. Indeed, a given radius R for the problem defines a level l of the smooth level method as follows :

$$l(R) = \min_{\|x - x_k\|_2^2 \leq R^2} \max_i \psi_i(x)$$

If the radius does not constraint the minimizer of the LP then $l(R) = f_{\min}^*$, if the ball constraint is active then the corresponding level can be found by either solving the above problem or it's dual. Using μ as the dual variable linked to the ball constraint, the KKT conditons read :

$$x^* = z - \frac{1}{2\mu} \sum_i \lambda_i g_i, \quad \mu = \frac{1}{2R} \left\| \sum_i \lambda_i g_i \right\|$$

And the dual problem determining the corresponding level is derived by plugging the stationarity condition in the Lagrangian as follows :

$$l(R) = \max_{\lambda \in \Delta_m} \sum_i \lambda_i \psi_i(x^*) = \max_{\lambda \in \Delta_m} \left\{ \sum_i \lambda_i \psi_i(x_k) - R \left\| \sum_i \lambda_i g_i \right\| \right\}$$

One can derive an easy to compute lower bound on the level by simply taking $\lambda = e_k$:

$$l(R) \geq f_k + \frac{1}{2L} \|g_k\|^2 - R \|g_k\| = f_{\max}^* - (R - R_{\min}^k) \|g_k\|$$

This link shows how the warm start strategy can stabilize early iterations of LBM by constraining the iterates in a similar manner to the level method. A big difference however is that the level method never allows the iterates to achieve the lower bound $f_{\min}^*$, while the warm start strategy allows the iterates to achieve this bound as soon as the radius is large enough. When that critical radius is reached, linear convergence is usually observed, while the level method only achieves sublinear convergence. However, there can be a sudden jump in functional residual if the radius increase is too abrupt and the warm start has not negated the entirety of the fixed cost yet. A careful tuning of the growth rate can allow LBM to reach the best of both worlds, with a fast initial convergence and a linear convergence once the radius is large enough. The hyperparameter free warm start usually achieves that goal.

5.5 Averaging Techniques

LBM usually create iterates which move significantly between iterations, which can lead to slow convergence and oscillations. Additionally, when testing on simple 2D examples, it was observed that LBM oscillates around the minimum. To alleviate this problem a common and cheap strategy usually used in convex optimization is averaging. The general idea is to define y_k , the averaged iterate as :

$$y_k = \frac{\sum_i w_i x_i}{\sum_i w_i}$$

Where the w_i are weights associated to each sample point. Multiple choices of weights will be explored in this section, some leading to interesting convergence properties. A few possible choice of weights are the following :

- **Simple averaging** (avg) : The most natural choice of weights is to simply set $w_i = 1$ for all i . This method is very simple and can help mitigate the oscillations of the iterates, but it does not leverage any information on the samples and uses even the oldest sample as much as the most recent one, which can lead to bad performance.
- **Last $d + 1$ iterates** (d+1) : Since only $d + 1$ iterates are used at each iteration, one could consider that only $d + 1$ iterations can be relevant to the LBM inner problem. This motivates the choice of weights $w_i = 1$ if $i \geq k - d$ and $w_i = 0$ otherwise. This method is simple and leverages the fact that only $d + 1$ iterates are relevant to the optimization process.
- **Active averaging** (active) : An idea very similar to the $d + 1$ averaging would be to only use the samples linked to the active constraints only. In some cases, that might be less than $d + 1$ samples. The weights are thus defined as $w_i = 1$ if $\lambda_i > 0$ and $w_i = 0$ otherwise.
- **Dual Averaging** (dual) : This method will be studied in the next section, as it presents interesting properties. It uses the dual variables λ_i as weights for the averaging. Those dual variables can be seen as a measure of the relevance of the corresponding sample to the optimization process, and thus can lead to interesting convergence properties.

5.5.1 Dual Averaging

This dual averaging method uses the dual variables λ_i as weights for the averaging. This choice is quite natural as the dual variable lie on a simplex and measure the relevance of each cut used in determining x^* . Additionally most solvers, when solving the inner problem, use primal-dual methods and thus provide

access to the dual variables at no additional cost. Using this averaging method creates an interesting dynamic on quadratics, assume that :

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x + c, \quad \nabla f(x) = Ax - b, \quad Ax^\star = b$$

Assume as well that the ball constraint is inactive, meaning that the dual problem is feasible. Then, the dual feasibility conditions imply that $p_k = \sum_i \lambda_i g_i = 0$ which yields the following :

$$0 = \sum_i \lambda_i g_i = \sum_i \lambda_i (Ax_i - b) = A \underbrace{\left(\sum_i \lambda_i x_i \right)}_{y_k} - b$$

This means that the averaged iterate y_k is the minimizer of the quadratic function. If the function is quadratic and the bundle is well formed (i.e. the dual problem is feasible), then the dual averaging method converges to the exact minimum in one iteration. This one iteration convergence on quadratics leads us to consider this dual averaging as some sort of second order method. To study the convergence properties of this averaging method, let's make assumptions similar to those made to study Newton's method. Namely, assume the Hessian is ρ -Lipschitz and that $\nabla^2 f(x^\star) \succeq \mu I$ in addition to the function being L -smooth and convex. Under those assumptions, using Lemma 4.1.1 from [19], we have :

$$\|e_i\| = \|g_i - \nabla^2 f(x^\star)(x_i - x^\star)\| \leq \frac{\rho}{2}\|x_i - x^\star\|^2$$

Now using the fact that $p_k = \sum_i \lambda_i g_i = 0$, as we assumed the ball constraint is not active, we have the following :

$$\begin{aligned} \sum_i \lambda_i g_i &= \sum_i \nabla^2 f(x^\star)(x_i - x^\star) + \sum_i \lambda_i e_i = 0 \\ \iff y_k - x^\star &= -[\nabla^2 f(x^\star)]^{-1} \sum_i \lambda_i e_i \end{aligned}$$

This expression can be combined with the bound on e_i to yield the following bound on the distance between the averaged iterate and the optimal solution. Denote $I_k^\star$ the set of indices such that $\lambda_i > 0$ at iteration k , we have :

$$\|y_k - x^\star\| \leq \frac{\rho}{2\mu} \sum_i \lambda_i \|x_i - x^\star\|^2 \leq \frac{\rho}{2\mu} \max_{i \in I_k^\star} \|x_i - x^\star\|^2$$

If the iterates are close enough to the optimal solution, this bound implies that the distance between the averaged iterate and the optimal solution is quadratically smaller than the distance between the iterates and the optimal solution, which is a very interesting convergence property. This property makes the dual averaging method particularly effective on many problems. One can show that the averaged iterate y_k lies in an extended localization set. Notice that $y_k = \sum_i \lambda_i z_i$ as $p_k = 0$, then the following bound appears naturally :

$$\begin{aligned} \psi_j(z_i) &= f_j + \langle g_j, z_i - x_j \rangle + \frac{1}{2L}\|g_j\|^2 \\ &= f_j + \langle g_j, x_i - x_j \rangle + \frac{1}{2L}\|g_i - g_j\|^2 - \frac{1}{2L}\|g_i\|^2 \leq f_i - \frac{1}{2L}\|g_i\|^2 \end{aligned}$$

The regular minimum localization set is defined by $\psi_i(x) \leq f_{\max}^\star$, while it is hard to conclude that $y_k \in \mathcal{X}_k$, it can be shown that it lies in an extended localization set from the above inequality :

$$\psi_j(y_k) = \sum_i \lambda_i \psi_j(z_i) \leq \sum_i \lambda_i \left(f_i - \frac{1}{2L}\|g_i\|^2 \right) = \sum_i \lambda_i s_i$$

This new set has an underlying geometry similar to the minimum localization set but is strictly larger than $\mathcal{X}_k$. It is found by relaxing the right hand side of the localization set constraints with $\sum_i \lambda_i s_i \geq f_{\max}^* = \min_i s_i$. Only the s_i linked to the active constraints contribute to this new right hand side. The combination of these properties make the dual averaging method quite effective in practice and a good practical choice.

5.5.2 Averaging technique comparison

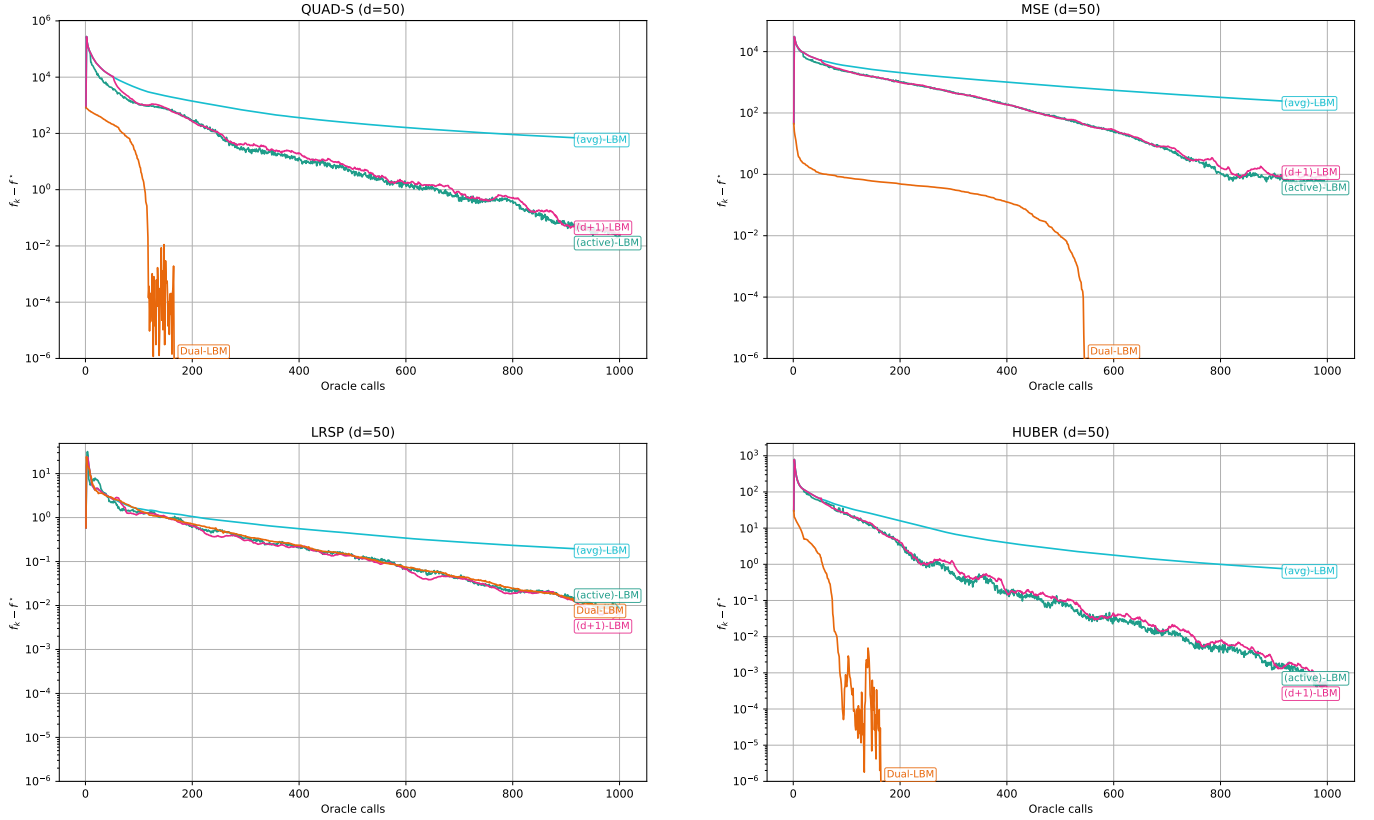


Figure 5.2: Showcase of different averaging techniques on multiple problems. The description of the problems can be found in [chapter 6](#). The dual averaging method is by far the most effective on all problems, especially when the problem is similar to a quadratic.

Averaging technique comparison (Figure 5.2) : shows the different averaging techniques on multiple problems that will be described in the benchmark chapter [chapter 6](#). The simple averaging is quite ineffective as it uses even the old, very bad samples. The last $d + 1$ iterates method is quite effective as it leverages the fact that only $d + 1$ iterates are relevant to the optimization process but it is getting consistently beaten by using only the samples linked to the active cuts. The dual averaging method is very effective on all problems, especially when the problem is similar to a quadratic. For the benchmarking, the dual averaging method will be used as it is by far the most effective.

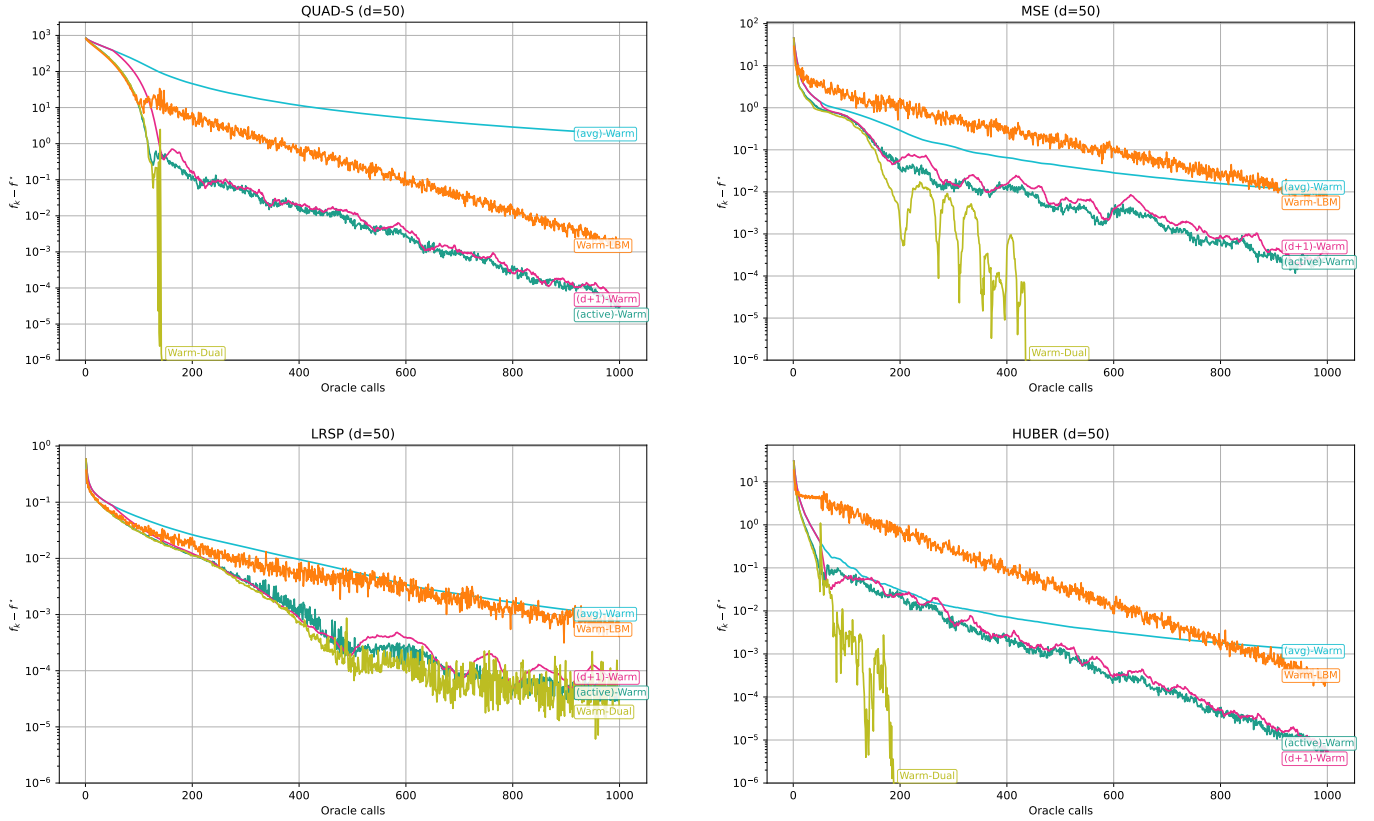


Figure 5.3: Showcase of different averaging techniques using the hyperparameter free warm-start strategy for LBM on multiple problems. The description of the problems can be found in [chapter 6](#).

Averaging technique comparison with warm start (Figure 5.3) : shows the different averaging techniques on the same problems using the hyperparameter free warm start strategy. The results are quite similar to the previous comparison, but do not suffer the fixed cost. The Warm-Dual approach is slightly less performant on (QUAD-S) than the standard dual averaging method but it performs better on (LRSP) and (MSE), making it more well rounded. These results show that the averaging techniques generalize well to the warm start setting.

5.6 Pruning Techniques

The number of samples stored in the model can grow quite large, which can lead to high computational cost when solving the inner problem. ACCPM and Vaidya’s method provide good heuristics to determine how relevant a sample is to the optimization process but not every optimization method has the capability to prune the set of constraints in such an efficient manner. Notably methods such as LBM or the level methods could greatly benefit from such pruning techniques. At iteration k , LBM leverages at most $d + 1$ samples, those samples are the one with non zero dual variable λ_i and zero slack $\langle g_i, x^* \rangle + b_i = t$. This gives some sense as to why it is possible to keep a small bundle while maintaining the convergence properties : at every step only $d + 1$ sample affect the minimizer. This section will focus on LBM and propose pruning techniques that can be used to reduce the number of samples to $\approx 3 - 5d$ without affecting the convergence of the method.

- **Least Recently Used** : simply keep the most recently used samples (sample i is used if $\lambda_i \neq 0$), discarding the ones that haven’t been relevant for a while. This method is very simple and yields

passable results but leverages no additional information on the samples such as their slack or dual variable.

- **Max Slack** : For LBM the slack is $\psi_i(x^*) - t^*$. Since the LBM iterate is defined as the point that maximize the minimum slack, it is natural to remove the sample with the largest slack if the bundle is too big. This method is quite effective in practice and can be easily implemented by keeping track of the slacks of each sample.
- **Max Distance** : an other natural score that one can use to determine the relevance of a sample is the distance between the cutting plane defined by the sample k and the current LP solution x^* . This distance to the hyperplane has been presented in [subsection 2.9.3](#) as v_i , here the distance is mesured from x^* rather than x^* and is given at iteration k for the i -th sample by :

$$d_i^k = \frac{\psi_i(x^*)}{\|g_i\|} = \frac{\langle g_i, x^* \rangle + b_i - t}{\|g_i\|}.$$

If at iteration k , the bundle is too big, then it suffices to remove the constraint with the largest d_i^k .

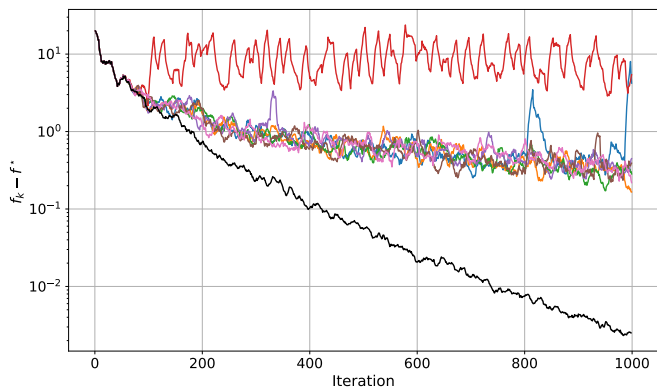
- **Sum of Distance** : instead of looking at the distance at a single iteration, one can look at the sum of distances across iterations and remove those with the largest sum. This method is quite effective in practice as it allows to remove samples that are consistently far from the iterates, which are likely to be less relevant for the optimization process. Similarly one can also define a sum of slacks method.
- **Cumulative Score** : this method is slightly more involved: similarly to *Sum of Distnace*, this method keeps track of a score between iterations. The main difference here is that we choose to accumulate the distance only for the worst half of constraints. This yields the following formula for the score q_i of constraint i at iteration k :

$$q_i = \sum_{j \leq k} \max(d_i^j - \text{median}(d^j), 0)$$

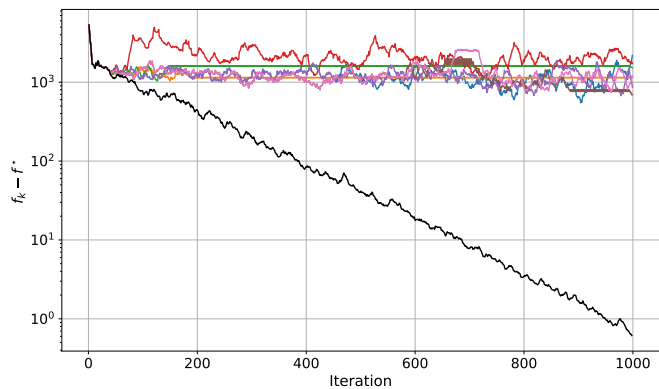
At iteration k , if the bundle is too big, then the constraint with the largest score q_i is removed. This method allows to remove constraints that are consistently far from the iterates.

5.6.1 Technique comparison

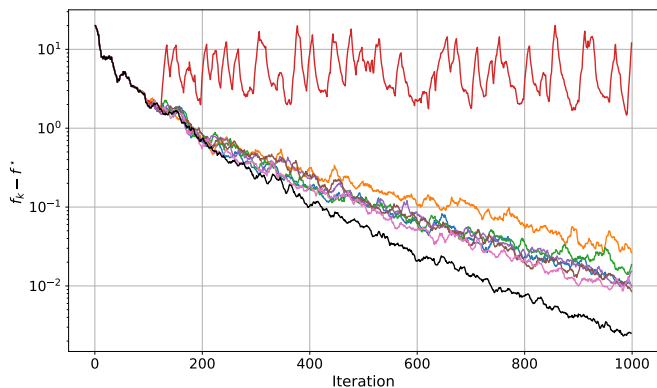
Now let's run a comparison between the different pruning techniques in order to see which methods are the most effective. It is also interesting to analyze how removing constraints affects the convergence rate of the methods. In order to have readable plots, the functional residuals in log space are smoothed using a smoothing window. The different pruning techniques will be tested on two simple problems : a quadratic and a logistic regression in order to asses their performance, the exact problem description can be found in [chapter 6](#) at [section 6.1](#) and [section 6.2](#). The dimension is set to $d = 40$ and the bundle sizes tested are $d, 2d, 3d, 4d, 5d$.



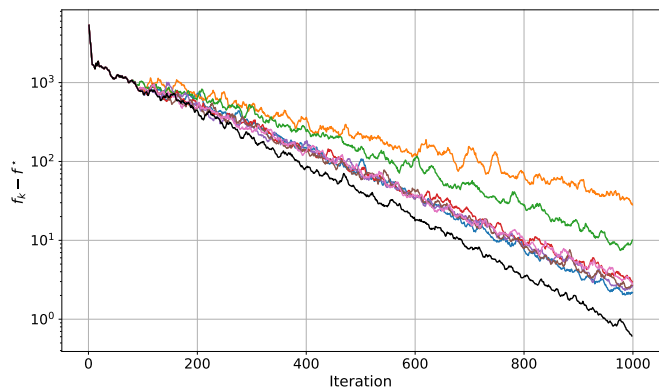
(a) Logistic Regression (section 6.2), $m = d = 40$



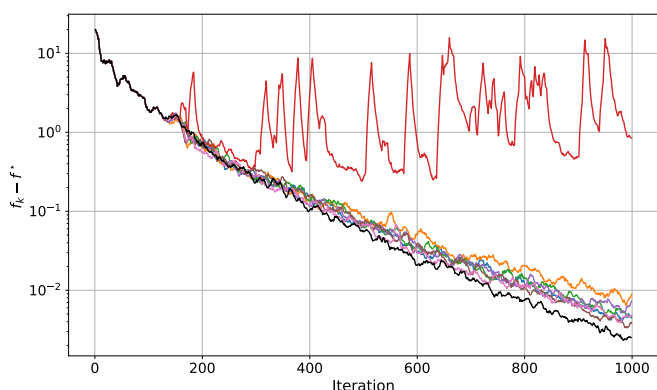
(b) Quadratic (section 6.1), $m = d = 40$



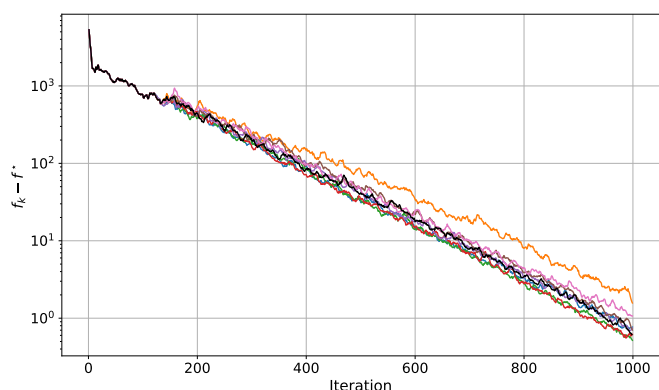
(c) Logistic Regression (section 6.2), $m = 2d = 80$



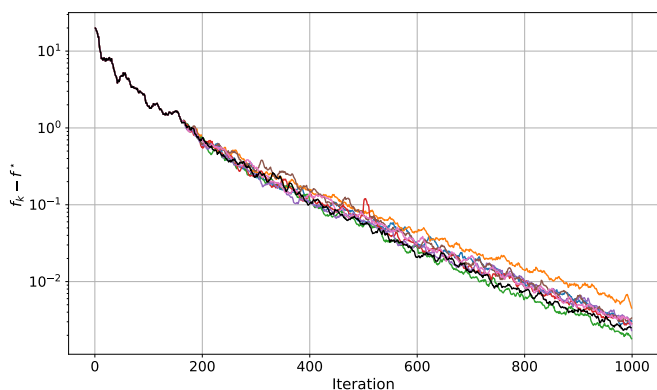
(d) Quadratic (section 6.1), $m = 2d = 80$



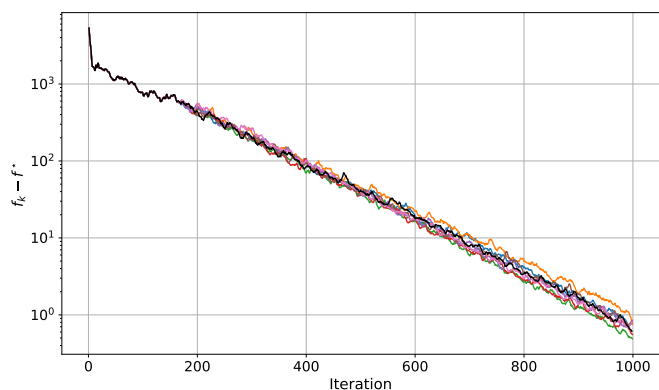
(e) Logistic Regression (section 6.2), $m = 3d = 120$



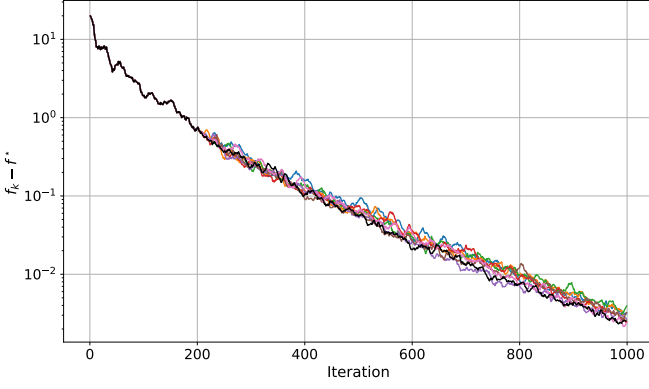
(f) Quadratic (section 6.1), $m = 3d = 120$



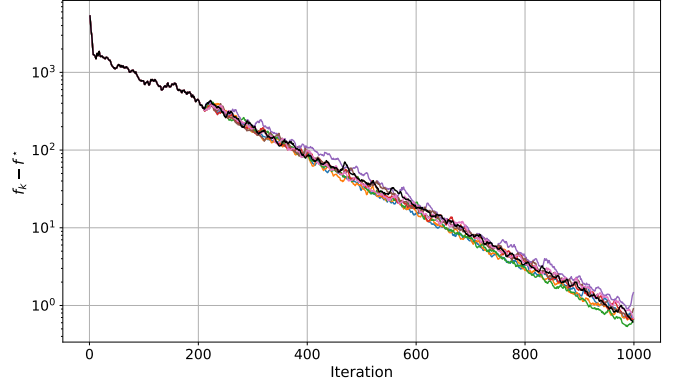
(g) Logistic Regression (section 6.2), $m = 4d = 160$



(h) Quadratic (section 6.1), $m = 4d = 160$



(i) Logistic Regression (section 6.2), $m = 5d = 200$



(j) Quadratic (section 6.1), $m = 5d = 200$

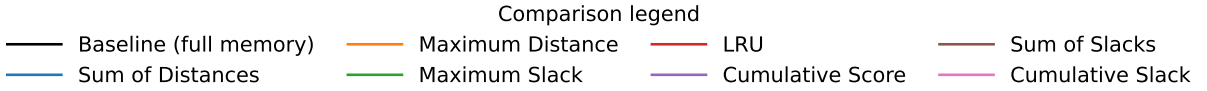


Figure 5.4: Pruning technique comparison across bundle sizes $m \in \{d, 2d, 3d, 4d, 5d\}$ with $d = 40$, on logistic regression and quadratic problems.

Pruning technique comparison (Figure 5.4): All methods have very similar performance, except the maximum-distance, maximum-slack and LRU approaches, which perform worse than the cumulative approaches. When a good cut gets removed the asymptotic convergence rate is usually reduced; this is what happens when there is not enough memory to keep all the important cuts (i.e. when $m < 3d$). The LRU method seems to perform similarly to the others most of the time but sometimes removes a very important cut, leading to a large spike in the functional residual. The maximum-distance and maximum-slack strategies only utilize a single iteration of information, which leads to worse results than the other methods as they remove cuts that are inactive at the current iteration but might become active in the future, reducing the convergence rate. The other methods have almost indistinguishable performance, with the cumulative-score method having a very slight edge. For the benchmarking, that method will be used and the bundle size will be set to $m = 4d$ to ensure that the convergence rate is not affected by pruning.

Chapter 6

Numerical Experiments

This chapter presents the selection of test functions and compares the performance of the different methods on these problems. These functions will cover classical machine learning examples and usual theoretical settings. It also details the generation of these test functions from synthetic data. As theoretical convergence guarantees are difficult to establish for several of the methods considered, empirical performance is used to assess their respective strengths and weaknesses. Multiple parameters play a major role in the performance of the different methods, these are outlined below.

- The dimension of the problem d : any method leveraging the geometry of the localization set will have deteriorating performance as the dimension increases. Increasing the dimension also often results in more expensive subproblems. In order to keep the number of iterations high with respect to the dimension, the number of iterations N is set to $20d$ for all the benchmarks presented in this chapter.
- The smoothness constant provided to the algorithm L : for a given function denote L_f the smallest L such that f is L -smooth. The algorithm can be given any $L \geq L_f$ but the interpolation inequalities will be made looser by providing a larger L , which will results in less informative cutting planes.
- The distance to the minimizer R : most of the algorithms presented earlier require some additional information on the distance to the minimizer, whether it be in the form of an initial polyhedron in the case of ACCPM or a bound on $\|x^* - x_0\|$. Providing a larger R will results in a larger initial localization set, which will results in a larger initial volume and thus a larger initial gap. In this entire benchmarking section, $R = 2\|x^* - x_0\|$.
- The condition number of the function κ : when the function is strongly convex the condition number is given by the ratio between the smoothness constant and the strong convexity constant. A larger condition number will results in a more difficult problem since the curvature of the function is more varying, which will make it harder for the cutting planes to capture the geometry of the function.

As mentioned, ACCPM (Analytic Center Cutting Plane Method) requires a starting polyhedron. Many choices are possible but here we consider the case where, given R , the initial polyhedron is the ∞ -norm ball around x_0 of radius R . This choice results in a larger volume than other methods, especially when d is large. The main reason behind this choice is that computing the analytic center of the intersection of an ℓ_2 -ball and a polyhedron leads to numerical issues due to vanishing Hessian during Newton steps. In practice using an ∞ -norm ball results in very similar performance, while being much more stable. This

choice breaks the invariance by orthogonal transformation; in order to not create artificial structure, the ∞ -norm ball is rotated by a random orthogonal matrix (following the Haar measure), yielding $P_0 = \{x \mid \|Q(x - x_0)\|_\infty \leq R\}$. This approach ensures no specific coordinate system is favored.

For LBM there will be four different graphs presented : one without warm start, one with warm start. The dual average of the both variant are also presented. The warm start used is the hyperparameter free warm start described in [subsection 5.4.3](#). Only the dual averaging approach is presented as it was consistently the best averaging approach across all the benchmarks. For the level methods : Level, S-Level (smoothness augmented level method) and LLM (Level Method with the optimal lower bound), the parameter is set to $\alpha = 0.8$. For the pruning handlers, ACCPM and S-ACCPM (ACCPM with smoothness augmented cutting planes) do not need constraint pruning as their execution time is relatively low. On the other hand, level methods and LBM all use the same Cumulative score pruning handler, which was outlined in [section 5.6](#). These methods keep $m = 4d$ constraints in the bundle, which should not deteriorate meaningfully their performance.

In order to keep the comparison graph readable, the plots only display the best residual so far ($\min_k f_k - f^*$) for each method. Some methods such as Gradient Descent, GMM (Gradient with memory, using the piece-wise affine model) and FGM will be left out of the comparison as they don't bring much insight. FGM (Fast Gradient Method) is very similar to OGM (and slightly worse), IGMM (GMM using the optimal lower bound) is a better variant of GMM, and Gradient Descent is known to be much worse than the other methods. a-PBM (smoothness accelerated proximal bundle method) acts like a less performant version of OGMM (Optimized version of IGMM) and is thus not included in the comparison to make plots easier to read. Vaidya's method is also left out of the comparison as it requires much larger computational power to run and is not generally used in practice, even in the convex non-smooth setting.

6.1 Quadratic Functions

Quadratic functions are present in almost every smooth convex benchmark, which is due to their inherent link to smoothness. They are known to exhibit the worst-case convergence rates for optimal first order methods. Notably OGM achieves it's worst case rate on $f(x) = \frac{L}{2}\|x\|^2$ [14]. For this benchmark, we consider the following function :

$$f(x) = \frac{1}{2}x^T Qx, \quad Q = \text{diag}(v) \succeq 0$$

Since choosing $v_i = 0$ simply results in a flat direction contributing nothing to the objective, we consider the case where $v_i > 0$ for all i . This choice may seem overly simple but due to the invariance properties outlined in [section 5.1](#) it will not affect the performance. Indeed, through a transformation of type $x \rightarrow Qx + b$, it is possible to recover the general dense quadratic from the diagonal case (assuming $v_i \neq 0$), one can also recover the rank deficient case by using the low rank transformation.

The vector v will decide on the spectrum of the function, and thus the condition number. Here two approaches are possible, either one chooses the eigenvalues to be on a logscale from the largest to smallest (denote this problem QUAD-L) or one uses the same approach as the one outlined in [7], where the diagonal elements are $v_i = \sin^2\left(\frac{\pi i}{2n}\right)$ (denoted QUAD-S). Here we consider the starting point $x_0 = x^* + Q^{-1}\mathbf{1}$. This is often considered as a hard starting point as it amplifies directions with small

curvature, which are usually the most difficult to reduce.

This benchmark also covers the case of mean squared error minimization $\min_x \|Ax - b\|_2^2$, which is a common problem in machine learning. In order to also evaluate the performance of the different methods on this common problem, the same approach as [22] is used. It consists in generating each element of A and b using an i.i.d. standard Gaussian distribution and choosing $x_0 = 0$ (denote this problem as MSE). The goal is to generate a problem with eigenvalues following the Marchenko-Pastur distribution to evaluate the method in the average setting.

6.1.1 Baseline benchmark

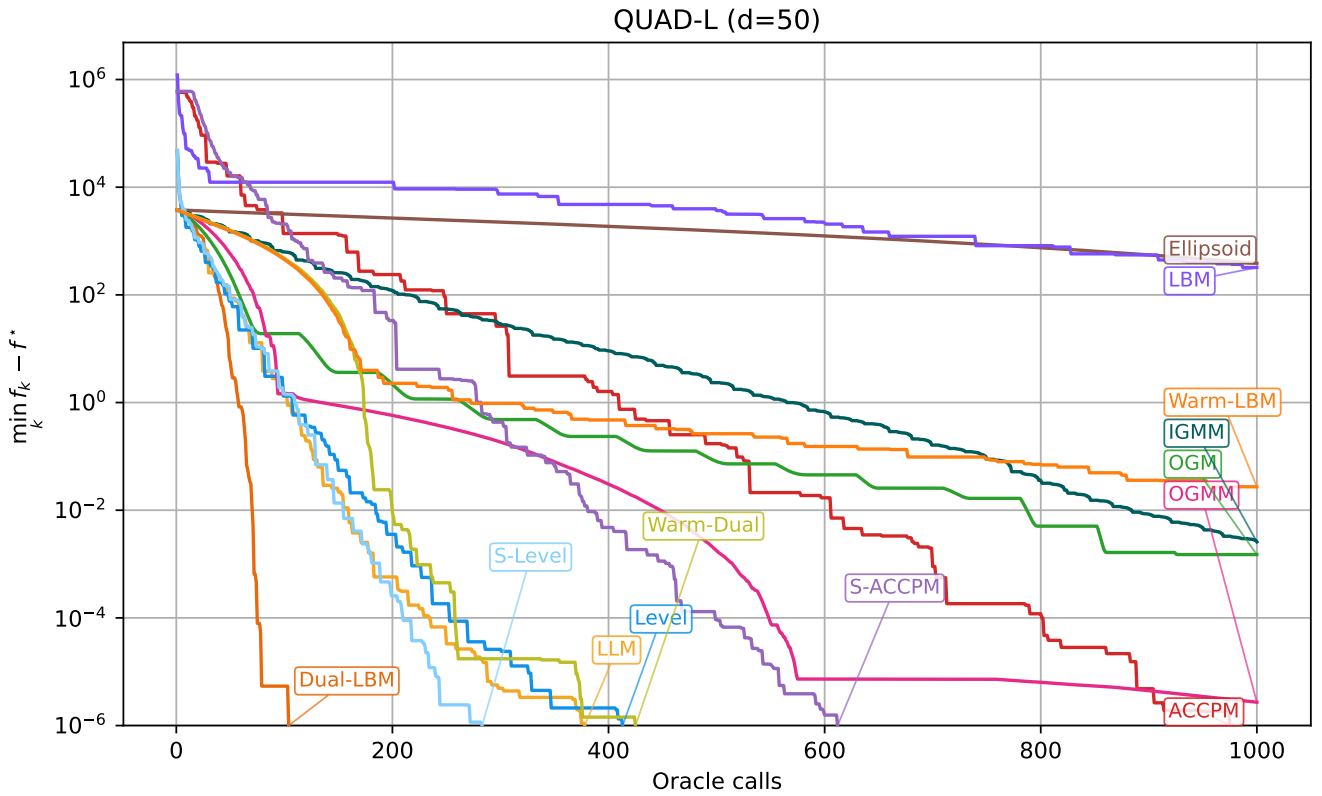


Figure 6.1: Performance of the different methods on (QUAD-L) for $d = 50$ and 1000 oracle calls. The QUAD-L problem has $\kappa = 10^3$

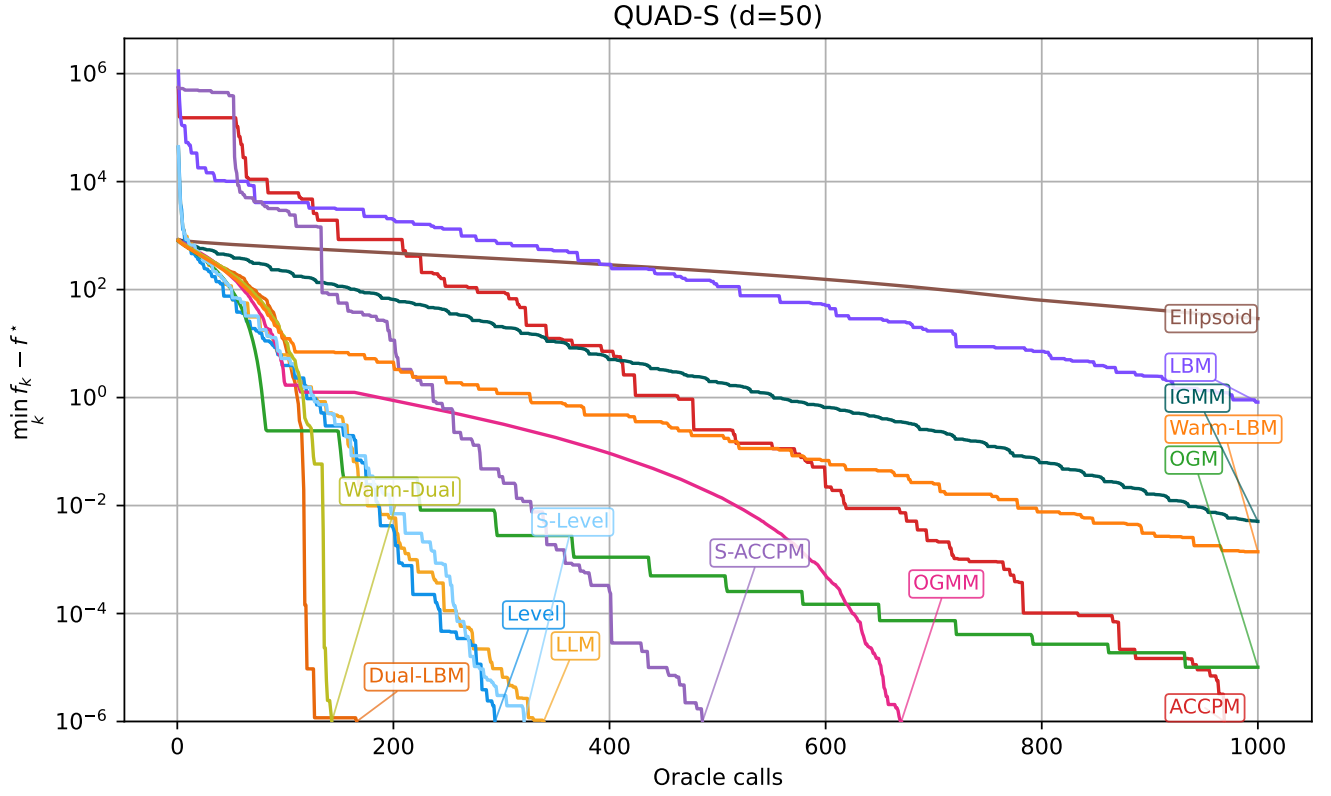


Figure 6.2: Performance of the different methods on (QUAD-S) for $d = 50$ and 1000 oracle calls. The condition number is $\kappa \approx 10^3$.

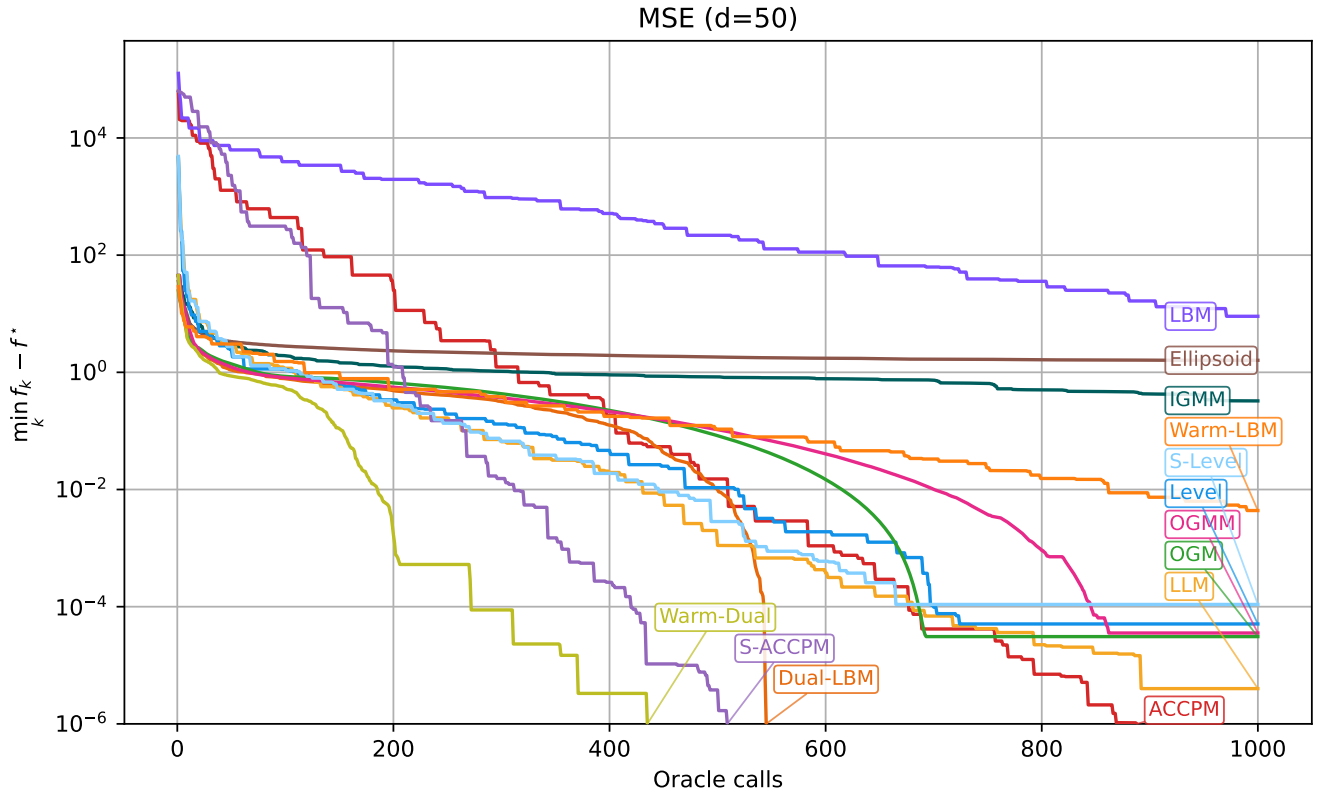


Figure 6.3: Performance of the different methods on (MSE) for $d = 50$ and 1000 oracle calls. The condition number of the problem is $\kappa \approx 6.6 \cdot 10^4$.

Performance on quadratic problems (Figure 6.1, Figure 6.2, Figure 6.3) The level methods (Level, S-Level and LLM) perform very well on these problem, showing what appears to be a linear convergence

rate. While S-Level seems to be performing better on QUAD-L, the performance of the 3 level methods is almost identical on QUAD-S and MSE. Overall, for the level method it is hard to determine whether smoothness brings a significant improvement or not.

The Dual-LBM method is the best performer on all 3 problems which is probably due to the one step convergence property on quadratics when the bundle is well posed. LBM is getting outperformed significantly by the warm started variant, which is to be expected since the warm start negates the biggest part of the fixed cost and offers the same asymptotic convergence rate. While Warm-Dual performs worse than the regular Dual-LBM method it is still a strong performer, offering performance comparable to the level methods.

ACCPM performs well on both problem but it is outclassed by the smooth version S-ACCPM which tends to use about half the number of iterations to reach the same tolerance. Similarly on the 2 pure QUAD problems, the smooth ellipsoid method outperforms the standard one, on MSE the performance seems similar. However, in both cases, their performance is quite disappointing, which is due to the relatively large dimension ($d = 50$).

On these problems, the more traditional methods are getting outclassed by quite a margin, the best performers tends to be OGMM and OGM, which makes sense since they should both reach an optimal convergence rate on smooth convex function. Note that here the problem is strongly convex, thus regular gradient methods and particularly IGMM all reach a linear convergence rate, but due to the relatively high conditioning, convergence is still quite slow. In any case, performance on quadratic problems is informative but not a decisive factor in method selection as the conjugate gradient method is known to have finite termination on these kind of problems (in exact arithmetic). It is still a good first test in order to asses the strengths and weaknesses of the different methods. Let us now study what happens when changing the dimension and the smoothness constant provided to the algorithm.

6.1.2 Benchmark with increased dimension

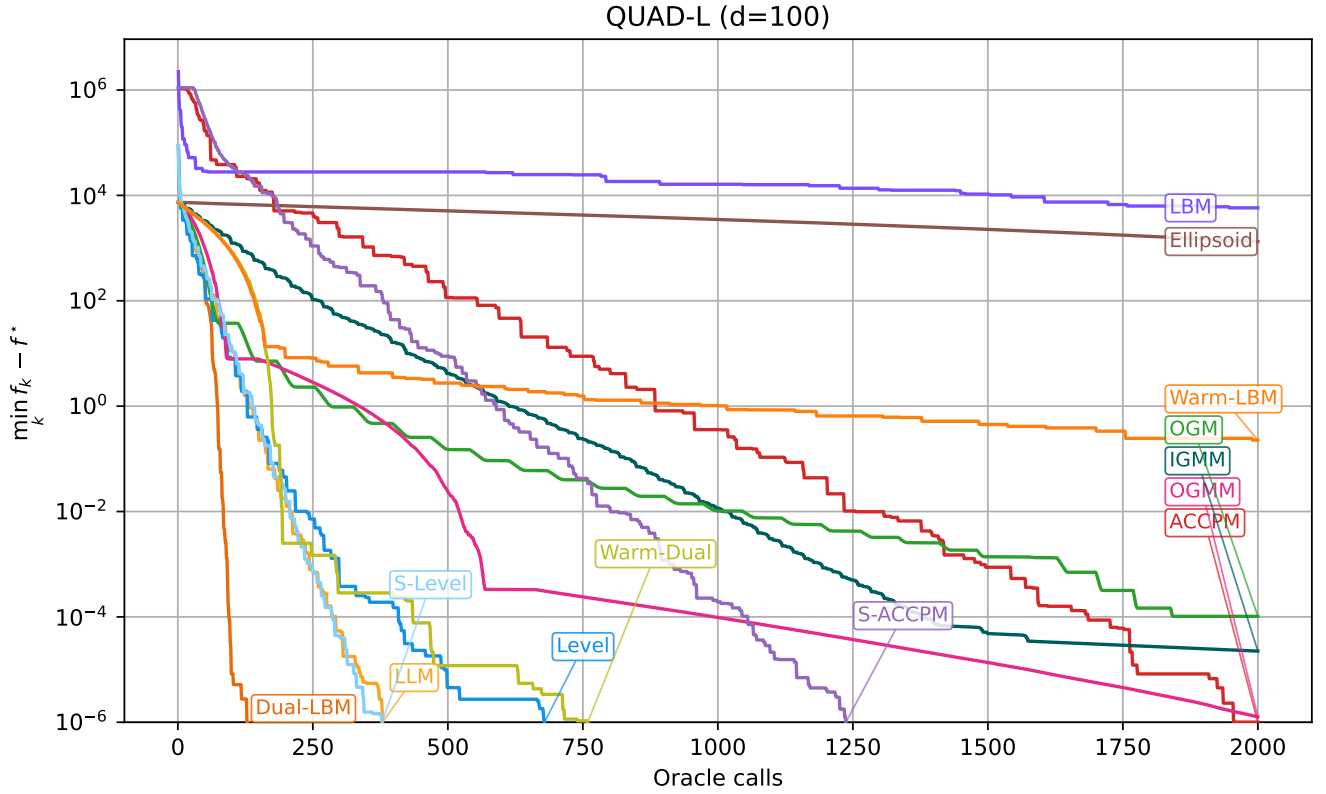


Figure 6.4: Performance of the different methods on (QUAD-L) for $d = 100$ and 2000 oracle calls. The QUAD-L problem has $\kappa = 10^3$.

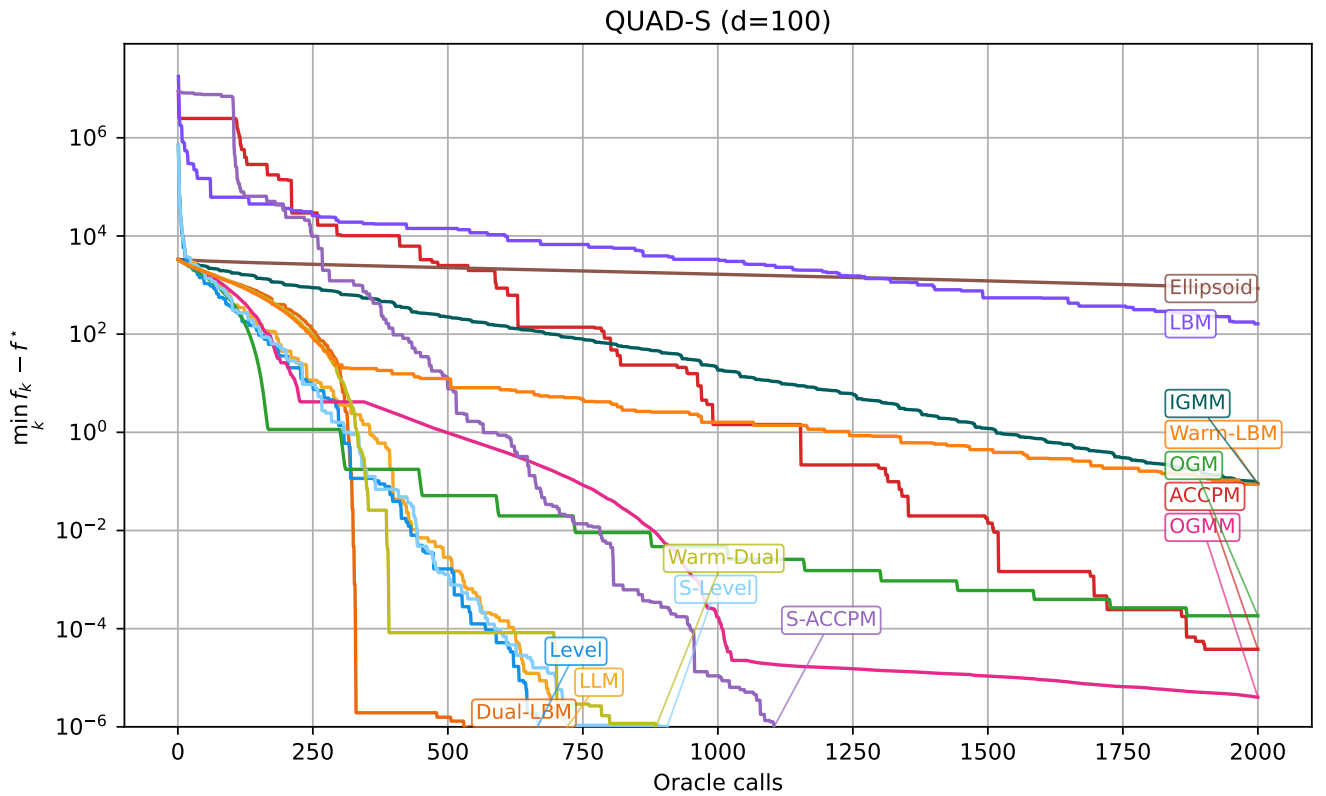


Figure 6.5: Performance of the different methods on (QUAD-S) for $d = 100$ and 2000 oracle calls. The condition number is $\kappa \approx 4.05 \cdot 10^3$.

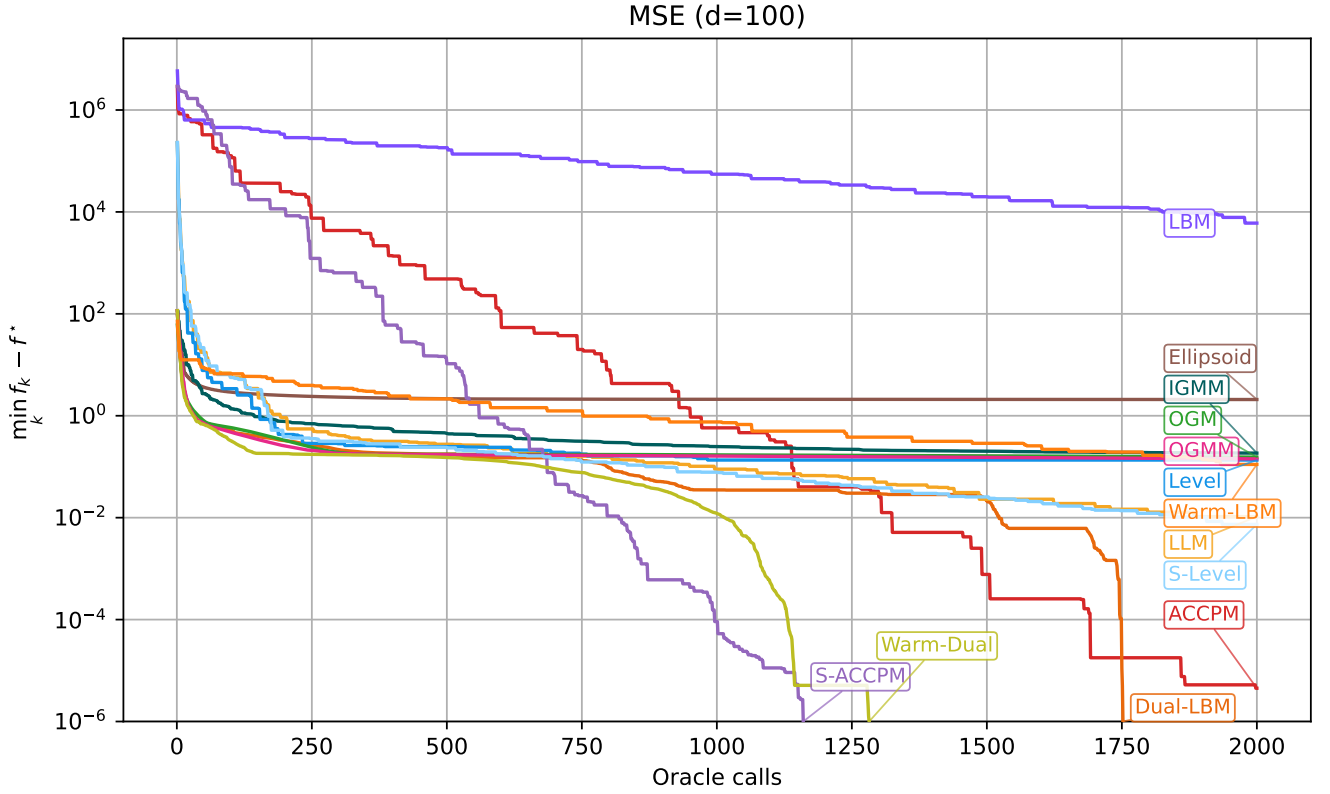


Figure 6.6: Performance of the different methods on (MSE) for $d = 100$ and 2000 oracle calls. The condition number of the problem is $\kappa \approx 1.55 \cdot 10^7$.

Performance on higher dimensional quadratic problems (Figure 6.4, Figure 6.5, Figure 6.6) As expected, the performance of the cutting plane methods is significantly worse on these higher dimensional problem, ACCPM and the level methods seems to require about twice the amount of iteration to reach the same tolerance as the one used in the $d = 50$ case. The Dual-LBM approach also requires more iterations to converge as, if the dimension is higher, more cuts are required to reach the well formed bundle condition. Warm-Dual suffers similarly. On the other hand, the more traditional first order methods are not affected as much by the increase in dimension as they don't leverage the geometry of the localization set as much. It is hard to compare performance of level methods on the MSE problem because some method run into numerical issues. Also note that the conditionings on MSE and QUAD-S are not exactly the same as the previous section as they are not imposed explicitly by their generation process.

6.1.3 Benchmark with increased L

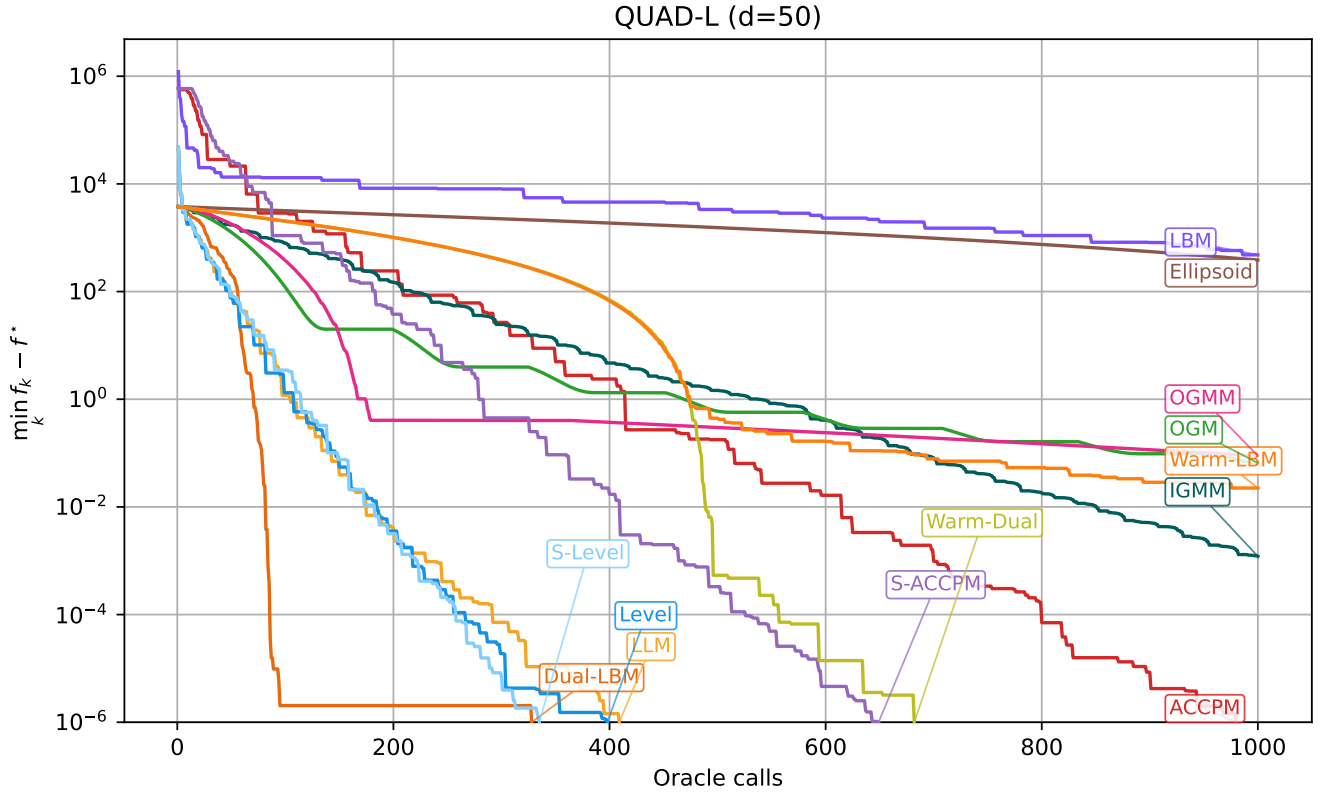


Figure 6.7: Performance of the different methods on (QUAD-L) for $d = 50$, $L = 3L_f$ and 1000 oracle calls. The QUAD-L problem has $\kappa = 10^3$

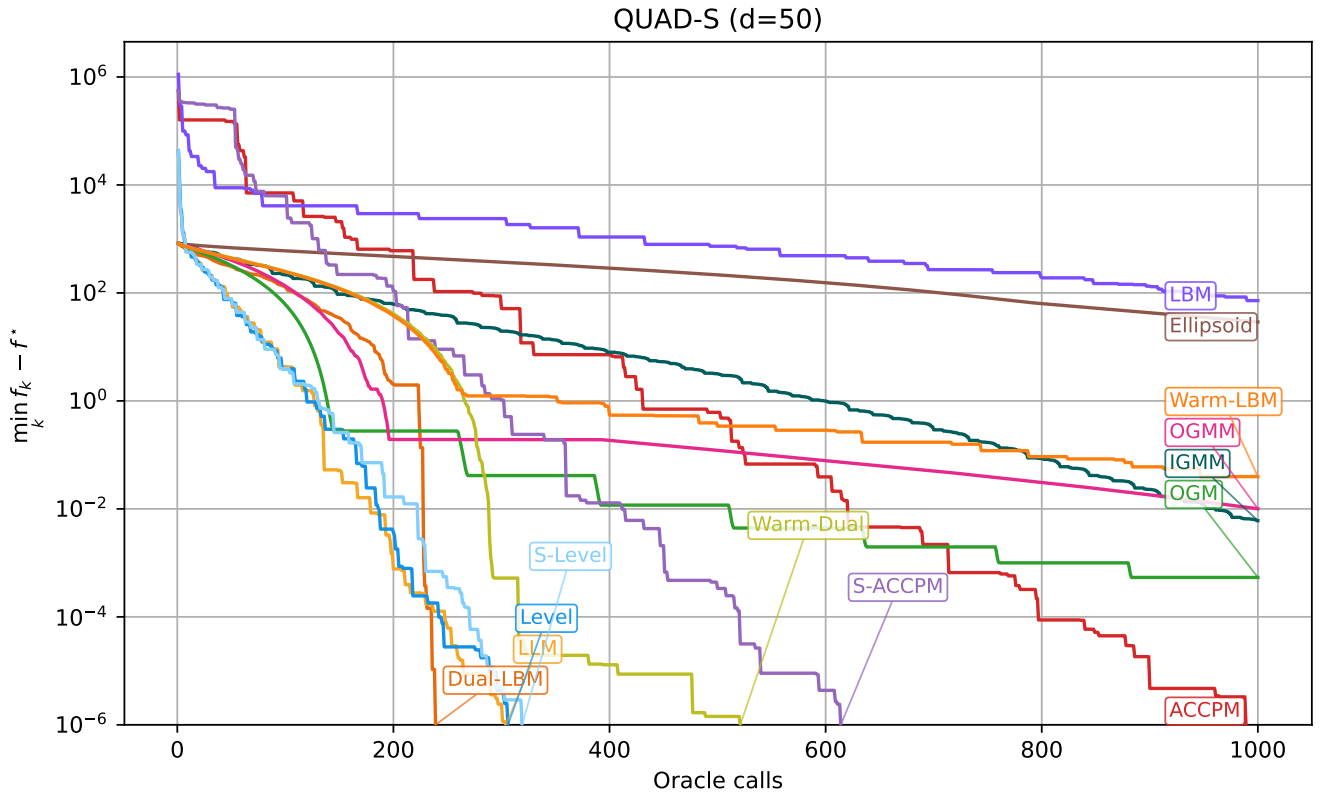


Figure 6.8: Performance of the different methods on (QUAD-S) for $d = 50$, $L = 3L_f$ and 1000 oracle calls. The condition number is $\kappa \approx 10^3$.

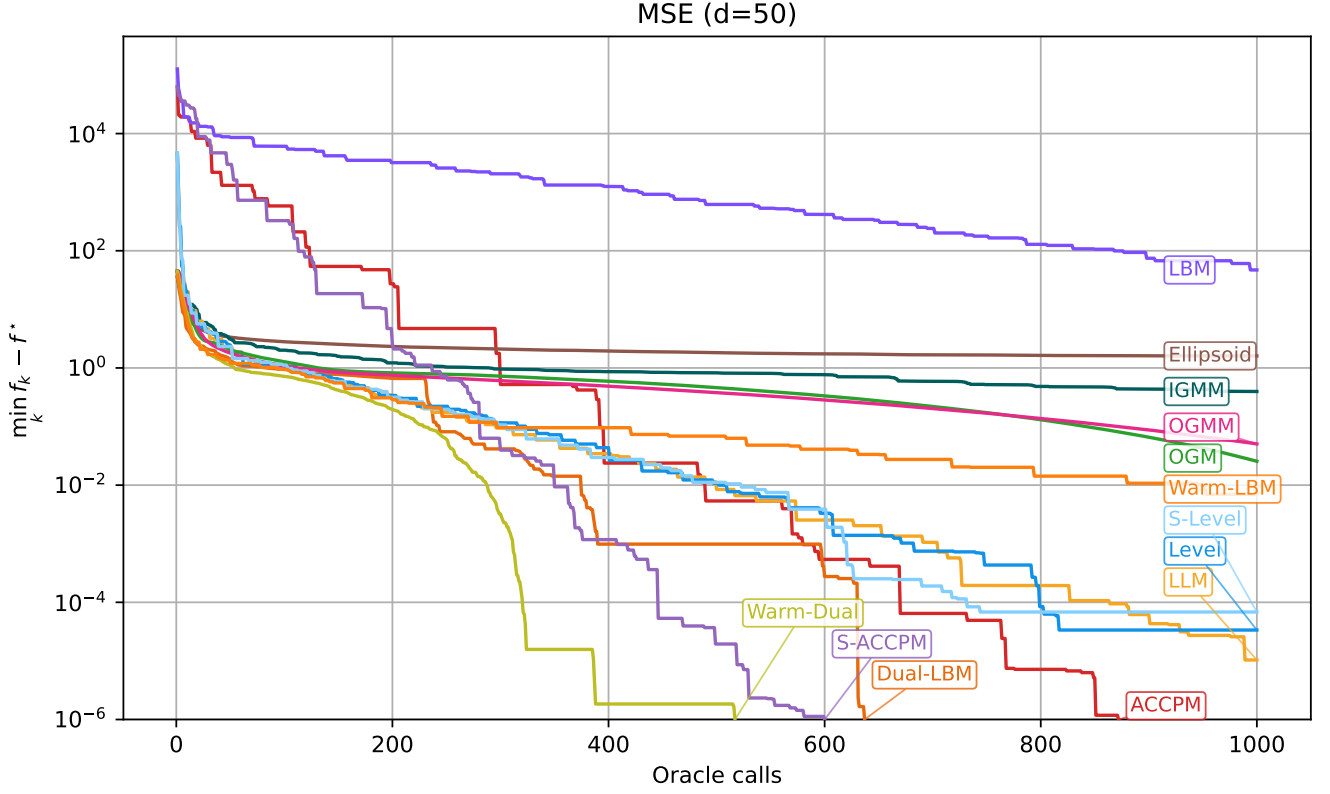


Figure 6.9: Performance of the different methods on (MSE) for $d = 50$, $L = 3L_f$ and 1000 oracle calls. The condition number of the problem is $\kappa \approx 6.62 \cdot 10^4$.

Performance with increased L (Figure 6.7, Figure 6.8, Figure 6.9). Here the problems are the exact same as the one presented in the baseline benchmark, but the smoothness constant provided to the algorithm is 3 times larger than the actual smoothness constant of the function. This results in looser interpolation inequalities and thus less informative samples. As expected, this change only affect methods that rely on the smoothness constant explicitly : methods such as the Ellipsoid method, regular ACCPM and the regular level method are not affected. All the other methods relying on the smoothness constant see some deterioration in their performance, notably regular first order method such as OGM, IGMM and OGMM are strongly affected. The warm started variants of LBM suffer particularly as their initial radius is much smaller and thus takes a longer time to reach the same performance. This benchmark is not as interesting as the others as the results are quite straightforward : providing a larger L results in a loss of performances, asymptotically, methods relying on cutting plane will tend to their convex non-smooth version for $L \rightarrow \infty$.

6.2 Logistic Regression

Another very common example of practical problem considered in the smooth convex litterature when evaluating performance is the logistic regression problem. It is a canonical problem in machine learning with many practical applications. Contrary to quadratics, the curvature is not constant it varies with current iterate through the sigmoid function $\sigma(t) = (1 + \exp(-t))^{-1}$. For the labels $y_i \in \{\pm 1\}$, the problem is defined as follows :

$$f(x) = -\frac{1}{n} \sum_{i=1}^n \log(\sigma(y_i a_i^\top x))$$

where a_i are the n data points. Let $A = [a_1, \dots, a_n] \in \mathbb{R}^{n \times d}$. The hessian is given by :

$$\nabla^2 f(x) = \frac{1}{n} A W(x) A^\top, \quad W(x) = \text{diag}(\sigma(z_i)(1 - \sigma(z_i)))_{i=1}^n$$

with $z_i = y_i a_i^\top x$. This expression shows that logistic regression can be viewed as a weighted quadratic problem: the matrix $A^\top A$ determines the underlying geometry, while the diagonal weights $W(x)$ modulate the curvature according to the confidence of each sample. When z_i is close to zero the curvature is high whereas point with a large $|z_i|$ have experience small curvature. As a result, logistic regression behaves like a quadratic objective whose curvature changes smoothly across the domain and may become nearly flat in saturated regions. Note that this function is strictly convex.

The approach used is similar to the one proposed in [7] and consists in building a matrix A that is sparse (1.0% of non zero entries) the labels are then sampled from $\{\pm 1\}$ using the probability distribution $\mathbb{P}(y_i = 1) = \sigma(a_i^\top x_0)$, with x_0 is the same starting point as the one used for the quadratics (denote this problem LRSP). This approach results in an ill-conditioned problem, which is known to be difficult for most first order methods. The optimum is computed by running L-BFGS until a tolerance of 10^{-12} is reached. The starting point is set to $x_0 = 0$, which is standard for logistic regression.

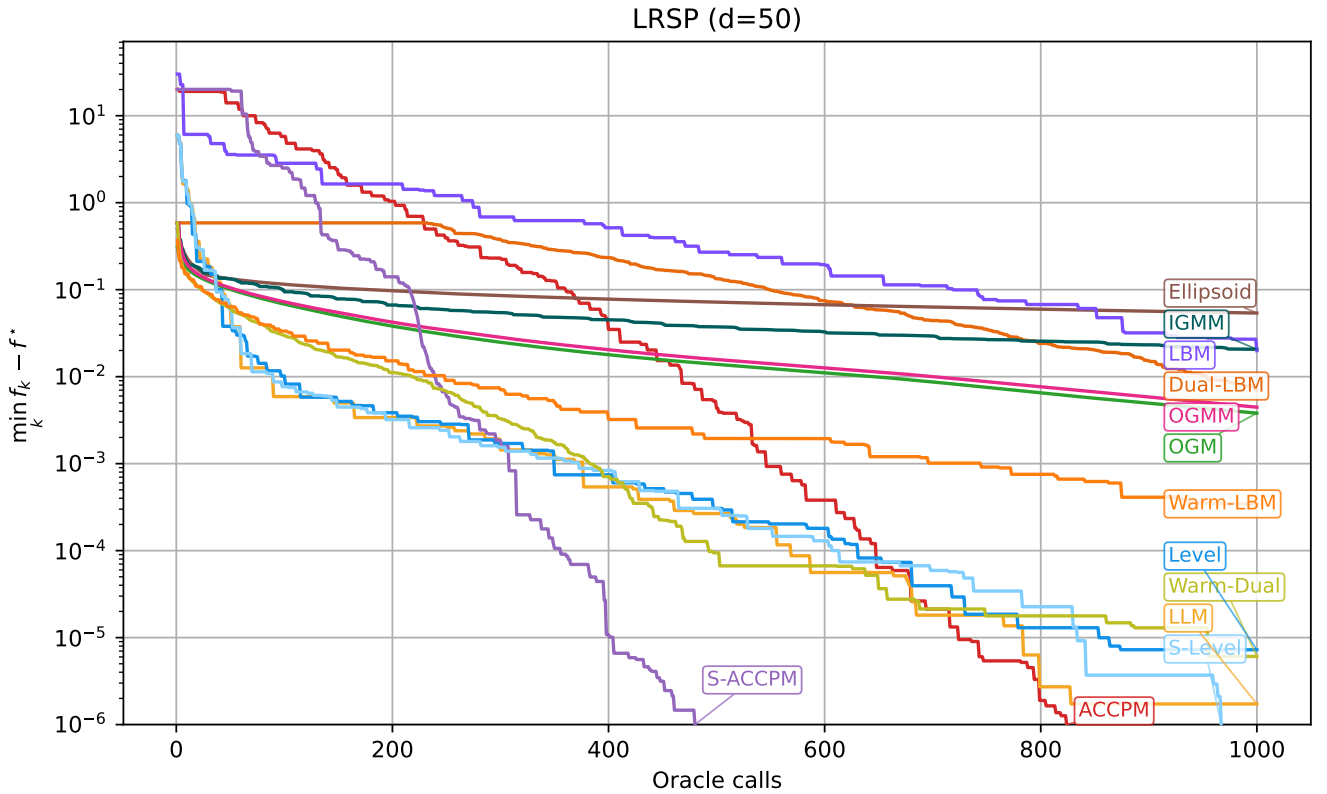


Figure 6.10: Performance of the different methods on (LRSP) for $d = 50$ and 1000 oracle calls.

Performance on logistic regression (Figure 6.10) : The performance of the different methods on this problem is very similar to that of the quadratics in the sense that the top performers remain the same. S-ACCPM gives very similar results in terms of number of oracle calls to reach tolerance. The others top performers remain the level methods and the Warm-Dual LBM which exhibit much better results than classical methods such as OGM, OGMM and IGMM. Warm-LBM also outperforms the regular methods but not by as large a margin as the level methods and is once again much better than regular LBM. The performance of the Ellipsoid method is once again really disappointing due to the large dimension.

6.3 Huber Regression

The third type of problem considered is problem sometimes used in some smooth convex benchmarks (see [12]). Coming from machine learning, the Huber regression problem has been proposed as a way to make linear regression more robust to outliers. The huber loss is a smoothing of the absolute value function, define the cutoff δ and the function ℓ_δ as follows :

$$\ell_\delta(t) = \begin{cases} \frac{1}{2}t^2 & \text{if } |t| \leq \delta \\ \delta(|t| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

This loss function is 1-smooth convex and behaves like a quadratic function for small values of t and like an absolute value function for large values of t . It also has a discontinuous second derivative at $t = \pm\delta$. The problem is defined as follows :

$$f(x) = \frac{1}{n} \sum_{i=1}^n \ell_\delta(a_i^\top x - b_i)$$

This yields a problem that is smooth convex but not strongly convex, allowing evaluation of the performance of the different methods in a setting where the curvature can be 0 (linear part of the loss) and changes abruptly (at the cutoff). The data generation follows the same approach as the one presented in [12] : the matrix $A \in \mathbb{R}^{(d+1) \times d}$ is sparse with 1's on the main diagonal and -1 's on the first subdiagonal. The vector b is taken to be all 1 except for the final entry that is $-1.1d$. That b is close to a vector that would yield a zero loss but the last entry creates a small gap between the optimal value and 0, which makes the problem more challenging. In order to asses how curvature affect the different methods, two different values of δ are considered : $\delta = 200$ which results in a large region with high curvature and $\delta = 0.2$ which results in a much smaller region with high curvature (denote this HUBER-flat). The starting point is set to $x_0 = 0$. The optimum is computed by running L-BFGS until a tolerance of 10^{-12} is reached.

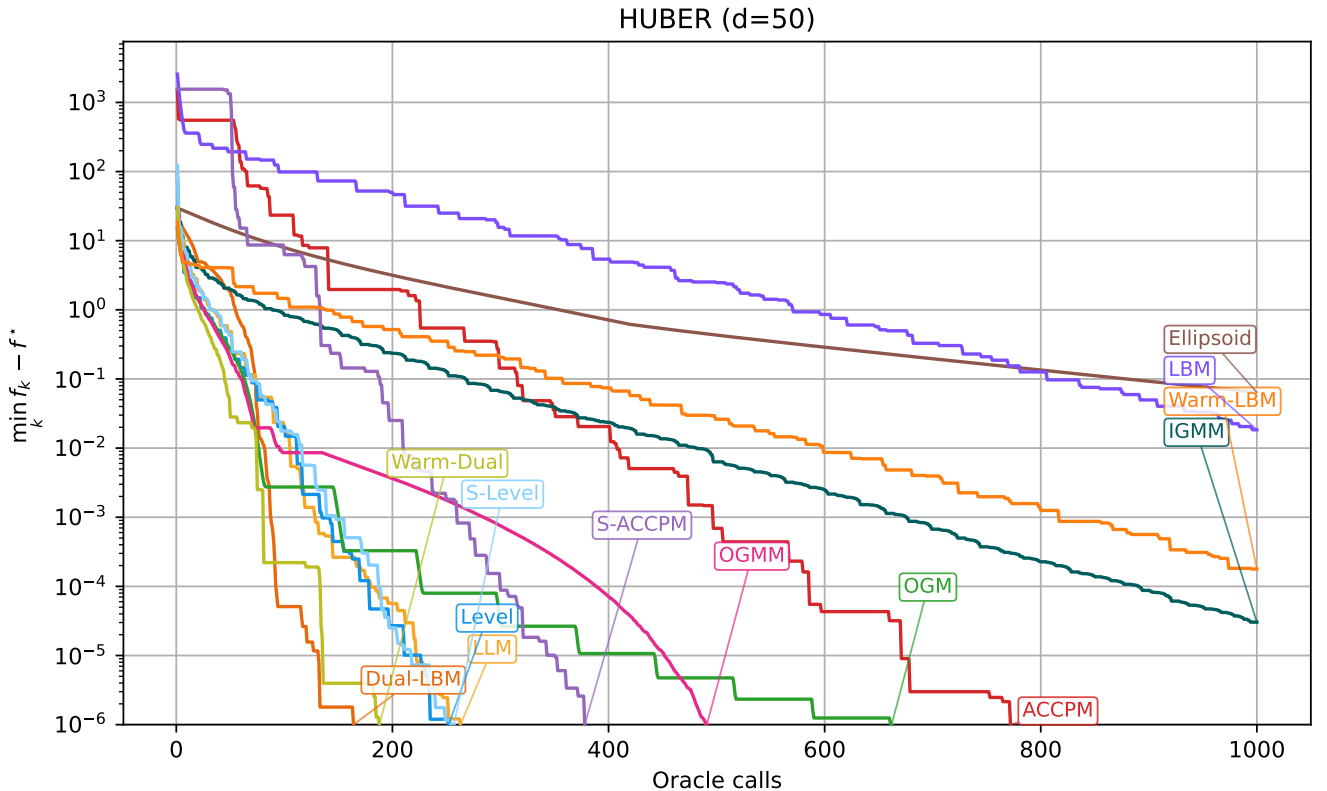


Figure 6.11: Performance of the different methods on (HUBER) for $d = 50$, $\delta = 200$ and 1000 oracle calls.

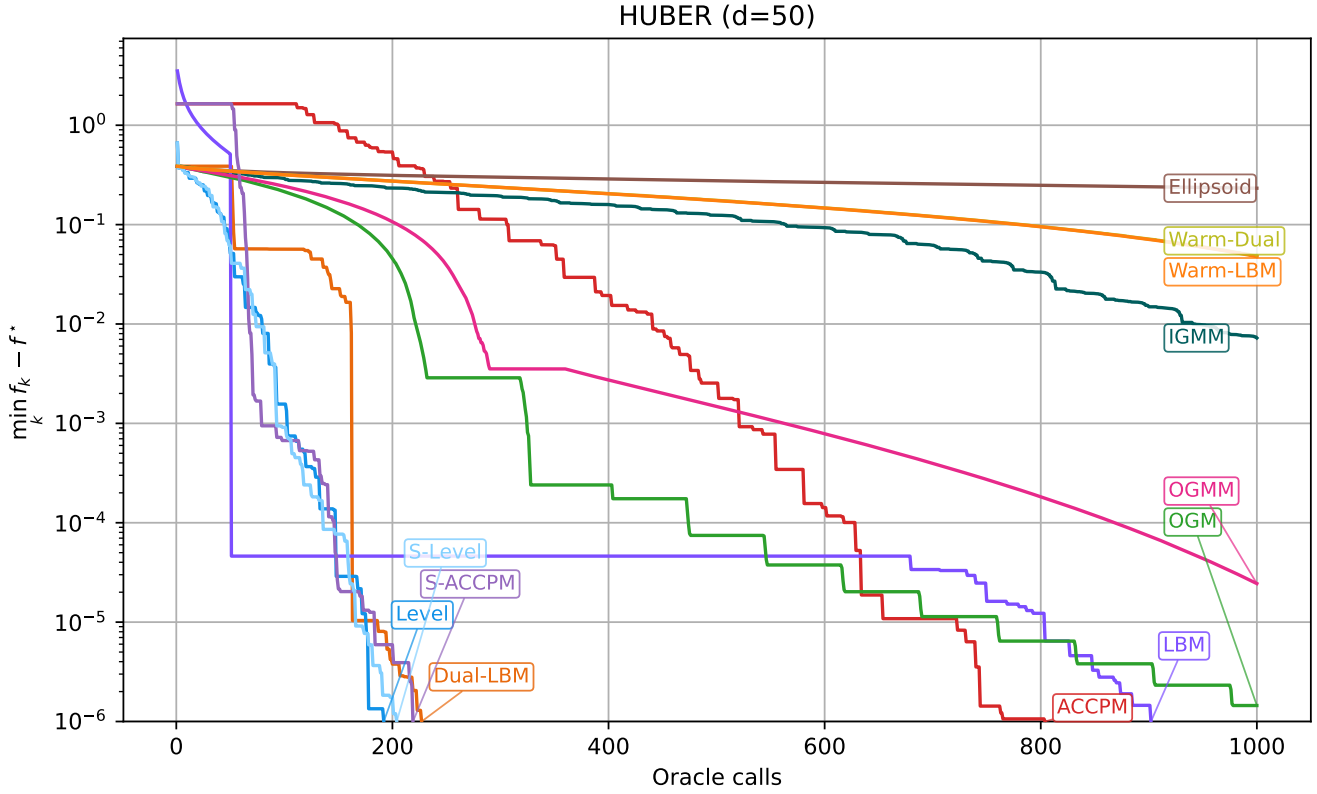


Figure 6.12: Performance of the different methods on (HUBER-flat) for $d = 50$, $\delta = 0.2$ and 1000 oracle calls.

Performance on Huber regression (Figure 6.11, Figure 6.12) The performance of the different methods on this problem is quite interesting. The level methods perform very well once again, showing what appears to be a linear convergence rate, which is quite surprising given the non strong convexity and the abrupt change in curvature. The Dual-LBM method also performs very well, which probably due to the inherent quadratic structure of this problem. ACCPM and S-ACCPM offer very similar performance to the other tests. OGM and OGMM seem to be able to exploit the structure quite well despite the sudden curvature change. LBM, IGMM and the Ellipsoid method display linear convergence rate as well but are much slower than the top methods. Notably, on the second graph, δ is set to 0.2, which results in a much smaller region with high curvature, which implies a lower difference between the starting point and the optimal solution. Interestingly, the performance of the warm start variant of LBM becomes much worse than the regular LBM, which is due to the radius given by the warm start not increasing fast enough when the curvature is small. Regular first order methods such as OGMM, OGM and IGMM also see a deterioration of their performance. Level methods only see a slight decrease in performance, which is probably due to them not being as reliant on curvature.

6.4 Overall Assessment

This section outlines the main conclusions that can be drawn for each method from the different benchmarks presented above. It will also cover their main drawbacks in terms of computational cost, numerical stability and sensitivity to dimension. It is important to understand that while this benchmark covers classical smooth convex optimization problems, it is not exhaustive and doesn't provide theoretical guarantees that are usually used to compare methods in the smooth convex setting. The dimension of the problems considered is also relatively small compared to what can be encountered in practice. The goal of this master thesis is to explore the potential of memory in the cutting plane regime where the number

of iterations is small compared to the dimension.

LBM based methods : this new kind of method is interesting, providing a linear convergence rate on all the different problems studied. The warm start approach also negates the fixed functional residual cost of the method, while still providing the same asymptotic convergence rate (when choosing the right warm start approach). The dual averaged version on the other hand, seems to be a quite strong variant of the method, especially on problems displaying an explicit quadratic structure (such as the quadratics in [section 6.1](#) and the huber regression [section 6.3](#)). The combination of both into the Warm-Dual version yields a generally quite efficient algorithm that can be computed as a byproduct of the warm start inner problem. The performance seems to decrease strongly with dimension, so these methods are only competitive for low dimensional problems. Their computational cost is relatively low compared to other methods (very easy QP / LP).

Smooth Ellipsoid / Ellipsoid : these methods perform very similarly on all problems and the added smoothness doesn't seem to bring enough to the table to make them attractive. The performance of these methods degrade a lot with the dimension and they are outperformed by a large margin by both the other cutting plane methods and regular first order methods. Both versions also run into numerical issues as dimension increases, which is due to numerical instability changing the definiteness of the matrix. Even when using special techniques to improve numerical stability, the methods can still run into problems and are thus not practical. Overall, these methods are not really competitive for these kind of problems.

S-ACCPM / ACCPM : these methods perform very similarly on all benchmarks which is not surprising as it is a method leveraging geometric properties of polyhedrons rather than smooth convex interpolation conditions. This method is quite robust to the conditioning and offers a good performance on all benchmarks. Since it does not utilize much of the structure of smooth convex functions, the improved cutting planes provided by smoothness tend to have a much greater impact than on the other methods. The dependence on dimension of ACCPM is quite strong as doubling the dimension seems to require about twice the number of iterations to reach the same tolerance. Using a local newton steps approach, the computational cost of each iteration is quite low, depending mainly on the number of cuts. Overall, it seems to be a strong contender for problems of moderate dimension.

Level methods : these methods performed really well on most problems, showing what appears to be a linear convergence rate on the quadratics and huber regression. The logistic regression benchmark however, shows that the performance of these methods is strongly linked to the problem itself and even though they provided good performance, they might deteriorate even further on some hard problems (and offer a less than linear rate). The computational cost of these methods is also non negligible since they require solving a QP at each iteration in addition to solving the LBM supproblem (which is a QP or a LP depending on the boundedness setup). They also seem to become somewhat numerically unstable on some problems, which is a major drawback.

Chapter 7

Conclusion

This thesis investigated smooth convex optimization from a perspective centered on the full exploitation of oracle information. The motivating observation was that standard first-order methods discard most of the history of past evaluations, and the central question was whether retaining and combining this information more aggressively could lead to algorithms that are more effective in practice, particularly in settings where oracle calls are costly and their number is small relative to the ambient dimension. This question is further motivated by the classical lower bounds for first-order methods, which are designed for regimes where the number of iterations is small compared to the dimension. In a fixed finite-dimensional space, these bounds leave open the possibility of doing better, and the methods developed here aim to exploit precisely this gap.

It is worth situating this work relative to a parallel line of research that also exploits the full oracle history in the smooth convex setting, namely GMM, IGMM, OGMM, and a-PBM. These methods achieve dimension-uniform convergence rates, with OGMM and a-PBM matching the optimal $\mathcal{O}(1/k^2)$ worst-case rate, and they do so by constructing global smooth lower models from accumulated oracle information. However, their design is fundamentally different from the cutting-plane approach: they base themselves on regular OGM and Inexact Proximal Steps respectively and are analyzed in terms of worst-case oracle complexity uniformly in the dimension. The cutting-plane methods studied in this thesis take a different stance, explicitly targeting the finite-dimensional regime where the number of iterations is small relative to the dimension and where the geometry of the localization set can be exploited directly. The two approaches are therefore complementary rather than competing, and the question of whether they can be unified into a single dimension-aware framework is one of the most interesting open problems left by this work.

Chapter 4 proposed several families of methods pursuing this goal, ranging from simple adaptations of standard convex techniques to new methods based on smooth convex interpolation conditions. Chapter 5 addressed theoretical properties and practical aspects allowing some of these methods to be implemented efficiently (notably managing an ever growing number of samples and warm starting).

The benchmarking results of Chapter 6 paint a nuanced picture. The LBM-based methods emerged as a promising family, achieving linear convergence rates across all problems considered and offering a favorable computational cost relative to their performance. The warm-start and dual-averaged variants both contribute meaningful improvements, and their combination yields a generally efficient algorithm.

Their main limitation is a sensitivity to dimension, which confines their practical interest to the low-dimensional regime this thesis specifically targets. ACCPM and its smooth variant proved to be robust and consistent methods, performing well across all benchmarks and problem conditionings. Notably, S-ACCPM consistently outperformed its non-smooth counterpart by a significant margin, suggesting that the improved cutting planes provided by smoothness have a particularly strong impact in the geometric framework underlying these methods. The Level methods delivered strong performance across the board, exhibiting what appears to be linear convergence on structured problems such as quadratics and Huber regression, and remaining competitive on harder problems. The Ellipsoid-based methods were not competitive: their performance degrades sharply with dimension and they suffer from numerical instability that persists even with dedicated remedies, making them impractical in the setting considered here.

A common thread across all methods is that the benefits of memory exploitation are most pronounced in the low-dimensional, limited-oracle regime that motivated this work. This is consistent with both the known lower bounds and the volume lemma : in the asymptotic regime where the number of iterations exceeds the dimension, the finite-dimensional lower bound ultimately dominates and no cutting-plane method can outperform OGM by more than a constant factor. The methods developed here are precisely designed for the complementary regime, where this bound has not yet kicked in and the accumulated oracle information can be leveraged more aggressively. The volume lemma also states that shrinking the localization set is less effective as the dimension grows : shrinking $\text{Vol}(\mathcal{X}_m)$ by a factor of ρ only improves the guarantee on the functional residual by a factor $\rho^{2/d}$, which means that this approach is inherently less effective in high dimension.

Several limitations of the present work deserve mention. The analysis is restricted to the deterministic exact-oracle setting, and the benchmark, while covering a representative range of classical smooth convex problems, is not exhaustive. The problems considered are also of relatively small dimension, which is by design but limits the conclusions that can be drawn about scalability. None of the proposed methods come with strong theoretical convergence guarantees tailored to the smooth convex finite-dimensional setting, which means the empirical observations remain difficult to interpret theoretically.

Several research directions naturally follow from the results and limitations of this thesis.

- A fundamental open problem is the derivation of a finite-dimensional lower bound for smooth convex optimization. Such a bound would characterize the best achievable convergence rate as a function of both the number of iterations and the ambient dimension, and would provide the theoretical foundation that is currently missing for the methods studied here.
- A closely related and ambitious direction would be to design algorithms that are dimension-aware in a principled way. The ideal method would achieve the standard oracle complexity lower bound when the number of iterations is small relative to the dimension, and transition to matching the finite-dimensional lower bound once enough information has been collected. Concretely, this means achieving the optimal rate when $k \leq (d-1)/2$ and adapting to the finite-dimensional regime beyond that threshold. Such a method would bridge the gap between the cutting-plane approach studied here and the dimension-uniform methods such as OGMM and a-PBM, and would require a much better theoretical understanding of both regimes and how to transition between them.

-
- The dual averaging method had strong empirical performance across all benchmarks and seems to leverage some second order information from the accumulated oracle history. A more thorough theoretical and empirical investigation of this method could lead to a better understanding of how it works and how to further improve it. In particular, it would be interesting to understand whether the observed convergence rates can be grounded theoretically in the smooth convex setting, and whether the method can be made more robust. Warm-starting could also see some improvement, notably in low curvature environments, where it performs poorly.
 - The strong empirical performance of S-ACCPM relative to its non-smooth counterpart warrants further investigation. The results suggest that smooth interpolation conditions interact particularly well with the geometric structure of ACCPM, and a more thorough theoretical and empirical study of this variant could lead to a better understanding of why this is the case and how to exploit it further.
 - The Level methods produced strong empirical results throughout the benchmarks, and a more systematic investigation of their behavior on harder or higher-dimensional problems, combined with a better understanding of the conditions under which numerical instability arises, could help determine whether these methods can be made more robust and whether their observed convergence behavior can be grounded theoretically in the smooth convex setting.

Bibliography

- [1] Moïse Blanchard, Junhui Zhang, and Patrick Jaillet. *Memory-Constrained Algorithms for Convex Optimization via Recursive Cutting-Planes*. arXiv.org. June 16, 2023.
- [2] S Boyd, L Vandenberghe, and J Skaf. “Analytic Center Cutting-Plane Method”. In: ().
- [3] Mateo Díaz and Benjamin Grimmer. *Optimal Convergence Rates for the Proximal Bundle Method*. May 1, 2023.
- [4] Yoel Drori and Marc Teboulle. *An optimal variant of Kelley’s cutting-plane method*. arXiv.org. Sept. 9, 2014.
- [5] Yoel Drori and Marc Teboulle. *Performance of first-order methods for smooth convex minimization: a novel approach*. June 14, 2012.
- [6] David Fersztand and Xu Andy Sun. *On the Acceleration of Proximal Bundle Methods*. Apr. 29, 2025.
- [7] Mihai I. Florea and Yurii Nesterov. *An optimal lower bound for smooth convex functions*. Apr. 29, 2024.
- [8] Hans Frenk, Joaquim Gromicho, and Shuzhong Zhang. “A deep cut ellipsoid algorithm for convex programming: Theory and applications”. In: *Mathematical Programming* 63 (Feb. 1, 1994), pp. 83–108.
- [9] P. Girardeau, V. Leclere, and A. B. Philpott. “On the Convergence of Decomposition Methods for Multistage Stochastic Convex Programs”. In: *Mathematics of Operations Research* 40.1 (Feb. 2015), pp. 130–145. ISSN: 0364-765X, 1526-5471.
- [10] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. *Beyond Minimax Optimality: A Subgame Perfect Gradient Method*. arXiv.org. Dec. 9, 2024.
- [11] Yassine Kamri, François Glineur, and Julien M. Hendrickx. “On the minimal and maximal smooth convex interpolating functions”. In: *Book of Abstracts, Benelux Workshop on Systems and Control*. Extended abstract. Belgium, 2023.
- [12] Sahar Karimi and Stephen Vavasis. *Nonlinear conjugate gradient for smooth convex functions*. Jan. 3, 2024.
- [13] Donghwan Kim and Jeffrey A. Fessler. “Generalizing the optimized gradient method for smooth convex minimization”. In: *SIAM Journal on Optimization* 28.2 (Jan. 2018), pp. 1920–1950. ISSN: 1052-6234, 1095-7189.
- [14] Donghwan Kim and Jeffrey A. Fessler. “On the convergence analysis of the optimized gradient method”. In: *Journal of Optimization Theory and Applications* 172.1 (Jan. 2017), pp. 187–205. ISSN: 0022-3239, 1573-2878.

-
- [15] Krzysztof C. Kiwiel. “Proximity control in bundle methods for convex nondifferentiable minimization”. In: *Mathematical Programming* 46.1 (Jan. 1, 1990), pp. 105–122. ISSN: 1436-4646.
 - [16] D. Lambert et al. “Finite Convex Integration”. In: *Journal of Convex Analysis* 11.1 (2004), pp. 131–146.
 - [17] Marko Mäkelä. “Survey of Bundle Methods for Nonsmooth Optimization”. In: *Optimization Methods and Software* 17.1 (Jan. 1, 2002). eprint: <https://doi.org/10.1080/10556780290027828>, pp. 1–29. ISSN: 1055-6788.
 - [18] Yu. Nesterov. “Complexity estimates of some cutting plane methods based on the analytic barrier”. In: *Mathematical Programming* 69.1 (July 1995), pp. 149–176. ISSN: 0025-5610, 1436-4646.
 - [19] Yurii Nesterov. *Lectures on Convex Optimization*. Vol. 137. Springer Optimization and Its Applications. Cham: Springer International Publishing, 2018.
 - [20] Yurii Nesterov and Mihai I. Florea. “Gradient Methods with Memory”. In: *Optimization Methods and Software* 37.3 (May 4, 2022), pp. 936–953. ISSN: 1055-6788, 1029-4937.
 - [21] “On Vaidya’s Volumetric Cutting Plane Method for Convex Programming”. In: *Mathematics of Operations Research* ().
 - [22] Fabian Pedregosa and Damien Scieur. “Average-Case Acceleration Through Spectral Density Estimation”. In: ().
 - [23] Boris T. Polyak. *Introduction to Optimization*. New York, NY, USA: Optimization Software, 1987. ISBN: 0911575146.
 - [24] Anne Rubbens et al. “Interpolation Constraints for Computing Worst-Case Bounds in Performance Estimation Problems”. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023 62nd IEEE Conference on Decision and Control (CDC). Singapore, Singapore: IEEE, Dec. 13, 2023, pp. 3015–3022. ISBN: 979-8-3503-0124-3.
 - [25] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. *Smooth Strongly Convex Interpolation and Exact Worst-case Performance of First-order Methods*. arXiv.org. Feb. 19, 2015.
 - [26] Blake Woodworth and Nathan Srebro. *Open Problem: The Oracle Complexity of Convex Optimization with Limited Memory*. July 1, 2019.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl