Université de Montréal

Acceleration and New Analysis of Convex Optimization Algorithms

par

Lewis Liu

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Discipline

July 11, 2021

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Acceleration and New Analysis of Convex Optimization Algorithms

présenté par

Lewis Liu

a été évalué par un jury composé des personnes suivantes :

Ioannis Mitliagkas
(président-rapporteur)

Simon Lacoste-Julien
(directeur de recherche)

Guillaume Rabusseau
(membre du jury)

Résumé

Ces dernières années ont vu une résurgence de l'algorithme de Frank-Wolfe (FW) (également connu sous le nom de méthodes de gradient conditionnel) dans l'optimisation clairsemée et les problèmes d'apprentissage automatique à grande échelle avec des objectifs convexes lisses. Par rapport aux méthodes de gradient projeté ou proximal, une telle méthode sans projection permet d'économiser le coût de calcul des projections orthogonales sur l'ensemble de contraintes. Parallèlement, FW propose également des solutions à structure clairsemée. Malgré ces propriétés prometteuses, FW ne bénéficie pas des taux de convergence optimaux obtenus par les méthodes accélérées basées sur la projection. Nous menons une enquête détaillée sur les essais récents pour accélérer FW dans différents contextes et soulignons où se situe la difficulté lorsque l'on vise des taux linéaires globaux en théorie. En outre, nous fournissons une direction prometteuse pour accélérer FW sur des ensembles fortement convexes en utilisant des techniques d'intervalle de dualité et une nouvelle notion de régularité.

D'autre part, l'algorithme FW est une covariante affine et bénéficie de taux de convergence accélérés lorsque l'ensemble de contraintes est fortement convexe. Cependant, ces résultats reposent sur des hypothèses dépendantes de la norme, entraînant généralement des bornes invariantes non affines, en contradiction avec la propriété de covariante affine de FW. Dans ce travail, nous introduisons de nouvelles hypothèses structurelles sur le problème (comme la régularité directionnelle) et dérivons une analyse affine invariante et indépendante de la norme de Frank-Wolfe. Sur la base de notre analyse, nous proposons une recherche par ligne affine invariante. Fait intéressant, nous montrons que les recherches en ligne classiques utilisant la régularité de la fonction objectif convergent étonnamment vers une taille de pas invariante affine, malgré l'utilisation de normes dépendantes de l'affine dans le calcul des tailles de pas. Cela indique que nous n'avons pas nécessairement besoin de connaître à l'avance la structure des ensembles pour profiter du taux accéléré affine-invariant.

Dans un autre axe de recherche, nous étudions les algorithmes au-delà des méthodes du premier ordre. Les techniques Quasi-Newton approchent le pas de Newton en estimant le Hessien en utilisant les équations dites sécantes. Certaines de ces méthodes calculent le Hessien en utilisant plusieurs équations sécantes mais produisent des mises à jour non symétriques. D'autres schémas quasi-Newton, tels que BFGS, imposent la symétrie mais ne

peuvent pas satisfaire plus d'une équation sécante. Nous proposons un nouveau type de mise à jour symétrique quasi-Newton utilisant plusieurs équations sécantes au sens des moindres carrés. Notre approche généralise et unifie la conception de mises à jour quasi-Newton et satisfait des garanties de robustesse prouvables.

Abstract

Recent years have witnessed a resurgence of the Frank-Wolfe (FW) algorithm, also known as conditional gradient methods, in sparse optimization and large-scale machine learning problems with smooth convex objectives. Compared to projected or proximal gradient methods, such projection-free method saves the computational cost of orthogonal projections onto the constraint set. Meanwhile, FW also gives solutions with sparse structure. Despite of these promising properties, FW does not enjoy the optimal convergence rates achieved by projection-based accelerated methods.

On the other hand, FW algorithm is affine-covariant, and enjoys accelerated convergence rates when the constraint set is strongly convex. However, these results rely on norm-dependent assumptions, usually incurring non-affine invariant bounds, in contradiction with FW's affine-covariant property. In this work, we introduce new structural assumptions on the problem (such as the directional smoothness) and derive an affine invariant, norm-independent analysis of Frank-Wolfe. Based on our analysis, we propose an affine invariant backtracking line-search. Interestingly, we show that typical backtracking line-search techniques using smoothness of the objective function surprisingly converge to an affine invariant stepsize, despite using affine-dependent norms in the computation of stepsizes. This indicates that we do not necessarily need to know the structure of sets in advance to enjoy the affine-invariant accelerated rate. Additionally, we provide a promising direction to accelerate FW over strongly convex sets using duality gap techniques and a new version of smoothness.

In another line of research, we study algorithms beyond first-order methods. Quasi-Newton techniques approximate the Newton step by estimating the Hessian using the so-called secant equations. Some of these methods compute the Hessian using several secant equations but produce non-symmetric updates. Other quasi-Newton schemes, such as BFGS, enforce symmetry but cannot satisfy more than one secant equation. We propose a new type of quasi-Newton symmetric update using several secant equations in a least-squares sense. Our approach generalizes and unifies the design of quasi-Newton updates and satisfies provable robustness guarantees.

Liste des mots-clés

Dégradé Conditionnel

Frank-Wolfe

Méthode Broyden-Fletcher-Goldfarb-Shanno

Formule de Davidon-Fletcher-Powell

List of keywords

Conditional Gradient

Frank-Wolfe

Broyden-Fletcher-Goldfarb-Shanno Method

Davidon-Fletcher-Powell Formula

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et des abréviations

FW	Frank-Wolfe
CG	Conditional Gradient
BFGS method	Broyden–Fletcher–Goldfarb–Shanno method
DFP formula	Davidon–Fletcher–Powell formula

Remerciements

I am greatly grateful for an amazing set of family, friends, colleagues, and supervisors without whom my work would not have been possible.

Thank you to Prof. Simon Lacoste-Julien and Dr. Damien Scieur for serving as outstanding advisors. Your direction, research mentorship, and guidance has allowed me to grow as a researcher and follow the path I am on today.

To my many friends, co-authors, co-workers, and colleagues (alphabetically): Reza Babanezhad, Pierre-Luc Bacon, Nicolas Boumal, Gauthier Gidel, Thomas Kerdreux, Alex Lamb, Nicolas Loizou, Songtao Lu, Ioannis Mitliagkas, Prakash Panangaden, Doina Precup, Guillaume Rabusseau, Sharan Vaswani, Zhaoran Wang, Zhuoran Yang, Yufeng Zhang, Tuo Zhao, Zhaocheng Zhu; Thank you for everything you do.

In addition, thanks to the rest of my friends and coworkers in MILA and my former colleagues and collaborators at Tencent AI Lab, Northwestern University, IBM research, etc. I am sure I missed some people I should not have forgotten. Thanks to all of you.

Introduction

In this thesis, we present improved algorithms and analysis for two essential classes of convex optimization algorithms, Frank-Wolfe and Quasi-Newton methods. We first recall basic concepts and results in optimization as well as in acceleration schemes.

0.1. Convex Optimization

Our goal is to design efficient algorithms in a d-dimensional space for solving

$$\min_{x \in \mathcal{C}} f(x) \tag{OPT}$$

where $x \in \mathcal{C} \subseteq \mathbb{R}^d$, $\mathcal{C}$ is the constraint set, and f is the objective function. To make such a minimization problem tractable, we study it under the common convexity hypothesis on (OPT). A convex optimization problem satisfies the following conditions,

(1) The set $\mathcal{C}$ is convex, i.e., $\forall p, q \in \mathcal{C}$ and $\lambda \in [0,1]$, it holds that $\lambda p + (1-\lambda)q \in \mathcal{C}$.

(2) The function f is convex, i.e., $\forall p, q \in \mathcal{C}$ and $\lambda \in [0,1]$, we have

$$\lambda f(p) + (1-\lambda) f(q) \geq f(\lambda p + (1-\lambda)q).$$

These are sufficient conditions to guarantee that (potentially inefficient) methods exist to solve the minimization problem (OPT). Furthermore, it is possible to design more efficient algorithms with better theoretical properties by imposing additional structural assumptions. Below are the most frequent assumptions that we use throughout this thesis.

Differentiability. We assume that f is differentiable.

Strong convexity. We call f a strongly convex function with constant μ if we have

$$f(x) \geq f(y) + \nabla f(y)^\top (x-y) + \frac{\mu}{2}\|x-y\|^2 \tag{0.1.1}$$

for all x,$y \in \mathbb{R}^d$.

Smoothness. We call f a smooth function with constant L if we have

$$f(s) \leq f(t) + \nabla f(t)^\top (s-t) + \frac{L}{2}\|x-y\|^2 \tag{0.1.2}$$

for all s,$t \in \mathbb{R}^d$.

We remark that these definitions hold for any norm $\|\cdot\|$. Without specifying, we refer to the Euclidean norm. In general, these structural assumptions upper and lower bound the function by quadratics. Furthermore, smoothness and strong convexity provide useful inequalities, which can be used to design efficient algorithms.

0.2. Affine Invariance

Given an affine change of coordinates $x = Ay$ where $A \in \mathbb{R}^{d\times d}$ is a nonsingular matrix, we can transform the original optimization problem defined in in (OPT) as follows,

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{C}, \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & \hat{f}(y) \\ \text{s.t.} & y \in \hat{\mathcal{C}}, \end{array} \tag{0.2.1}$$

where the variable $y \in \mathbb{R}^d$, and

$$\hat{f}(y) \triangleq f(Ay) \quad \text{and} \quad \hat{\mathcal{C}} \triangleq A^{-1}\mathcal{C}. \tag{0.2.2}$$

Note that both problems in (0.2.1) are equivalent, therefore they should have technically identical complexity bounds per se, unless A is pathologically ill-conditioned. Hence, we expect the analysis of optimization algorithms for convex problems to yield *affine invariant* rates. Such consideration is the starting point to study and to extend the following two popular types of algorithms in this thesis.

Frank-Wolfe Algorithm. The *affine covariance* nature of Frank-Wolfe (FW) Algorithm should imply an affine invariant rate from the analysis. However, the theoretical accelerated rate over strongly convex sets is affected by affine transformations. See Section 0.3 for details. We provide the definition of affine covariance below.

Definition 0.2.1 (Affine covariance)**.** An algorithm is affine covariant when its iterates (x_k) (resp. (y_k)) for problem (0.2.1) satisfy

$$x_k = Ay_k.$$

Quasi-Newton Methods. [57] developed the complexity analysis of Newton's method via the self-concordance argument, which produces affine invariant convergence rates and the iterates themselves are invariant. Meanwhile, for large-scale problems, Quasi-Newton methods are computationally efficient alternatives with Hessian approximation which keep the proximity to affine invariance properties. See Section 0.4 for details.

0.3. Frank-Wolfe Algorithm

Recent years have witnessed a promising resurgence of Frank-Wolfe algorithm (also known as conditional gradient (CG) methods [29]) in sparse optimization and large-scale machine

Algorithm 1 Frank-Wolfe Algorithm

Input: $x_0 \in \mathcal{C}$.

1: **for** $k = 0, 1, \ldots, K$ **do**
2: $\quad v_k \in \operatorname{argmax}_{v \in \mathcal{C}} \langle -\nabla f(x_k), v - x_k \rangle$ ▷ LMO
3: $\quad \gamma_k = \operatorname{argmin}_{\gamma \in [0,1]} f(x_k + \gamma(v_k - x_k))$ ▷ Line-search
4: $\quad x_{k+1} = (1 - \gamma_t)x_k + \gamma_k v_k$ ▷ Convex update
5: **end for**

learning problems, compared to projected or proximal gradient methods. Motivating applications span over structural SVMs [50], structured energy minimization [77], greedy particle optimization [30], among others. Despite of its nice properties, Frank-Wolfe suffers from a slower convergence rates than accelerated rates achieved by projection-based methods, i.e., Nesterov's accelerated gradient descent [60]. To fill this gap, we study possible extensions to accelerate Frank-Wolfe algorithm based on estimate sequences [59] and the duality-gap technique [23], with new upper and lower bounds for the duality gap as byproduct. In addition, we propose a new variant of Frank-Wolfe algorithm with adaptive stepsizes, which provides promising directions for global acceleration of Frank-Wolfe over smooth and strongly convex sets.

In detail, Frank-Wolfe algorithms [29] form a class of first-order methods solving constrained optimization problems such as

$$\min_{x \in \mathcal{C}} f(x). \tag{0.3.1}$$

The schemes in this class decompose non-linear constrained problems into a series of linear problems on the original constraint set, *i.e.* linear minimization oracles (LMO) indicated in Algorithm 1. They form a practical family of algorithms

Besides, with the appropriate line-search, the iterates of the FW are *affine covariant* under the affine transformation $y = Bx + b$ of problem (0.3.1),

$$\min_{y \in \tilde{\mathcal{C}} = B\mathcal{C} + b} \tilde{f}(y) \stackrel{\text{def}}{=} f(B^{-1}(y - b)), \quad B \text{ invertible.} \tag{0.3.2}$$

In other words, the behavior of Algorithm 1 is insensitive to affine transformations of the space. This means that, ideally, the theoretical rate for a affine covariant algorithm should be *affine invariant*.

The original Frank-Wolfe algorithm (Algorithm 1) generally enjoys a slow sublinear rate $\mathcal{O}(1/K)$ over general compact convex set and smooth convex functions [43]. In that setting, [14, 43] define a modulus of smoothness that leads to an affine invariant analysis of the Frank-Wolfe algorithm, matching with the affine covariant behavior of the algorithm.

Many works have then sought to find structural assumptions and algorithmic modifications that accelerate this sublinear rate of $\mathcal{O}(1/K)$. The strong convexity of the set (or more

generally uniform convexity, see [45]) is one of such structural assumptions which lead to various accelerated convergence rates. For example, linear convergence rates can be achieved when the unconstrained optimum is outside the constraint set [51, 22, 25, 66]; sublinear rates $\mathcal{O}(1/K^2)$ can be achieved when the function is also strongly convex but without restrictions on the position of the optimum [31]. However, there exists no affine invariant analysis for these accelerated regimes stemming from the strong convexity of the constraint set $\mathcal{C}$.

Recall that the smoothness of a function and the strong convexity of a set are defined as follows.

Definition 0.3.1. The function f is **smooth** over the set $\mathcal{C}$ w.r.t. the norm $\|\cdot\|$ if there exists a constant $L > 0$ such that, for any x ,$y \in \mathcal{C}$, we have

$$f(y) \leq f(x) + \langle \nabla f(x),\, y - x \rangle + \frac{L}{2}\|x - y\|^2. \tag{0.3.3}$$

Definition 0.3.2. A set $\mathcal{C}$ is **α-strongly convex** with respect to a norm $\|\cdot\|$ if, for any $(x,y) \in \mathcal{C}$, $\gamma \in [0,1]$ and $\|z\| \leq 1$, we have

$$\gamma x + (1-\gamma)y + \alpha\gamma(1-\gamma)\|x - y\|^2 z \in \mathcal{C}. \tag{0.3.4}$$

In the "non-affine invariant" analyses, structural assumptions like the L-smoothness (Definition 0.3.1) of f and the α-strong convexity of $\mathcal{C}$ (Definition 0.3.2) lead to accelerated convergence rate of the Frank-Wolfe algorithm, but are typically conditioned on parameters L,α and others, which depend on a particular choice of a norm. This is surprising given that the Frank-Wolfe algorithm (under appropriate line-search) does not depend on any norm choice.

Obtaining *practical* accelerated affine invariant rates is hard, as an affine invariant step size is required. Indeed, some adaptive stepsizes rely on theoretical affine invariant quantities which are in general not accessible. Therefore, by practical, we consider rates that can be achieved without a deep knowledge of the problem structure and constants.

For instance, scheduled stepsizes like $\gamma_k = \frac{2}{k+2}$ makes the Frank-Wolfe algorithm practically affine covariant, yet they do not capture the accelerated convergence regimes with an $\mathcal{O}(1/K^2)$ rate. Exact line-search guarantees a practically affine covariant algorithm while capturing accelerated convergence regimes but significantly increases the time to perform a single iteration. Finally, it is possible to use backtracking line-search such as [63]. Unfortunately, backtracking techniques rely on the choice of a specific norm, thus potentially breaking affine invariance of the algorithm.

This raises naturally the following questions:

Can we derive affine invariant rates for the Frank-Wolfe algorithm on strongly convex sets?

Can we design an affine invariant backtracking line-search for Frank-Wolfe algorithms?

This thesis provides a positive answer to these questions, by proposing the following contributions. **(1)** we conduct affine invariant analyses of the Frank-Wolfe Algorithm 1, when the function f is smooth w.r.t. to a specific distance function $\omega(\cdot)$ and the set $\mathcal{C}$ is strongly convex also w.r.t. $\omega(\cdot)$. We then introduce new structural assumptions extending the class of problems for which such accelerated regimes hold in the case of Frank-Wolfe, called *directionally smooth functions with direction* δ. From this definition, **(2)** we propose an affine invariant backtracking line-search for finding the optimal stepsize. Finally, **(3)** we show that existing backtracking line-search methods, which use a specific norm, converges surprisingly to the optimal norm-invariant, affine invariant stepsize, meaning that affine-dependent and affine invariant backtracking techniques perform similarly. As a byproduct, under the condition of strongly convex sets we provide a promising direction to accelerate FW using duality gap techniques, which is illustrated in Appendix **??**.

0.4. Quasi-Newton Methods

In another line of work, we consider second-order methods for unconstrained minimization of a smooth, possibly non-convex function $f\colon \mathbb{R}^d \to \mathbb{R}$. Despite a locally quadratic convergence rate, the well-known Newton method iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1} \nabla f(\mathbf{x}_k) \tag{0.4.1}$$

is not suitable for large-scale problems, in part because it requires solving a $d \times d$ linear system involving the Hessian at every iteration. To address this issue, quasi-Newton algorithms replace the update rule (0.4.1) by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1} \nabla f(\mathbf{x}_k) \quad \text{or} \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k \ \nabla f(\mathbf{x}_k), \tag{0.4.2}$$

where $\mathbf{B}_k \approx \nabla^2 f(\mathbf{x}_k)$ and $\mathbf{H}_k \approx \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}$ are approximations of the Hessian and its inverse (respectively) at $\mathbf{x}_k$. Choosing the right approximation has drawn considerable attention in the optimization literature, notably the DFP update [17], Broyden method [10], SR1 update [13] and the well-known BFGS method [11], [28], [32] [76]. In general, those

methods estimate a matrix $\mathbf{B}_k$ or $\mathbf{H}_k$ satisfying the *secant* equation

$$\begin{aligned} \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}) &= \mathbf{B}_k(\mathbf{x}_k - \mathbf{x}_{k-1}) \text{ or} \\ \mathbf{H}_k(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) &= \quad \mathbf{x}_k - \mathbf{x}_{k-1}, \end{aligned} \tag{0.4.3}$$

then perform the quasi-Newton step (0.4.2). It is also possible to satisfy *several* secant equations. For instance, the multisecant Type-I and Type-II Broyden methods [27] find a *non-symmetric* matrix $\mathbf{B}_k$ or $\mathbf{H}_k$ satisfying a block of secants: for a memory size m and for $i = k - m + 1 \dots k$,

$$\begin{aligned} \nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1}) &= \mathbf{B}_k[\mathbf{x}_i - \mathbf{x}_{i-1}] \text{ or} \\ \mathbf{H}_k[\nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1})] &= \quad \mathbf{x}_i - \mathbf{x}_{i-1}. \end{aligned}$$

By contrast, other methods like BFGS and DFP enforce the symmetry of the update, but they satisfy only *one* secant equation, in which case [65] showed their high dependence in the stepsize. Indeed, while BFGS and DFP enjoy an optimal convergence rate on quadratics using exact line-search [62], [65] showed that with a *unitary* stepsize, these updates converge particularly slowly on a simple quadratic function with just two variables. Moreover, it was also observed that BFGS updates are sensitive to gradient noise, and designing quasi-Newton methods for stochastic algorithms is still a challenge [12, 8, 7, 6].

Unfortunately, except for quadratic functions [71], it is usually impossible to find a symmetric matrix that satisfies more than one secant equation. [34] adopted Hessian-vector products instead of the secant equations. Moreover, line search has been shown to be computationally expensive. Finally, the stabilisation procedure for stochastic BFGS usually requires a growing batch size to reduce the gradient noise, making it unpractical in many applications.

In this thesis, we tackle those problems by proposing a symmetric multisecant update, that satisfies the secant equations in a least-squares sense. We show their optimality on quadratics *with unitary stepsize*, and prove their robustness to gradient noise, making them good candidates in the context of stochastic optimization.

Chapitre 1

Affine Invariant Analysis of Frank-Wolfe on Strongly Convex Sets

This paper was accepted at ICML 2021 in the main conference track. Its authors are: Thomas Kerdruex, Lewis Liu*[1], Simon Lacoste-Julien, Damien Scieur*.*

Contribution: Thomas, Damien, and myself led the project. Damien and I studied the properties of Frank-Wolfe over strongly convex sets, and I explore how the choice of norms affects the convergence rate of the algorithm, which originated the idea of affine invariant analysis. We three derive the affine invariant analysis of Frank-Wolfe together, where Thomas found the direction of adapting the existing affine invariant proof to the strongly convex set scenario. Further, I explore the motivation and examples for the necessity of the affine invariant analysis. Also, I prototyped the experiments and proposed the affine-invariant backtracking Frank-Wolfe. Damien and I provided explanation for the similar performance between the classical backtracking algorithm and the affine-invariant one. Simon provided ideas and directions for the project, and fixed essential technical issues in our proof.

In Section 1.1, we review some existing work on affine invariant analysis of Frank-Wolfe algorithms. In Section 1.2, we motivate the need for affine invariant analysis of Frank-Wolfe on strongly convex sets. In Section 1.3 and 1.4, we introduce the structural assumptions on the optimization problem that we will consider for analysing Frank-Wolfe. In Section 1.5 we detail our affine invariant analysis of Frank-Wolfe on strongly convex set. In Section 1.6 and 1.7 we provide a backtracking line-search that directly estimate the affine invariant quantities we developed and we explain how it relates with existing ones. We conclude in Section 1.8 with numerical experiments.

[1]* indicates equal contribution

1.1. Related Work on Affine Analysis of Frank-Wolfe

Other linear convergence rates of Frank-Wolfe algorithms exists with affine invariant analysis. For instance, corrective variants of Frank-Wolfe exhibit (affine invariant) linear convergence rates when the constraint set is a polytope [48, 49] and the objective function is (generally) strongly convex. See Table 1.1 for a review of all affine invariant analyses of Frank-Wolfe algorithms.

These affine invariant analyses emphasize that there is no specific choice of norm to be made in Frank-Wolfe algorithms as well as there is no need for affine pre-conditionners. Frank-Wolfe algorithms are arguably *free-of-choice* methods, *i.e.* little needs to be known on the optimization problem's structures to obtain the accelerated regimes. This is in line with recent works showing that the Frank-Wolfe methods exhibit accelerated adaptive behavior under a variety of structural constraints of (0.3.1) which depend on inaccessible parameters, *e.g.* Hölderian Error Bounds on f [46, 83, 67] or local uniform convexity of $\mathcal{C}$ [45].

Affine invariant analyses introduce constants seeking to characterize structural properties without a specific choice of norm. This has then been the basis for works extending the accelerated convergence analysis to non-smooth or non-strongly convex functions [64, 37], which then explore new structural assumptions on f.

Related Work	$\mathcal{C}$	Str. cvx. f	x^*	Algo	Stepsize	Rate
[14]	Simplex	✗	Any	FW	Scheduled	$\mathcal{O}(1/T)$
[43]	Convex	✗	Any	FW	Scheduled	$\mathcal{O}(1/T)$
[48]	Any	✓	Interior	FW	Exact ls	Linear
[49]	Polytope	✓	Any	Corr. FW	Exact ls	Linear
[37]	Polytope	✓	Any	FW	Exact ls	Linear
Our work	Strongly cvx	✗	$\nabla f(x^\star) \neq 0$	FW	Backtracking ls	Linear
	Strongly cvx	✓	Any	FW	Backtracking ls	$\mathcal{O}(1/T^2)$

Tableau 1.1. Existing *affine invariant* analysis of Frank-Wolfe for smooth convex functions under different schemes.

Strong convexity. The strong convexity assumption is to be taken broad sense. In [48, 49], the authors consider generalized strong convexity, an affine-invariant measure of strong convexity, while [37] consider strongly convex functions relative to a pair $(\mathcal{C}, \omega)$ where ω is a distance-like function. In our work, we not directly assume strong convexity, but the *directional smoothness* of the function (see later Definition 1.4.1), whose constant is bounded if various assumptions are satisfied for (0.3.1) (Theorem 1.4.4).

Stepsize. By *scheduled* stepsizes, we consider, for instance, the classical $\gamma_k = \frac{1}{k}$. We denote by *exact-line search* when the optimal stepsize depends on an unknown affine-invariant quantity, whose accessible upper-bounds are affine-dependent (thus breaking the affine invariance of FW).

1.2. "Affine-dependent" Analysis of FW

It is known that when the function is *smooth* (Definition 0.3.1), the set is *strongly-convex* (Definition 0.3.2) and the gradient is lower bounded $\|\nabla f(x)\| \geq c$ over the constraint set (i.e., the constraints are active), the Frank-Wolfe algorithm 1 converges linearly [51, 22, 25], at rate (with $h_k \triangleq f(x_k) - f_\star$)

$$h_k \leq \left(\max\left\{\frac{1}{2}, 1 - \frac{c\alpha}{2L}\right\}\right)^k h_0. \tag{1.2.1}$$

Note that assuming the gradient to be lower bounded means the constraints are tight, i.e., the solution of the unconstrained counterpart lies outside the set of constraints. However, the constants L, α, and c depend on the choice of the norm for the smoothness and the strong convexity. In contrast, the Frank-Wolfe algorithm and iterates do not depend on such a choice, due to its affine covariance. Therefore, the rate of Algorithm 1 should be affine invariant. Unfortunately, it is possible to show that the known theoretical analyses can be *arbitrarily* bad in the case where the constants L, c, α depend on "affine variant" norms.

Example 1.2.1. Consider the projection problem

$$\min_x f(x) \stackrel{\text{def}}{=} \tfrac{1}{2}\|x - \bar{x}\|^2 \quad \text{such that } \tfrac{1}{2}\|x\|^2 \leq 1.$$

In such case, we have that $L = 1$, $\alpha = \frac{1}{\sqrt{2}}$ and $c = 1 - \|\bar{x}\|$ (L, α and c are defined according to the ℓ_2 norm). However, if we transform the problem into $\min_y f(By)$, the new constants become

$$L = \sigma_{\max}(B),\ \alpha = \tfrac{\sigma_{\min}(B)}{\sqrt{2}\sigma_{\max}(B)},\ c = \sigma_{\max}(B)(1 - \|\bar{x}\|).$$

Comparing the rate (1.2.1) of the two problems, identical to the eyes of the FW algorithm, we have that

$$f(x_k) - f^\star \leq \left(1 - \tfrac{(1-\|\bar{x}\|)}{2\sqrt{2}}\right)^k \left(f(x_0) - f^\star\right),$$
$$f(By_k) - f^\star \leq \left(1 - \tfrac{(1-\|\bar{x}\|)}{2\sqrt{2}}\kappa^{-1}(B)\right)^k \left(f(x_0) - f^\star\right),$$

where $\kappa(B) = \frac{\sigma_{\max}(B)}{\sigma_{\min}(B)}$ is the condition number of B. This means we can artificially make a large theoretical upper bound on the rate of convergence by using an ill-conditioned transformation (i.e., $\kappa(B)$ large). However, the speed of convergence of FW iterates are *not affected* by any linear transformation (dues to their affine-covariance), therefore the upper bound will not be representative of the true rate of convergence of FW.

When the optimum is in the relative interior of any compact set $\mathcal{C}$, FW converges linearly when f is strongly convex [36, 48]. On the other hand, linear convergence on strongly convex sets does not require strong convexity of f when the solution of the unconstrained problem lies outside the set [22]. Our paper hence focuses on extending the analysis where the unconstrained optimum is outside the constrain set [22].

These two analysis cover most practical cases, but not the situation where the unconstrained optimum is close to the boundary of $\mathcal{C}$. A recent analysis on strongly convex sets of [31] is not restrictive w.r.t. the position of the unconstrained optimum but conservative (convergence rate of $\mathcal{O}(1/K^2)$). It is interesting as it not only deals with the (previously unknown) situation where the unconstrained optimum is on the boundary on $\mathcal{C}$, but also when it is arbitrarily close to it, leading to poorly conditioned linear convergence regimes. In Appendix B.4, we provide an affine invariant analysis of [31].

1.3. Smoothness and Strong Convexity w.r.t. General Distance Functions

The major limitation in the definition of smoothness of a function (Definition 0.3.1) and the strong convexity of a set (Definition 0.3.2) is the presence of the norm in their definition, whose constants may be dependent on affine transformation of the space (see Example 1.2.1). Technically, the notion of norm in the definition of smoothness and strong convexity of a function can be extended to the concept of distance-generating function, for instance using Bregman divergence [5, 53] or gauge functions [16].

Although is it classical to use different distance-generating functions ω (that satisfies Assumption 1.3.1 below) to characterize the smoothness of a function, we are not aware of such analysis for strongly convex sets. We believe that such analysis may exist, but for completeness we propose here an extension of the strong convexity of a set w.r.t. a distance function ω.

Assumption 1.3.1. The function $\omega(\cdot)$ satisfies

- $\omega(x) = 0 \quad \Leftrightarrow \quad x = 0$,
- **Positivity:** $\omega(x) \geq 0$,
- **Triangular Inequality:** $\omega(x + y) \leq \omega(x) + \omega(y)$
- **Positive homogeneity:** $\omega(\gamma x) = \gamma\omega(x)$, $\gamma \geq 0$,
- **Bounded asymmetry:** $\max_x \frac{\omega(x)}{\omega(-x)} \leq \kappa_\omega$.

Since $\omega(x)$ is convex by the triangle inequality, we define the dual distance

$$\omega_*(v) = \max_{x:\omega(x)\leq 1} \langle v,x \rangle. \tag{1.3.1}$$

Remark 1.3.2. Usually, extensions of smoothness of a function use Bregman divergences (see e.g. [53, 5]). However, the assumption that the distance-generating function is positively homogeneous is crucial in our analysis, which is unfortunately, not satisfied for most Bregman divergences.

A typical example satisfying such assumptions are gauge functions, also called *Minkowski functional*,

$$\omega_{\mathcal{Q}}(v) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\tau \geq 0} \tau \quad \text{subject to} \quad v \in \tau \mathcal{Q},$$

where $0 \in \text{int}\mathcal{Q}$. Such distance-generating function satisfies Assumption 1.3.1 if the set $\mathcal{Q}$ is convex and compact, and contains 0 in its interior. Moreover, gauge functions are affine invariant.

Usually, most works using gauge function assume that the set $\mathcal{Q}$ is *centrally symmetric* [16, 55], which add the assumption that

$$\omega(x) = \omega(-x).$$

In that case, the gauge function is a norm [Theorem 15.2.][68]. Removing symmetry extends non-trivially the definition of strongly convex sets w.r.t. the distance function ω. We now recall the definitions of smoothness and strong convexity of a function w.r.t. a distance function ω.

Definition 1.3.3. A function f is smooth (resp. strongly convex) w.r.t. the distance function ω if, for a constant L_ω (resp. μ_ω), the function satisfies

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_\omega}{2} \omega^2(y - x), \tag{1.3.2}$$

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu_\omega}{2} \omega^2(y - x). \tag{1.3.3}$$

Definition 1.3.4. A set $\mathcal{C}$ is α_ω-strongly convex w.r.t. ω if, for any $(x,y) \in \mathcal{C}$ and $\gamma \in [0,1]$, we have

$$z_\gamma + \alpha_\omega \gamma(1-\gamma) \frac{(1-\gamma)\omega^2(x-y) + \gamma\omega^2(y-x)}{2} z \in \mathcal{C},$$

where $z_\gamma = \gamma x + (1-\gamma)y$, for all z such that $\omega(z) \leq 1$.

This definition extends the one of strongly convex sets with a general distance function that may not be a norm, see for instance [31].

With Definition 1.3.4, the level sets of smooth and strongly convex functions are also strongly convex sets when the function ω is used. Such results appear for instance in [44] when ω is the ℓ_2 norm.

Lemma 1.3.5 (Strong Convexity of Sets)**.** Let f be a L-smooth and μ-strongly convex function w.r.t. ω. Then, the set

$$\mathcal{C} = \{x : f(x) - f_\star \leq R\}$$

is α-strongly convex w.r.t. ω, with $\alpha = \frac{\mu_\omega}{\kappa_\omega \sqrt{2 L_\omega R}}$.

We defer the proof in Appendix B.1. This result corresponds *exactly* to the one of [Theorem 12][44], when we use $\omega = \|\cdot\|_2$.

Scaling Inequality. All proofs of Frank-Wolfe methods on strongly convex sets leverage the same property. The *scaling inequality* (equivalent to strong convexity of $\mathcal{C}$ [Theorem 2.1.][33]) crucially relates the Frank-Wolfe gap with $\|x_t - v_t\|^2$, see *e.g.* [Lemma 2.1.][45]. We extend the scaling inequality to strongly convex sets with generic distance functions.

Lemma 1.3.6 (Distance Scaling Inequality)**.** Assume $\mathcal{C}$ is α_ω-strongly convex w.r.t. ω. Then for any $x \in \mathcal{C}$, $\phi \in \mathbb{R}^d \backslash \{0\}$, and $v_\phi \in \operatorname{argmax}_{v \in \mathcal{C}} \langle \phi, v \rangle$, we have $\phi \in N_{\mathcal{C}}(v_\phi)$ (normal cone) and

$$\langle \phi,\, v_\phi - x \rangle \geq \frac{\alpha_\omega}{2} \omega_* \Big(\phi \Big) \omega^2 (v_\phi - x). \tag{1.3.4}$$

In particular for any iterate x_k of Frank-Wolfe and its Frank-Wolfe vertex v_k (Line 2 in Algorithm 1), we have

$$\langle -\nabla f(x_k); v_k - x_k \rangle \geq \frac{\alpha_\omega}{2} \omega_* \Big(-\nabla f(x_k) \Big) \omega^2 (v_k - x_k).$$

Démonstration. We start with $v_\phi = \operatorname{argmax}_{v \in \mathcal{C}} \langle \phi;\ v \rangle$. Then, we use the definition of strong convexity of a set,

$$\gamma x + (1-\gamma) v_\phi + \alpha_\omega \gamma (1-\gamma) D_\gamma (x - v_\phi) z \in \mathcal{C} \quad \forall z : \omega(z) \leq 1.$$

where $D_\gamma(x-y) \stackrel{\text{def}}{=} \frac{\gamma \omega^2(x-y) + (1-\gamma)\omega^2(y-x)}{2}$. Then, by optimality of v_ϕ,

$$\langle \phi;\ v_\phi \rangle \geq \langle \phi;\ \gamma x + (1-\gamma) v_\phi + \alpha_\omega \gamma (1-\gamma) D_\gamma (x - v_\phi) z \rangle$$

After simplification,

$$\langle \phi;\ v_\phi - x \rangle \geq \alpha_\omega (1-\gamma) D_\gamma (x - v_\phi) \langle \phi;\ z \rangle$$

which holds in particular when $\phi = -\nabla f(x)$, $\gamma = 0$ and z being the argmax (see (1.3.1)). ■

1.4. Directional Smoothness

We separately introduced smoothness for functions, and strong convexity for sets w.r.t. a distance function ω. Analyses of Frank-Wolfe algorithm on strongly convex sets [51, 22, 25] show that, when f is convex and smooth, and the unconstrained minima of f are outside of $\mathcal{C}$, there is linear convergence.

We hence propose a novel condition that mingles the smoothness of f with the strong convexity of $\mathcal{C}$ when moving in a specific direction δ. We are interested in particular with the FW direction and we will see later that this assumption guarantees a linear convergence rate in this case. We call this condition the *directional smoothness*.

Definition 1.4.1. The function f is *directionally smooth* with direction function $\delta : \mathcal{C} \to \mathbb{R}^d$ if there exists a constant $\mathcal{L}_{f,\delta} > 0$ s.t. $\forall x \in \mathcal{C}$ and $h > 0$ with $x + h\delta(x) \in \mathcal{C}$,

$$f\Big(x + h\delta(x)\Big) \leq f(x) - h\langle -\nabla f(x), \delta(x)\rangle + \frac{\mathcal{L}_{f,\delta} h^2}{2}\langle -\nabla f(x), \delta(x)\rangle. \tag{1.4.1}$$

The rationale of Definition 1.4.1 is to replace the norm in the usual smoothness condition (Definition 0.3.1) by a scalar product between the *direction* and the negative gradient, in order to get an affine invariant quantity for the FW direction (see Proposition 1.4.3 below).

Assuming $\delta(x)$ is a descent direction, i.e., $\langle -\nabla f(x), \delta(x)\rangle > 0$, we can obtain a minimization algorithm for f, by minimizing (1.4.1) over h,

$$x_{k+1} = x_k + h_{\text{opt}}\delta(x_k), \quad h_{\text{opt}} = \min\{h_{\max}\ ;\ \mathcal{L}_{f,\delta}^{-1}\}.$$

Example 1.4.2. *(Gradient descent on smooth functions)* The gradient algorithm uses $\delta(x) = -\nabla f(x)$. In such case, the function is directionally smooth with constant L, and we obtain

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - h\|\nabla f(x)\|^2 + \tfrac{Lh^2}{2}\|\nabla f(x)\|^2 \\ &= f(x) + h\left(\tfrac{Lh}{2} - 1\right)\|\nabla f(x)\|^2. \end{aligned}$$

The best h is given by $h_{\text{opt}} = \frac{1}{L}$, which is also the optimal one [61].

The advantage of directional smoothness is its affine invariance in the case where $\delta(x)$ is the FW step.

Proposition 1.4.3 (Affine Invariance of $\mathcal{L}_{f,\delta}$)**.** If $\delta(x)$ is affine covariant (e.g. the FW direction $\delta(x) \triangleq v(x) - x$), then $\mathcal{L}_{f,\delta}$ in (1.4.1) is invariant to an affine transformation of the constraint set (proof in Appendix B.2.2).

The next theorem shows that, in the case of the FW algorithm, the directional smoothness constant is bounded if the function is smooth and the set is strongly convex for any distance function ω. We use this result later, to show that affine invariant backtracking line-search is equivalent to using the best distance function ω to define L_ω, c_ω and α_ω.

Theorem 1.4.4 (Directional Smoothness of FW)**.** Consider the function f, smooth w.r.t. the distance function ω, with constant L_ω, and the set $\mathcal{C}$, strongly convex with constant α_ω. Let $\delta(x) = x - v(x)$, $v(x)$ being the FW corner

$$v(x) \overset{\text{def}}{=} \operatorname*{argmin}_{v \in \mathcal{C}} \langle \nabla f(x), v\rangle.$$

Then, if $\omega_*(-\nabla f(x)) > c_\omega$ for all $x \in \mathcal{C}$ and some $c_\omega > 0$, the function $f(x)$ is directionally smooth w.r.t. to ω, with constant

$$\mathcal{L}_{f,\delta} \leq \frac{L_\omega}{c_\omega \alpha_\omega}. \tag{1.4.2}$$

Démonstration. See Appendix B.1.1 for the proof. ∎

1.5. Affine Invariant Linear Rates

With the directional smoothness constant $\mathcal{L}_{f,\delta}$ (affine invariant when δ is the FW direction), Theorem 1.5.1 shows an affine invariant linear rate of convergence of FW, generalizing existing convergence results of Frank-Wolfe on strongly convex sets [51, 22, 25].

Theorem 1.5.1 (Affine Invariant Linear Rates)**.** Assume f is a convex function and directionally smooth with direction function δ with constant $\mathcal{L}_{f,\delta}$. Then, the FW Algorithm 1 with stepsize

$$h_{\text{opt}} = \min\left\{1,\ \tfrac{1}{\mathcal{L}_{f,\delta}}\right\}, \quad \text{with } \delta = v(x) - x,$$

or with line-search, where $v(x)$ is the FW corner

$$v(x) = \operatorname*{argmin}_{v\in\mathcal{C}} \langle \nabla f(x), v\rangle,$$

converges linearly, at rate

$$f(x_k) - f_\star \leq \max\left\{\tfrac{1}{2},\, 1 - \tfrac{1}{2\mathcal{L}_{f,\delta}}\right\}\left(f(x_{k-1}) - f_\star\right).$$

Démonstration. We start with the directional smoothness assumption. For $0 < h < 1$,

$$f\Big(x_{k+1}\Big) \leq f(x_k) + \Big(h - \tfrac{\mathcal{L}_{f,\delta}h^2}{2}\Big)\left\langle \nabla f(x_k), \delta(x_k)\right\rangle$$

After minimization, we have two possibilities: $h_{\text{opt}} = \frac{1}{\mathcal{L}_{f,\delta}}$ or $h_{\text{opt}} = 1$. In the first case, we obtain

$$f\Big(x_{k+1}\Big) \leq f(x_k) + \tfrac{1}{2\mathcal{L}_{f,\delta}}\langle \nabla f(x_k), \delta(x_k)\rangle$$

Notice that the scalar product in the right-hand-side is the negative dual gap of Frank-Wolfe, that satisfies

$$\langle \nabla f(x_k), v(x) - x\rangle \leq -\left(f(x_k) - f_\star\right),$$

which gives the desired result. The second case follows immediately. ■

This provides an affine invariant analysis of the linear convergence regimes of FW on strongly convex sets.

The next proposition shows that the directional constant in Theorem 1.5.1 is bounded by (1.4.2) w.r.t. the distance function ω that gives the best ratio. This means that the Frank-Wolfe method acts like it optimizes the function in the best possible geometry, i.e., the geometry that gives the *best constants*.

Proposition 1.5.2 (Optimality of Dir. Smoothness)**.** Let Ω the set of function defined as

$$\Omega = \{\omega : \omega \text{ satisfies assumptions 1.3.1}\}.$$

Then, the directional smoothness constant follows

$$\mathcal{L}_{f,\delta} \leq \min_{\omega\in\Omega} \frac{L_\omega}{c_\omega \alpha_\omega},$$

where L_ω is the smoothness constant of the function f, α_ω the strong convexity of the set $\mathcal{C}$ and

$$c_\omega \leq \omega_*\Big(-\nabla f(x)\Big), \quad \forall x \in \mathcal{C}.$$

Démonstration. The proof is immediate by noticing that the FW algorithm do not use ω, therefore we can choose the best ω in Theorem 1.4.4. ■

To obtain a similar affine invariant analysis without restriction on the position of the optimum, *i.e.* the $\mathcal{O}(1/K^2)$ analysis in [31], one can define a similar property to the direction smoothness defined in Section 1.4. This new structural assumption additionally mingles together with the strong convexity of f. We provide details in Appendix B.4. We choose to focus the analysis for the linear convergence in the main text as it is the one most significant in practice.

1.6. Affine Invariant Backtracking

In previous sections, we proposed new constants to bound the rate of convergence of the Frank-Wolfe algorithm, which is affine invariant. The significant advantage of these constants is that, like FW, they are independent of any norm. However, the optimal step size of Frank-Wolfe needs the knowledge of these constants.

We propose in this section an affine invariant backtracking technique (Algorithm 2), based on directional smoothness. By construction, the backtracking technique finds automatically an estimate of the directional smoothness that satisfies

$$\mathcal{L}_k < 2\mathcal{L}_{f,\delta}, \quad k \geq \log_2\left(\tfrac{\mathcal{L}_0}{\mathcal{L}_{f,\delta}}\right).$$

1.7. Why Backtracking FW with norms is so efficient?

The stepsize strategy in Frank-Wolfe usually drives its practical efficiency. Sometimes, setting the stepsize optimally w.r.t. the theoretical analysis may be suboptimal in practice. Recently, [63] analyze the rate of the Frank-Wolfe algorithm for smooth function, using *backtracking line search*, described in Algorithm 6, Appendix B.3.

Algorithm 6 in Appendix B.3 is adaptive to the local smoothness constant, and ensures $L_{k+1} < 2L_f$, L_f being the smoothness constant of the function in the ℓ_2 norm. [63] observed that the estimate of the Lipchitz constant is often significantly smaller than the theoretical one; they wrote: *"We compared the average Lipschitz estimate L_t and the L, the gradient's Lipschitz constant. We found that across all datasets the former was more than an order of magnitude smaller, highlighting the need to use a local estimate of the Lipschitz constant to use a large stepsize."*

Algorithm 2 Affine invariant backtracking

Input: FW corner v_k, point x_k, directional smoothness estimate $\mathcal{L}_k$, function f.

1: $\mathcal{L} \leftarrow \mathcal{L}_k$. Define the optimal stepsize and next iterate in the function of the directional Lipchitz constant:

$$\gamma_\star(\mathcal{L}) \stackrel{\text{def}}{=} \min\{\tfrac{1}{\mathcal{L}},1\},$$
$$x(\mathcal{L}) \stackrel{\text{def}}{=} (1-\gamma_\star(\mathcal{L}))x_k + \gamma_\star(\mathcal{L})v_k.$$

2: Create the model of f between x_k and $x(\mathcal{L})$ based on equation (1.4.1),

$$m(\mathcal{L}) \stackrel{\text{def}}{=} f(x_k) + \gamma_\star(\mathcal{L})\,(1-\gamma_\star(\mathcal{L}))\,\langle \nabla f(x_k), v_k - x_k\rangle$$

3: Set the current estimate $\tilde{\mathcal{L}} \stackrel{\text{def}}{=} \frac{\mathcal{L}_k}{2}$.

4: **while** $f(x(\tilde{\mathcal{L}})) > m(\tilde{\mathcal{L}})$ (Sufficient decrease not met because $\tilde{\mathcal{L}}$ is too small) **do**

5: Double the estimate : $\tilde{\mathcal{L}} \leftarrow 2\cdot\tilde{\mathcal{L}}$.

6: **end while**

Output: Estimate $\mathcal{L}_{k+1} = \tilde{\mathcal{L}}$, iterate $x_{k+1} = x(\tilde{\mathcal{L}})$

With our analysis, however, we can explain why the estimate of the smoothness constant is much better than the theoretical one. The answer is simple:

Despite using a non-affine invariant bound, the stepsize resulting from the estimation of the Lipchitz constant via the backtracking line-search finds $\frac{1}{\mathcal{L}_{f,\delta}}$.

Proposition 1.7.1. Consider the "local Lipchitz constant" $L_{\text{loc}}(x)$ that satisfies (0.3.3) with $y = x + h\delta(x)$, i.e.,

$$\begin{aligned} f(x+h\delta(x)) \leq & f(x) + \nabla f(x)(x+h\delta(x)) \\ & + L_{\text{loc}}(x)\tfrac{h^2}{2}\|\delta(x)\|_2^2. \end{aligned}$$

Then, $L_{\text{loc}}(x)$ is bounded by

$$L_{\text{loc}}(x) \leq \mathcal{L}_{f,\delta}\frac{\langle -\nabla f(x), \delta(x)\rangle}{\|\delta(x)\|^2}.$$

Assuming $L_{\text{loc}}(x)$ "locally constant", the backtracking line-search finds $L_k < 2L_{\text{loc}}(x_k)$, and its stepsize $\gamma_\star$ satisfies

$$\min\left\{1, \frac{1}{2\mathcal{L}_{f,\delta}}\right\} \leq \gamma_\star.$$

Démonstration. See Appendix B.2.1 for the proof. ■

Therefore, the optimal stepsize from the backtracking line-search with the ℓ_2 norm is *exactly* the optimal affine invariant stepsize of our affine invariant analysis from Theorem 1.5.1.

In conclusion, *even if we use non-affine invariant norms* to find the smoothness constant, surprisingly, *the backtracking procedure finds the optimal, affine invariant stepsize.*

1.8. Illustrative Experiments

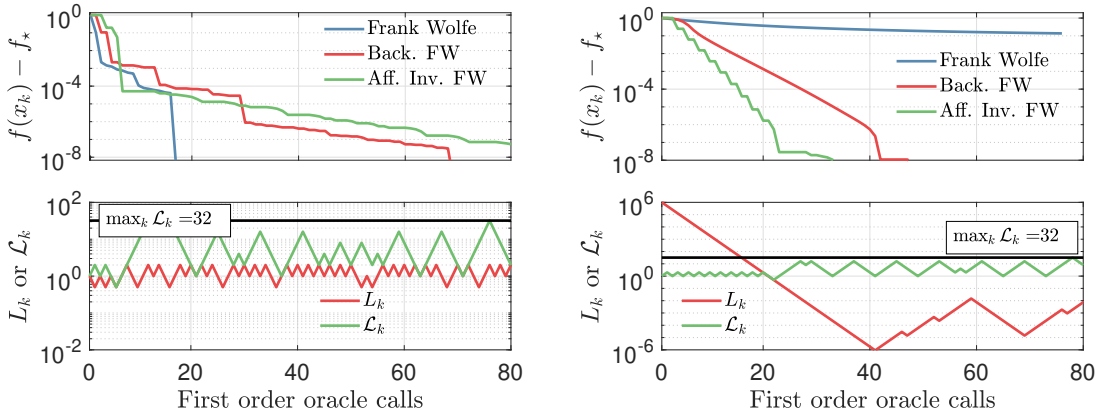


Fig. 1.1. Comparison of FW variants on the projection problem. Left: $B = I$, Right: $\kappa(B) = 10^6$. The top row is the gap $f_k - f^*$, and the bottom row corresponds to the estimation of the directional-smoothness constant $\mathcal{L}_k$ or the smoothness constant L_k, where the black line report the maximum value of $\mathcal{L}_k$. The reason why adaptive FW methods are slower in the left figure is because, in the worst case, the number of iterations to reach a certain precision can be up to four times larger than the worst-case bound on non-adaptive methods. We clearly see that the directional smoothness parameter $\mathcal{L}_{f,\delta}$ is affine invariant, as its estimate is $\max_k \mathcal{L}_k = 32$ in both scenarios.

Quadratic / logistic regression. We consider the constrained quadratic and logistic regression problem,

$$\min_{x \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} l(a_i^T x, y_i), \tag{1.8.1}$$

where l is the quadratic or the logistic loss. Here we adopt the ℓ_2-ball, defined as

$$\mathcal{C} = \{x : \|x\|_2 \leq R\}, \quad R > 0.$$

Specifically, we compare our affine invariant backtracking method in Algorithm 2 against the naive FW Algorithm 1 with stepsize $1/L$ [22] and back-tracking FW [63] on the Madelon dataset [39]. The results are shown in Figure 1.2. In detail, we set R such that the unconstrained optimum $\mathbf{x}^*$ satisfies $\|\mathbf{x}^*\|_2 = 1.1R$, and the initial iterate $\mathbf{x}_0 = \mathbf{0}$. As predicted by our theory, the affine invariant algorithm performs well at the beginning, but after a few iterations the two backtracking techniques behave similarly.

Projection. We solve here the projection problem described in Example 1.2.1, for two cases of B: One that corresponds to the original problem, i.e. $B = I$, the second one where B is an ill-conditioned matrix (with the condition number $\kappa(B) = 10^6$). The vector x_0 is random

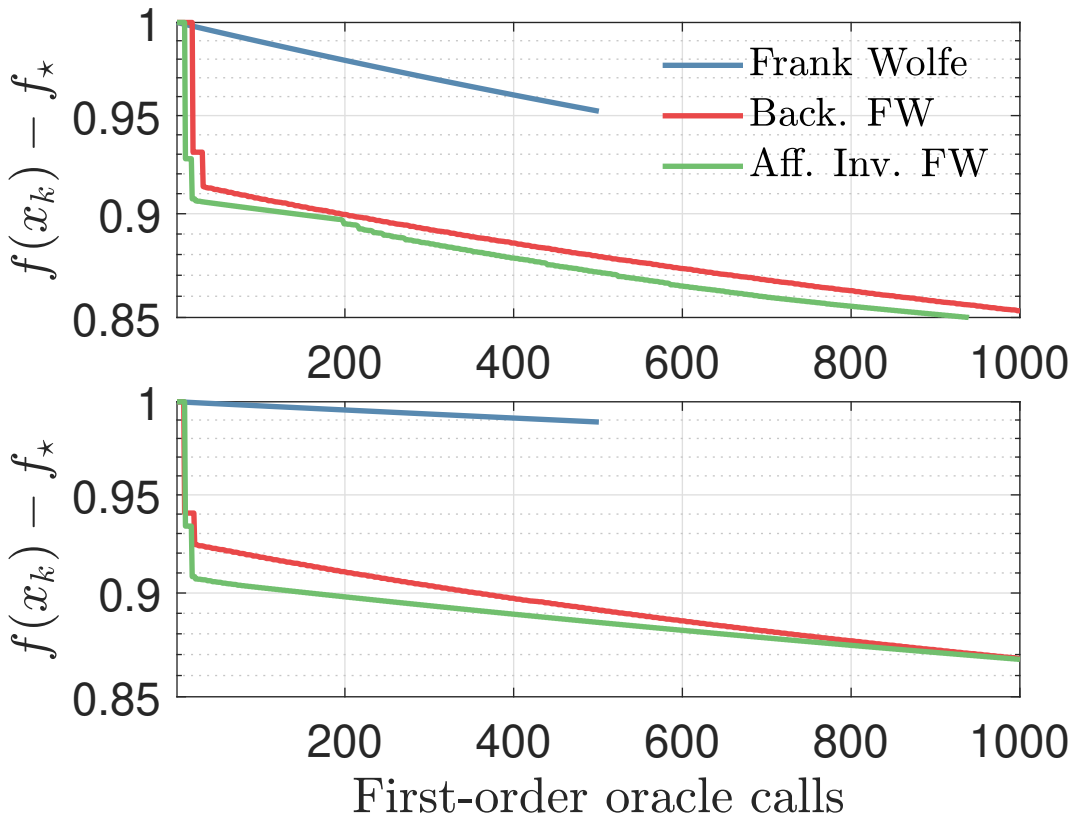


Fig. 1.2. Classification problem on Madelon dataset, with (*Top*) Quadratic loss and (*Bottom*) Logistic loss.

in the ℓ_2 ball, and $\bar{x} = \mathbf{1}_d \cdot (1.1/\sqrt{d})$. We report the results in Figure 1.1. We compare the standard FW algorithm with stepsize $1/L$, the FW with backtracking line-search (Algorithm 6) and FW with affine invariant backtracking technique (Algorithm 2). If the problem is well-conditioned ($\kappa(B) = 1$), all methods perform similarly. This is not the case, however, for the ill-conditioned setting, where the FW with no adaptive stepsize converges extremely slowly compared to the two other methods. We also see that the affine invariant backtracking converges quicker than the standard backtracking. This is explained by the fact that the latter takes a longer time to find the right constant L_k, while $\mathcal{L}_k$ remains untouched after an affine transformation.

1.9. Conclusion

In this chapter, our theoretical convergence results on strongly convex sets complete the series of accelerated affine invariant analyses of Frank-Wolfe algorithms. To obtain these, we formulate a new structural assumption with respect to general distance functions, the directional smoothness, which we will explore more systematically in future works. Also, we

present a new affine invariant backtracking line-search method based on directional smoothness. Within our framework of analysis, we provide a new explanation for the reasons behind the efficiency of the existing backtracking line search, and we show theoretically and experimentally they also find affine-invariant stepsizes.

Chapitre 2

Generalization of Quasi-Newton Methods

This paper was accepted at AISTATS 2021 in the main conference track. Its authors are: Damien Scieur[1], Lewis Liu*, Thomas Pumir, Nicolas Boumal.*

Contribution: Damien and myself led the project. Damien, Thomas, and Nicolas formulate the multi-secant update for quasi-Newton methods. Damien and I explored the theoretical motivation for considering such a new update scheme. In particular, I provided the proof of the robustness of our algorithms, wrote the technical section of the paper as well as some parts of other sections. Damien wrote the introduction to the methods, principles and the experiment illustration. Other authors also wrote a first version without the robustness results.

In Section 2.3 we list the desirable properties of quasi-Newton schemes, and end with a generic quasi-Newton update. The choice of its parameters, like the loss/regularization functions, the preconditioner, the number of secants or the initialization leads to different, existing methods but also to potentially new ones. Then, Section 2.4 proposes a novel quasi-Newton scheme (Algorithm 3) based on our framework, combining the ideas of DFP/BFGS and multisecant Broyden methods. This algorithm has the advantage of presenting a regularization term, which controls the stability of the update.

2.1. Notations

We use boldface small letters, like $\mathbf{x}$, to refer to vectors and boldface capital letters, like $\mathbf{A}$, for matrices. We use d to refer to the *dimension* of the problem, and m for the *memory* of the algorithm (we will see later that m is the number of secant equations). For a function $f\colon \mathbb{R}^d \to \mathbb{R}$, its gradient and Hessian at $\mathbf{x}$ are denoted by $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ respectively. Consistently with the notations in the literature, we use $\mathbf{H}$ to denote an approximation of the *inverse* of the Hessian, while we use $\mathbf{B}$ to denote an approximation of the Hessian. We denote the usual *Frobenius* norm as $\|\cdot\|$. Moreover, for any square matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and

[1]* indicates equal contribution

any positive definite matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$, we define the norm $\|\mathbf{A}\|_{\mathbf{W}}$ as

$$\|\mathbf{A}\|_{\mathbf{W}} = \|\mathbf{W}^{\frac{1}{2}}\mathbf{A}\mathbf{W}^{\frac{1}{2}}\|. \tag{2.1.1}$$

We often use the matrices $\mathbf{X}, \mathbf{G} \in \mathbb{R}^{d\times m+1}$, that concatenates the iterates and their gradients as follow,

$$\mathbf{X} = [\mathbf{x}_i, \ldots, \mathbf{x}_{i+m}], \quad \mathbf{G} = [\nabla f(\mathbf{x}_i), \ldots, \nabla f(\mathbf{x}_{i+m})].$$

Also, we define $\mathbf{C}$, and $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ as

$$\Delta\mathbf{X} = \mathbf{X}\mathbf{C}, \qquad \Delta\mathbf{G} = \mathbf{G}\mathbf{C},$$

where $\mathbf{C} \in \mathbb{R}^{m+1\times m}$ is a matrix of rank $m-1$ such that $\mathbf{1}_{m+1}^T\mathbf{C} = 0$, $\mathbf{1}_{m+1}$ being a vector of size $m+1$ full of ones. Typically, $\mathbf{C}$ is the column-difference matrix

$$\mathbf{C} = \begin{bmatrix} -1 & 0 & 0 & \ldots & \\ 1 & -1 & 0 & \ldots & \\ 0 & 1 & 1 & \ldots & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & 0 & 1 \end{bmatrix}.$$

2.2. Related work

The idea of updating an approximation of the Hessian or its inverse can be traced back to [17, 18] with the DFP update. Several updates, such as the Broyden method [10] or the BFGS method [11, 28, 32, 76] have been proposed since then. Notably, [20], [21] proposed to approximately invert the Hessian using a Conjugate Gradient method. Limited memory BFGS (L-BFGS) [52], where a limited number of vectors are stored for the approximation of the Hessian, has proven to be a powerful type of quasi-Newton method. The use of multisecant equations has also been used in a different context by [35] and [41], and their connection with Anderson Acceleration [2] was studied by [27]. This connection, combined with recent results on Anderson Acceleration [78, 79, 69, 74, 75], especially in the stochastic [73] and non-smooth [84] settings, may indicates that multisecant methods also enjoy some good theoretical properties. To scale up second-order methods, recent works focus on stochastic quasi-Newton methods. The use of stochastic quasi-Newton updates has been investigated by [72], [54], [56], [12] and [34], while approximating the Hessian through sampling methods has been proposed by [26], [82] and [1], among others.

We now present two popular quasi-Newton updates: the BFGS method, and the multisecant Broyden method. They will serve as a basis to motivate the needs of generalization of quasi-Newton updates.

Single secant DFP/BFGS updates The BFGS update finds a symmetric matrix $\mathbf{H}_k$ that satisfies the secant equation (0.4.3). Among the many possible solutions, it selects the one

closest to $\mathbf{H}_{k-1}$ in a weighted Frobenius norm (2.1.1), specifically,

$$\begin{aligned} \mathbf{H}_k = & \underset{\mathbf{H}=\mathbf{H}^T}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{H}_{k-1}\|_{\mathbf{W}} \\ & \text{s.t.} \quad \mathbf{H}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) = \mathbf{x}_k - \mathbf{x}_{k-1}. \end{aligned} \tag{2.2.1}$$

where $\mathbf{W}$ is *any* positive definite matrix such that $\mathbf{W}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) = \mathbf{x}_k - \mathbf{x}_{k-1}$ [62, §8.1] — a similar claim holds for the update formula of $\mathbf{B}_k$, known as DFP, whose update reads

$$\begin{aligned} \mathbf{B}_k = & \underset{\mathbf{B}=\mathbf{B}^T}{\operatorname{argmin}} \|\mathbf{B} - \mathbf{B}_{k-1}\|_{\mathbf{W}^{-1}} \\ & \text{s.t.} \quad \mathbf{B}(\mathbf{x}_k - \mathbf{x}_{k-1}) = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}). \end{aligned} \tag{2.2.2}$$

The matrix is then inverted using the Woodbury matrix identity. In the two update rules, the matrices $\mathbf{W}$ and $\mathbf{W}^{-1}$ are used implicitly, i.e., we do not need to form $\mathbf{W}$ to evaluate $\mathbf{H}_k$ nor $\mathbf{B}_k$.

Solving (2.2.1) repeatedly, BFGS builds a sequence $\mathbf{H}_1, \mathbf{H}_2, \ldots$ of matrices such that each $\mathbf{H}_k$ satisfies the kth secant equation. While it may satisfy the $k-1$ other secants approximately, the update rule offers no such guarantees. The same holds for the DFP update.

Multi-secant Broyden updates In the case of Broyden updates, we seek a matrix $\mathbf{B}$ for the type-I, or $\mathbf{H}$ for the type-II, that satisfies the secant equations only, without any restriction on the symmetry of the estimate. The update of the standard Broyden method reads, for $i = k-m, \ldots, k$,

$$\begin{aligned} \mathbf{B}_k = & \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{B} - \mathbf{B}_{k-m}\| \\ & \text{s.t.} \quad \mathbf{B}(\mathbf{x}_i - \mathbf{x}_{i-1}) = \nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1}), \\ \mathbf{H}_k = & \underset{\mathbf{H}}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{H}_{k-m}\| \\ & \text{s.t.} \quad \mathbf{H}(\nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1})) = \mathbf{x}_i - \mathbf{x}_{i-1}. \end{aligned} \tag{2.2.3}$$

As for the DFP update, the matrix $\mathbf{B}_k$ can also be inverted cheaply. In [27], the authors show how to extend this update to the case where we want to satisfy more than one secant equation. However, its solution is generally not symmetric.

2.2.1. Contributions

Quasi-Newton methods approximate the Hessian. The previous section shows they do this in very different ways that seem incompatible given the work of [71]. Despite their differences, they share similarities, such as the idea of secant equations. This leads to the following questions:

Algorithm 3 Type-I Symmetric Multisecant step (See Appendix C.1 for the type-II version)

Input: Function f and gradient ∇f, initial approximation of the Hessian $\mathbf{B}_{\text{ref}}$, maximum memory m (can be ∞), relative regularization parameter $\bar{\lambda}$.

1: Compute $g_0 = \nabla f(x_0)$ and perform the initial step

$$\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{B}_0^{-1}\boldsymbol{g}_0$$

2: **for** $t = 1,2,\ldots$ **do**

3: Form the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ (see Section 2.1) using the m last pairs $(\mathbf{x}_i, \nabla f(\mathbf{x}_i))$.

4: Compute the quasi-Newton direction $\mathbf{d}$ as

$$\mathbf{d}_t = -\mathbf{Z}_\star^{-1} g_t,$$

see (Inv-RSP) with $\mathbf{A} = \Delta\mathbf{X}$, $\mathbf{D} = \Delta\mathbf{G}$, $\mathbf{Z}_{\text{ref}} = \mathbf{B}_{\text{ref}}$, $\lambda = \bar{\lambda}\|\mathbf{A}\|$.

5: Perform an approximate-line search

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h_t\mathbf{d}_t, \quad h_t \approx \operatorname*{argmin}_h f\Big(\mathbf{x}_t + h_t\mathbf{d}_t\Big).$$

6: **end for**

- *Is it possible to design a generalized framework for quasi-Newton updates encompassing Broyden's, DFP and BFGS schemes?*
- *Can Symmetric and Multisecant techniques be combined into a single update?*

In this chapter, we propose a positive answer to these questions trough the following contributions.

- We propose a general framework that models and generalizes previous quasi-Newton updates.
- We derive new quasi-Newton update rules (Algorithm 3), which are symmetric and take into account *several secant equations*. The bottleneck is an (economic size) Singular Value Decomposition (SVD), whose complexity is linear in the dimension of the problem, therefore comparable to other quasi-Newton methods.
- We show the optimality of the convergence rate of any multisecant quasi-Newton update built using our framework, on quadratic functions *without line search*. This improves over the BFGS and DFP updates as they are inefficient with unitary step size on quadratics [65], and suboptimal if exact line-search is not used.
- We introduce novel *robust updates*, that provably reduce the sensitivity to the noise of our quasi-Newton schemes. This robustness property is a direct consequence of considering several secant equations at once.

2.3. Generalization of Quasi-Newton

We have seen in the previous section two different quasi-Newton (qN) updates: one that focuses on the *symmetry* of the estimate, the other on the number of satisfied *secant equations*. In this section, we propose a unified framework to design existing and new qN schemes.

2.3.1. Generalized (Multi-)Secant Equations

The central part of qN methods is the secant equation. The idea follows from the linearization of the gradient of the objective function. Indeed, consider the function $f(\mathbf{x})$, assumed to be smooth, strongly convex and twice differentiable. The linearization of its gradient around the minimum $\mathbf{x}_\star$ satisfies

$$\nabla f(\mathbf{x}) \approx \underbrace{\nabla f(\mathbf{x}_\star)}_{=0} + \nabla^2 f(\mathbf{x}_\star)(\mathbf{x} - \mathbf{x}_\star). \tag{2.3.1}$$

After a "Newton step", we get

$$\mathbf{x} - [\nabla^2 f(\mathbf{x}^\star)]^{-1} \nabla f(\mathbf{x}) \approx \mathbf{x}_\star.$$

Unfortunately, we do not have access to the matrix $\nabla^2 f(\mathbf{x}^\star)$ as we do not know $\mathbf{x}_\star$. Moreover, solving the linear system $[\nabla^2 f(\mathbf{x}^\star)]^{-1} \nabla f(\mathbf{x})$ may be costly when d is large.

To overcome such issues, consider a sequence $\{\mathbf{x}_0, \ldots, \mathbf{x}_m\}$ of points at which we have computed the gradients. Then, (2.3.1) can be stated as

$$\mathbf{G} = \nabla^2 f(\mathbf{x}_\star)(\mathbf{X} - \mathbf{X}_\star),$$

where $\mathbf{X}_\star = \mathbf{x}_\star \mathbf{1}_{m+1}^T$, i.e., the matrix concatenating $m+1$ copies of the vector $\mathbf{x}_\star$. Matrices $\mathbf{X}$ and $\mathbf{G}$ are defined in Section 2.1.

Ideally, the estimate $\mathbf{B}$ of the Hessian, or the estimate of its inverse $\mathbf{H}$, has to satisfy the condition

$$\mathbf{G} = \mathbf{B}(\mathbf{X} - \mathbf{X}_\star) \quad \text{or} \quad \mathbf{H}\mathbf{G} = (\mathbf{X} - \mathbf{X}_\star).$$

However, the dependency on $\mathbf{x}_\star$ makes the problem of estimating $\mathbf{B}$ or $\mathbf{H}$ intractable. To remove this problematic dependency, consider a matrix $\mathbf{C} \in \mathbb{R}^{m+1 \times m}$ of rank m such that $\mathbf{1}_{m+1}^T \mathbf{C} = 0$ (see Section 2.1 for an example). After multiplying by $\mathbf{x}_\star$ on the right, we simplify $\mathbf{X}_\star \mathbf{C} = 0$ and we obtain the *multisecant equations*

$$\Delta\mathbf{G} = \mathbf{B}\Delta\mathbf{X}, \quad \text{or} \quad \mathbf{H}\Delta\mathbf{G} = \Delta\mathbf{X}, \tag{2.3.2}$$

where $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are defined in Section 2.1. In the specific case where we have only one secant equation, (2.3.2) corresponds exactly to the standard secant equation in (2.2.1). In the case where $\mathbf{C}$ is the column-difference operator, we obtain the multisecant equations usually used in multisecant Broyden methods.

2.3.2. Regularization and Constraints

The matrices $\mathbf{B}$ (Broyden Type-I and DFP updates) and $\mathbf{H}$ (Broyden Type-II and BFGS) are selected so as to minimize the distances w.r.t. the reference matrices, called $\mathbf{B}_{\text{ref}}$ and $\mathbf{H}_{\text{ref}}$ respectively, as shown in (2.2.3). In the case where there is only a sequence of single secant

equations, the reference matrix is taken as being the previous estimate, with an arbitrary initialization. In the case of a multisecant update, the reference matrix is arbitrary. Moreover, in the case of DFP and BFGS, we have in addition a *symmetry* constraint, restraining even more the search space for the estimate of the Hessian. For simplicity, we will consider only the type-I update here, i.e., the estimate $\mathbf{B}$. The formulation for estimate $\mathbf{H}$ can be easily derived by swapping $\Delta\mathbf{G}$ and $\Delta\mathbf{X}$.

The intuition behind the regularization term is due to the number of degrees of freedom in the problem. The secant equation $\mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}$ defines the behavior of the operator $\mathbf{B}$, mapping from $\text{span}\{\Delta\mathbf{X}\}$ to $\text{span}\{\Delta\mathbf{G}\}$. However, the dimension of these two spans is as most $m < d$. This means we have to define the behavior of $\mathbf{B}$ *outside* of $\text{span}\{\Delta\mathbf{X}\}$ and $\text{span}\{\Delta\mathbf{G}\}$, i.e., from $\text{span}\{\Delta\mathbf{X}\}^\perp$ to $\text{span}\{\Delta\mathbf{G}\}^\perp$.

Since $\mathbf{B}$ outside the span is not driven by the secant equations, we have to define an operator $\mathbf{B}_{\text{ref}}$, defining the default behavior of $\mathbf{B}$ outside the span of secant equations. This means that, in the case where $\mathbf{B}$ satisfies exactly the secant equations, then $\mathbf{B}$ reads

$$\mathbf{B} = [\Delta\mathbf{G}\Delta\mathbf{X}^\dagger] + \Theta(\mathbf{I} - \mathbf{P}),$$

where $\mathbf{P}$ is the projector to the span of $\Delta\mathbf{X}$, $\Delta\mathbf{X}^\dagger$ is a pseudo-inverse of $\Delta\mathbf{X}$, and Θ depends on $\mathbf{B}_{\text{ref}}$ and constraints (different Θ lead to different qN updates). This way, $\mathbf{B}$ satisfies the secant equation, since multiplying $\mathbf{B}$ by $\Delta\mathbf{X}$ gives $\Delta\mathbf{G}$, since

$$\mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}\Delta\mathbf{X}^\dagger\Delta\mathbf{X} + \Theta(\mathbf{I} - \mathbf{P})\Delta\mathbf{X}.$$

We have $\mathbf{P}\Delta\mathbf{X} = \Delta\mathbf{X}$, thus $(\mathbf{I}-\mathbf{P})\Delta\mathbf{X} = 0$ (by construction of $\mathbf{P}$). Moreover, $\Delta\mathbf{G}\Delta\mathbf{X}^\dagger\Delta\mathbf{X} = \Delta\mathbf{G}$ by definition of the pseudo-inverse.

The way $\mathbf{B}$ behaves outside the span is thus driven by Θ, which depends on the regularization, the initialization $\mathbf{B}_{\text{ref}}$ and the constraints. To make a parallel with machine learning problems, Θ can be seen as the "generalization" (or "out-of-sample") term. We give example choices for Θ in Appendix C.5.6.

Consider the regularisation function $\mathcal{R}(\cdot,\mathbf{B}_{\text{ref}})$, assumed to be strictly-convex, whose minimum is attained at $\mathbf{B}_{\text{ref}}$, and the convex constraint set $\mathcal{C}$. We can write the qN update estimation problem as

$$\min_{\mathbf{B}\in\mathcal{C}} \mathcal{R}(\mathbf{B},\mathbf{B}_{\text{ref}}) \quad \text{subject to } \mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}. \tag{2.3.3}$$

This approach generalizes the way we define qN updates. Indeed, for instance, we recover DFP by setting $\mathcal{R} = \|\mathbf{B} - \mathbf{B}_{\text{ref}}\|_{\mathbf{W}^{-1}}$, $\mathcal{C} = \mathbb{S}^{d\times d}$ (the set of symmetric matrices), $m = 1$ and $\mathbf{B}_{\text{ref}} = \mathbf{B}_{k-1}$ in (2.3.3). We also recover the Type-I Broyden method by setting $\mathcal{R} = \|\mathbf{B}-\mathbf{B}_{\text{ref}}\|$ and $\mathcal{C} = \mathbb{R}^{d\times d}$.

2.3.3. Generalized Quasi-Newton Update

A natural extension, given the updates of DFP/BFGS and multisecant Broyden, would be the symmetric multi-secant update. This update would read, for an arbitrary regularization function,

$$\min_{\mathbf{B}\in\mathbb{S}^{d\times d}} \mathcal{R}(\mathbf{B},\mathbf{B}_{\text{ref}}) \quad \text{subject to } \mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}.$$

In the case where $m > 1$, this multisecant technique seems promising as it combines the advantages of multisecant Broyden and symmetric updates.

Assuming $\Delta\mathbf{X}, \Delta\mathbf{G}$ have full column rank, these equations always have a solution $\mathbf{B}$. However, there exists a *symmetric* solution *if and only if* $\Delta\mathbf{X}^T\Delta\mathbf{G}$ is symmetric [71, 40].

When $\Delta\mathbf{X}^T\Delta\mathbf{G}$ is symmetric, [71] derived a multisecant BFGS update rule. This assumption indeed holds for quadratic objectives, but not for general objective functions when $m \geq 2$, that is, when we consider more than one secant condition [71, Example 3.1]. Hence, a naive extension of symmetric quasi-Newton update leads to infeasible problems.

To tackle the problem of infeasible updates, we can relax the constraint on the secant equations by a *loss function* $\mathcal{L}(\cdot, \Delta\mathbf{X}, \Delta\mathbf{G})$. We finally end up with the *generalized (type-I and type-II) qN update*

$$\mathbf{B}_k = \lim_{\lambda\to 0} \operatorname*{argmin}_{\mathbf{B}\in\mathcal{C}} \mathcal{L}(\mathbf{B},\Delta\mathbf{X},\Delta\mathbf{G}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{B},\mathbf{B}_{\text{ref}}) \tag{GQN-I}$$

$$\mathbf{H}_k = \lim_{\lambda\to 0} \operatorname*{argmin}_{\mathbf{H}\in\mathcal{C}} \mathcal{L}(\mathbf{H},\Delta\mathbf{G},\Delta\mathbf{X}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{H},\mathbf{H}_{\text{ref}}) \tag{GQN-II}$$

where we assume that $\mathcal{L}$ and $\mathcal{R}$ are strictly convex, and sufficiently simple to have an explicit formula for $\mathbf{H}_k$. The limits here simply state that we first minimize the loss function, then with the remaining degrees of freedom we minimize the regularization term. In the case where the update (2.3.3) is feasible, then (GQN-I)/(GQN-II) and (2.3.3) are equivalent.

2.3.4. Preconditioning

As shown for instance in DFP and BFGS, it is common to use a preconditioner to reduce the dependence of the update to the units of the Hessian. We give here the example for type-II update. The type-I follows immediately by considering $\mathbf{W}^{-1}$ instead of $\mathbf{W}$.

The idea of preconditioning is, instead of considering $\mathbf{H}$, to set

$$\mathbf{M} = \mathbf{W}^{(1-\alpha)}\mathbf{H}\mathbf{W}^{\alpha},$$

where $\mathbf{W}$ ideally has the same units as the *Hessian* of the function f. For example, in BFGS, $\mathbf{W}$ is *any* matrix such that $\mathbf{W}\Delta\mathbf{X} = \Delta\mathbf{G}$, which always exists in the case where $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are vectors. Ideally, the preconditioner cancels the units in the update rules, i.e., $\mathbf{W}$ has to have the same units as the Hessian.

In the case where we consider a preconditioner,

$$\mathbf{M}\mathbf{W}^{-\alpha}\Delta\mathbf{X} = \mathbf{W}^{1-\alpha}\Delta\mathbf{G}, \;\; \mathbf{M}_{\text{ref}} = \mathbf{W}^{\alpha-1}\mathbf{H}_{\text{ref}}\mathbf{W}^{-\alpha}.$$

We now have the *type-II Preconditioned Generalized Quasi-Newton* update

$$\underset{\mathbf{M}\in\tilde{\mathcal{C}}}{\text{argmin}} \;\; \mathcal{L}(\mathbf{M},\mathbf{W}^{-\alpha}\Delta\mathbf{X},\mathbf{W}^{(1-\alpha)}\Delta\mathbf{G}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{M},\mathbf{M}_{\text{ref}}) \tag{PGQN-II}$$

where $\tilde{\mathcal{C}} = \mathbf{W}^{(1-\alpha)}\mathcal{C}\mathbf{W}^{\alpha}$, i.e., the image of the constraint after application of the preconditioner. To retrieve the update $\mathbf{H}$, it suffices to solve

$$\mathbf{H} = \mathbf{W}^{-(1-\alpha)}\mathbf{M}\mathbf{W}^{-\alpha}.$$

2.3.5. Rate of Convergence on Quadratics

Our theorem below shows that generalized qN methods (GQN-I) and (GQN-II) are optimal on quadratics under mild assumptions, in the sense that their performance is comparable to conjugate gradients.

Theorem 2.3.1. Consider *any* multisecant quasi-Newton method (GQN-II) with unitary step-size and $m = \infty$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k\nabla f(\mathbf{x}_k) \tag{2.3.4}$$

where f is the quadratic form $(\mathbf{x}-\mathbf{x}_\star)^T\frac{\mathbf{Q}}{2}(\mathbf{x}-\mathbf{x}_\star)$ for some $Q \succ 0$, and $\mathbf{H}$ satisfies exactly the secant equations. If the update (2.3.4) is a *preconditioned first-order method*, i.e., there exists a symmetric positive definite matrix $\tilde{\mathbf{H}}$ independent of k such that

$$\mathbf{x}_{k+1} \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\nabla f(\mathbf{x}_0),\dots,\nabla f(\mathbf{x}_k)\}$$

then $\mathbf{x}_k = \mathbf{x}_\star$ if $k \geq d+1$; for smaller k the method satisfies the rate

$$\|\nabla f(\mathbf{x}_k)\| \leq \mathcal{O}\left(\tfrac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k \|\nabla f(\mathbf{x}_0)\|,$$

Where κ is the inverse of the condition number of $\tilde{\mathbf{H}}\mathbf{Q}$.

The proof can be found in Appendix C.5. Notice that, for instance, the multisecant Broyden updates (2.2.3) or the multisecant BFGS update [71] satisfies the assumptions of Theorem 2.3.1 if $\mathbf{B}_{\text{ref}}$ or $\mathbf{H}_{\text{ref}}$ are symmetric positive definite matrices (see Appendix C.5.6). For all these methods, we have $\tilde{\mathbf{H}} = \mathbf{H}_{\text{ref}}$ (or $\mathbf{B}_{\text{ref}}^{-1}$). This indicates that the initialization is crucial, since a good initial approximation of $\mathbf{Q}^{-1}$ drastically reduces the condition number κ.

We have now a generic form of qN update, but it raises some important questions. Which practical losses and regularization functions should we use, and what happens if λ does not go to zero? The next section addresses the first point by giving an example that extends

(limited memory) DFP and multi-secant Broyden methods. Then, we analyse the robustness of the method when λ is non-zero.

2.4. Robust Symmetric Multisecant Updates

We now extend the BFGS and multisecant Broyden method into the type-II Symmetric Multisecant Update (2.4.1) below, solving the problem (PGQN-II) in the special case where the loss and the regularization are Frobenius norms. For simplicity, we do not consider any preconditioner here. The method reads

$$\mathbf{H}_k = \operatorname*{argmin}_{\mathbf{H}=\mathbf{H}^T} \|\mathbf{H}\Delta\mathbf{X} - \Delta\mathbf{G}\|_F^2 + \frac{\lambda}{2}\|\mathbf{H} - \mathbf{H}_{\text{ref}}\|^2 \tag{2.4.1}$$

and its type-I counterpart is $\mathbf{B}_k^{-1}$, where

$$\mathbf{B}_k = \operatorname*{argmin}_{\mathbf{B}=\mathbf{B}^T} \|\mathbf{B}\Delta\mathbf{G} - \Delta\mathbf{X}\|_F^2 + \frac{\lambda}{2}\|\mathbf{B} - \mathbf{B}_{\text{ref}}\|^2 \tag{2.4.2}$$

Explicit Formula. We now solve problem (2.4.1) efficiently. This is an extension of the *symmetric Procrusted problem* from [42]. Indeed, [42] solves the problem

$$\min_{\mathbf{Z}=\mathbf{Z}^T} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|,$$

where $\mathbf{A}$ and $\mathbf{D}$ are $\mathbb{R}^{d\times m}$ matrices, where $m > d$. In our case, we have $m \ll d$, and an extra regularization term, that makes the update formula more complicated. Fortunately, the matrix-vector multiplication $\mathbf{Z}\boldsymbol{v}$ can still be done efficiently even in our case, the bottleneck being the computation of the SVD of a thin matrix. The next theorem details the explicit formula to compute $\mathbf{M}_k$ (and its inverse if one wants to use a type-I method).

Theorem 2.4.1. Consider the Regularized Symmetric Procrustes (RSP) problem

$$\mathbf{Z}_\star = \operatorname*{argmin}_{\mathbf{Z}=\mathbf{Z}^T} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|^2, \tag{RSP}$$

where $\mathbf{Z}_{\text{ref}}$ is symmetric (otherwise, take the symmetric part of $\mathbf{Z}_{\text{ref}}$), $\mathbf{Z}, \mathbf{Z}_{\text{ref}} \in \mathbb{R}^{d\times d}$, and $\mathbf{A}, \mathbf{D} \in \mathbb{R}^{d\times m}$, $m \leq d$. Then, the solution $\mathbf{Z}_\star$ is given by

$$\mathbf{Z}_\star = \mathbf{V}_1\mathbf{Z}_1\mathbf{V}_1^T + \mathbf{V}_1\mathbf{Z}_2 + \mathbf{Z}_2^T\mathbf{V}_1^T + (\mathbf{I} - \mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I} - \mathbf{P}) \tag{Sol-RSP}$$

where

$$
\begin{aligned}
[\mathbf{U},\Sigma,\mathbf{V}_1] &= \mathbf{SVD}(\mathbf{A}^T, \texttt{'econ'}),\ \text{(economic SVD)}\\
\mathbf{Z}_1 &= \mathbf{S} \odot \left[\mathbf{V}_1^T \left(\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T + \lambda \mathbf{Z}_{\text{ref}}\right) \mathbf{V}_1\right],\\
\mathbf{S} &= \frac{1}{\Sigma^2 \mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T \Sigma^2 + \lambda \mathbf{1}\mathbf{1}^T},\\
\mathbf{P} &= \mathbf{V}_1 \mathbf{V}_1^T,\\
\mathbf{Z}_2 &= (\Sigma^2 + \lambda \mathbf{I})^{-1} \mathbf{V}_1^T (\mathbf{A}\mathbf{D}^T + \lambda \mathbf{Z}_{\text{ref}})(\mathbf{I} - \mathbf{P}).
\end{aligned}
$$

The fraction in $\mathbf{S}$ stands for the element-wise inversion (Hadamard inverse), and the notation $\odot$ stands for the element-wise product (Hadamard product). The inverse $\mathbf{Z}_\star^{-1}$ reads

$$
\begin{aligned}
\mathbf{Z}_\star^{-1} &= \mathbf{E}\left(\mathbf{Z}_1 - \mathbf{Z}_2 \mathbf{Z}_{\text{ref}}^{-1} \mathbf{Z}_2^T\right)^{-1} \mathbf{E}^T + (\mathbf{I} - \mathbf{P}) \mathbf{Z}_{\text{ref}}^{-1} (\mathbf{I} - \mathbf{P})\\
\mathbf{E} &= \mathbf{V}_1 - (\mathbf{I} - \mathbf{P}) \mathbf{Z}_{\text{ref}}^{-1} \mathbf{Z}_2^T. \qquad \text{(Inv-RSP)}
\end{aligned}
$$

The type-I update uses the matrix $\mathbf{Z}_\star^{-1}$, using $\mathbf{A} = \Delta\mathbf{X}$ and $\mathbf{D} = \Delta\mathbf{G}$. The type-II uses instead $\mathbf{Z}_\star$, with $\mathbf{A} = \Delta\mathbf{G}$ and $\mathbf{D} = \Delta\mathbf{X}$.

The next proposition shows the complexity of performing one matrix-vector multiplication with $\mathbf{Z}_\star$ and its inverse. The bottleneck of the method is the SVD of a $\mathbb{R}^{m\times d}$ matrix, whose complexity is $O(m^2 d)$, thus linear in the dimension.

Proposition 2.4.2. The complexity of evaluating $\mathbf{Z}_\star \boldsymbol{v}$ and $\mathbf{Z}_\star^{-1}\boldsymbol{v}$ is $O(m^2 d)$, assuming $m \ll d$ and that the complexity of $\mathbf{Z}_{\text{ref}}\boldsymbol{v}$ and $\mathbf{Z}_{\text{ref}}^{-1}\boldsymbol{v}$ is at most $O(m^2 d)$.

Robustness. The symmetric multisecant update can be used in two different modes, one that lets $\lambda \to 0$, the other, biased but more robust, that sets $\lambda > 0$.

The update formula is slightly simpler when $\lambda = 0$. However, due to the presence of matrix inversion, this may lead to instability issues in some cases, similarly to the BFGS method when

$$(\mathbf{x}_{k+1} - \mathbf{x}_k)^T (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)) \approx 0,$$

i.e., when the step and difference of gradients are close to being orthogonal. In BFGS, such issues are tackled by a filtering step, discarding the update if the scalar product goes below some threshold. Unfortunately, when the gradient is corrupted by some noise, the impact on the BFGS update can be huge.

In the case where $\lambda > 0$, we can show that our update is robust when $\mathbf{A}$ and $\mathbf{D}$ are corrupted.

Proposition 2.4.3. Let $\mathbf{Z}_\star(\lambda)$ be defined as the solution of (Sol-RSP) for some λ, and $\mathbf{Z}_\star(\lambda) = \lim_{\lambda\to 0} \mathbf{Z}_\lambda$. Let $\tilde{\mathbf{A}}$, $\tilde{\mathbf{D}}$ be a corrupted version of $\mathbf{A}$ and $\mathbf{D}$ where

$$\|\mathbf{A} - \tilde{\mathbf{A}}\| \leq \delta_{\mathbf{A}}, \quad \|\mathbf{D} - \tilde{\mathbf{D}}\| \leq \delta_{\mathbf{D}}.$$

Finally, let $\tilde{\mathbf{Z}}_\star(\lambda)$ be the solution of (Sol-RSP) using $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$. Then, we have

$$\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(0)\| \leq \underbrace{\|\mathbf{Z}_\star(\lambda) - \mathbf{Z}_\star(0)\|}_{\textbf{Bias}} + \underbrace{\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(\lambda)\|}_{\textbf{Stability}},$$

where

$$\|\mathbf{Z}_\star(\lambda) - \mathbf{Z}_\star(0)\| \leq \frac{\lambda\|\mathbf{Z}_\star(0) - \mathbf{Z}_{\text{ref}}\|}{\sigma_{\min}^2(\mathbf{A}) + \lambda}, \tag{2.4.3}$$

$$\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(\lambda)\| \leq \mathcal{O}\left(\frac{\delta_{\mathbf{A}} + \delta_{\mathbf{D}}}{\lambda}\right). \tag{2.4.4}$$

This suggests that λ should satisfy a trade-off to achieve the best performing approximation. Notice that when $\lambda = 0$ in the noise-less case, we recover the optimal $\mathbf{Z}_\star$, and when $\lambda \to \infty$, we have $\mathbf{Z}_\star = \mathbf{Z}_{\text{ref}}$.

Our result is called *robust* as we can bound the maximum perturbation without restriction on its magnitude. This is *not* the case in [42], whose main assumption is $\delta_{\mathbf{A}} \leq \sigma_{\min}(\mathbf{A})$ (which is extremely restrictive), where $\sigma_{\min}$ is the smallest non-zero singular value of $\mathbf{A}$.

Since the singular values of $\mathbf{A}$ are, in practice, often small, it is always recommended to set a small λ: we will show latter, in the numerical experiments, that even for quadratic functions (i.e., in the "perturbation-free regime"), a small value of λ drastically changes the final result, as this makes the method robust to numerical noise.

Scaling of λ. The parameter λ has to be scaled w.r.t. the problem input. It is clear, from Theorem 2.4.1, that the role of λ is to regularize the matrix inversion by lower-bounding the eigenvalues of the inverted matrix. Therefore, we advise to set $\lambda = \bar{\lambda}\|\mathbf{A}^T\mathbf{A}\|_2$, i.e., proportional to $\|\mathbf{A}^T\mathbf{A}\|_2$. This way, assuming $\sigma_{\min}$ small, the conditioning of $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}$ is upper-bounded by $1 + 1/\bar{\lambda}$.

2.5. Numerical Experiment

This section compares our symmetric multisecant algorithms to existing methods in the literature. We present in this section only a few experiments concerning stochastic-related experiments: We first compare the quality of the estimate of the Hessian (and its inverse). Then, we compare the speed of convergence when using this estimate to estimate the Newton-step in the case where the gradient is stochastic.

Hessian Recovery. Consider the problem of recovering the inverse of a symmetric Hessian $\mathbf{Q}^{-1}$ of a quadratic function, that satisfies

$$\mathbf{Q}^{-1}\Delta\mathbf{G} = \Delta\mathbf{X}, \quad \mathbf{Q} = \mathbf{Q}^T.$$

However, we have only access to $\tilde{\Delta\mathbf{G}}$, a corrupted version of $\Delta\mathbf{G}$. This notably happens when the oracle provides stochastic gradients.

In our case, we consider the worst-case ℓ_2 corruption

$$\tilde{\Delta\mathbf{G}} = \mathbf{U}_{\Delta\mathbf{G}} \max\{\Sigma_{\Delta\mathbf{G}} - \epsilon \cdot \sigma_1(\Delta\mathbf{G}), 0\}\mathbf{V}_{\Delta\mathbf{G}}^T,$$

where $\mathbf{U}_{\Delta\mathbf{G}}\Sigma_{\Delta\mathbf{G}}\mathbf{V}_{\Delta\mathbf{G}}^T$ is the SVD of $\Delta\mathbf{G}$, and ϵ is the relative perturbation intensity. When $\epsilon = 1$, the matrix $\tilde{\Delta\mathbf{G}}$ is full of zeros.

We estimate $\mathbf{Q}^{-1}$ using different techniques, that we compare using the relative residual error

$$\text{error}(\mathbf{Q}_{\text{est}}^{-1}) = \|\mathbf{Q}_{\text{est}}^{-1}\Delta\mathbf{G} - \Delta\mathbf{X}\|/\|\Delta\mathbf{X}\|.$$

Note that, in our error function, we use the noise-free version of $\Delta\mathbf{G}$.

Our baseline is the diagonal estimate, corresponding to the inverse of the Lipchitz constant of $\mathbf{Q}$, typically used as a step-size in the gradient method. We compare ℓ-BFGS, Multisecant Broyden updates [27] and our Type-1 and Type-2 multisecant algorithms, solving respectively (Inv-RSP) and (Sol-RSP) with $\mathbf{A} = \tilde{\Delta\mathbf{G}}$, $\mathbf{D} = \Delta\mathbf{X}$, $\mathbf{B}_0 = \mathbf{H}_0^{-1} = \|\mathbf{Q}\|$. The number of secant equations is 50 and the dimension of the problem is 250. The results are reported in Figure 2.2.

Optimization problem. We aim to solve

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{N}\sum_{i=0}^{N} \ell(\mathbf{a}_i^T\mathbf{x},\mathbf{b}_i) + \frac{\tau}{2}\|\mathbf{x}\|^2, \tag{2.5.1}$$

where $\ell(\cdot,\cdot)$ is a loss function. The pair $(\mathbf{A},\mathbf{b})$ is a dataset, where $\mathbf{a}_i \in \mathbb{R}^d$ is a data point composed by d features, and b_i is the label of the i^{th} data point.

Here, we present the specific case where ℓ is a quadratic loss, on the Madelon [38] dataset, with $\lambda = 10^{-2}\|\mathbf{A}\|$. We solve it using SAGA [19] stochastic estimates of the gradient, with a batch size of 64. We also have other experiments on other datasets, other losses and also on deterministic estimate of the gradient in Appendix C.8. We also show the evolution of the spectrum of $\mathbf{H}_k$ and $\mathbf{B}_k^{-1}$ in Figure C.1, Appendix C.8.

2.6. Discussion

We briefly discuss our contributions and propose possible improvements. Although our approach performs sufficiently well to be competitive with current qN updates, the authors believe the method can be improved in several aspects.

Contrary to BFGS, the update (2.4.2) (resp. (2.4.1)) does not guarantee its positive-definiteness when applied to a smooth and strongly convex function. However, for large enough λ the matrix is p.s.d. given that $\mathbf{H}_{\text{ref}}$ (resp. $\mathbf{B}_{\text{ref}}$) is also positive-definite. Also, it is possible to project a small matrix in (Inv-RSP) (resp. (Sol-RSP)) to ensure positive definiteness. We discuss this in more details in Appendix C.2. The ideal way would be to solve the symmetric Procrustes problem with a semi-definite constraint, but this is still considered as an open problem [42].

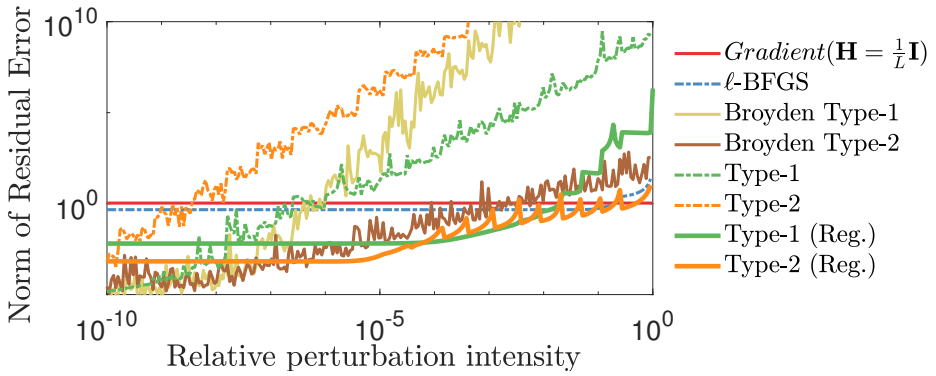


Fig. 2.1. Comparison of different methods to estimate a symmetric matrix. We see that symmetric multisecant methods perform well in a small-noise regime, but quickly get out of control for larger perturbations. This is not the case for their regularized counterpart ($\lambda = 10^{-10}$), clearly showing a more stable behavior. BFGS performs poorly compared to multisecant algorithms, since it can only satisfy one secant equation at a time. Finally, the type-II multisecant Broyden method seems stable, but does not recover a symmetric matrix.

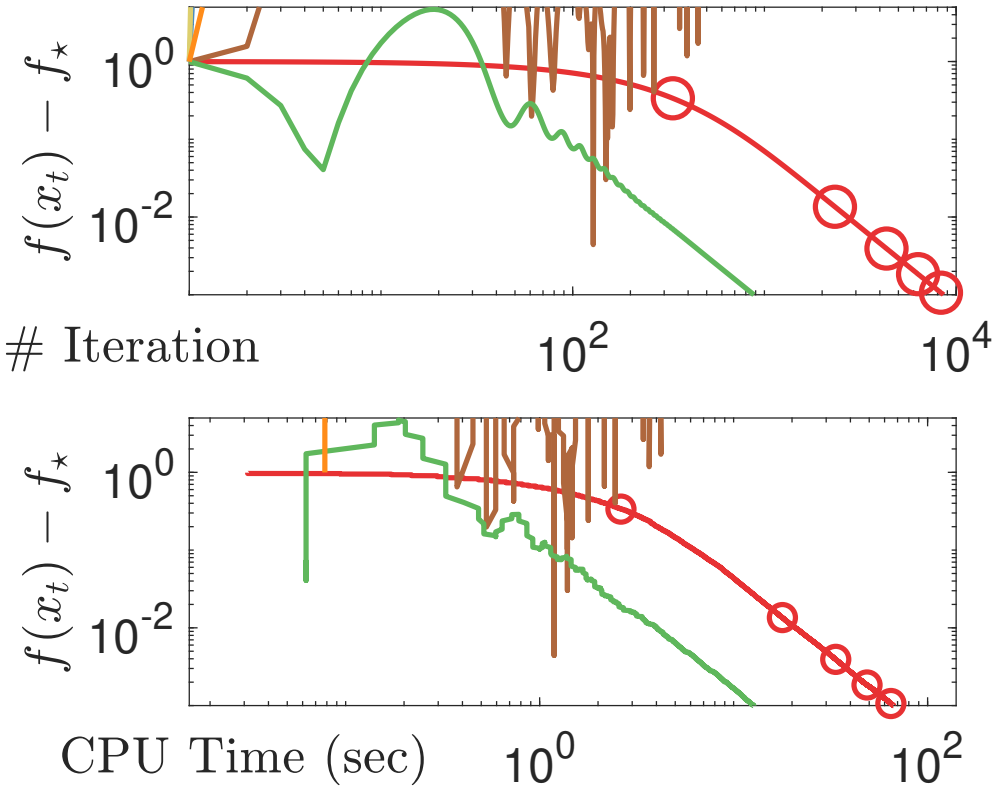


Fig. 2.2. Comparison of the stability of qN methods with stochastic gradients on Madelon dataset. We report the function value of the average of the iterates. The batch size is 64 points. Since the function is stochastic, we used only unitary stepsizes. The memory is 25, and the relative regularization $\bar{\lambda} = 10^{-2}$. The condition number is 10^3. $\ell-$BFGS and the Type-I multisecant Broyden are divergent in this situation. With unitary stepsizes, the regularized symmetric multisecant Type-I method is slightly faster than stochastic gradient.

A direct consequence of the non positive-definiteness is the lack of robustness guarantees for the Type-I method, that inverts a matrix that is possibly not positive definite. Therefore, it is probably impossible to bound the smallest eigenvalue, unless we use the robust projection trick in Appendix C.2. Surprisingly however, in our experiments the Type-I method seems to be the most stable among all updates.

Moreover, we considered here a plain method with *no preconditioner*. In BFGS and DFP updates, the preconditioner $\mathbf{W}$ is *any* matrix such that $\mathbf{W}\Delta\mathbf{X} = \Delta\mathbf{G}$ where $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are *vectors*. This matrix is used implicitly in the update: all occurrences of $\mathbf{W}\Delta\mathbf{X}$ are replaced by $\Delta\mathbf{G}$, in a way that $\mathbf{W}$ disappears. We cannot use a similar trick here, since such matrices do not exist in general when $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are matrices [71]. We propose in Appendix C.3 possible options to include such preconditioners that may potentially improve the method.

It is also possible to consider a general qN step, that takes the direction $\mathbf{HG}\boldsymbol{v}$ (or $\mathbf{B}^{-1}\mathbf{G}\boldsymbol{v}$), where $\boldsymbol{v}$ is a vector that sums to one, instead of taking the direction computed with the latest gradient, $\mathbf{H}\nabla f(\mathbf{x}_k)$. In the special case where $\boldsymbol{v}$ is full of zeros but one as the last element, this reduces to the standards qN step. We discuss this strategy in Appendix C.4, and we suspect this technique may reduce even more the impact of the noise on the qN step if $\boldsymbol{v}$ is chosen to be the averaging vector $\mathbf{1}_m/m$, for instance.

The complexity of the method is somewhat worse than current qN methods: $O(m^2d)$ instead of $O(md)$. The authors believe it may be possible to reduce the complexity by a factor m by using a low-rank SVD update [9] and by changing our direct formulas in Theorem 2.4.1 into recursive ones.

Another interesting direction is the study of the the matrix $\mathbf{C}$ that forms $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$. We suspect that, in the case where those matrices are corrupted, choosing the right $\mathbf{C}$ may affect the stability of the method. For instance, it is possible to design $\mathbf{C}$ to set more weight on some selected secant equations that may be more recent, or that contain less noise.

We proposed a novel method with distinct theoretical properties, including symmetry, optimality on quadratics with *unitary stepsize*, and robustness, and which performs encouragingly well in practice. In view of the new questions that multisecant methods raise, we hope our work can add to efforts for the design of possibly other, better-performing quasi-Newton schemes.

Chapitre 3

Conclusion

In this thesis, we illustrate new improvements and analyses for two fundamental types of convex optimization problems from the starting point of affine invariance. Specifically, we introduce new structural assumptions, e.g., directional smoothness (Chapter 1), and further derive an affine invariant analysis of Frank-Wolfe over strongly convex sets. As a byproduct, we present a new affine invariant backtracking line-search algorithm via directional smoothness and a new explanation for why the existing backtracking line-search works efficiently. In parallel, we present a promising direction to accelerate FW over strongly convex sets using duality gap techniques (Appendix A) and an another version of smoothness.

On the other hand, we investigate techniques that approximates the Newton step by estimating the Hessian using secant equations. To overcome the impossibility of having both a symmetric and multi-secant update, we propose a symmetric multisecant update satisfying the secant equations in a least-squares sense. In detail, we demonstrate its optimality on quadratics with unitary stepsize, and prove the robustness of our algorithm with respect to gradient noise. Such guarantees enable them to be prospective candidates in the context of stochastic optimization.

Références bibliographiques

[1] N. Agarwal, B. Bullins, and E. Hazan. Second-order Stochastic Optimization for Machine Learning in Linear Time. *J. Mach. Learn. Res.*, 18(1):4148–4187, January 2017.

[2] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[3] Francis Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.

[4] Davide Ballabio, Francesca Grisoni, Viviana Consonni, and Roberto Todeschini. Integrated qsar models to predict acute oral systemic toxicity. *Molecular informatics*, 38(8-9):1800124, 2019.

[5] Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

[6] Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. An investigation of newton-sketch and subsampled newton methods. *Optimization Methods and Software*, pages 1–20, 2020.

[7] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2019.

[8] Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching l-bfgs method for machine learning. *arXiv preprint arXiv:1802.05374*, 2018.

[9] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415:20–30, 05 2006.

[10] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.

[11] C. G. Broyden. The Convergence of a Class of Double-Rank Minimization Algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 09 1970.

[12] Richard H. Byrd, S. L. Hansen, Jorge Nocedal, and Yoram Singer. A Stochastic Quasi-Newton Method for Large-scale Optimization. *SIAM Journal on Optimization*, 26:1008–1031, 2016.

[13] Richard H. Byrd, Humaid Fayez Khalfan, and Robert B. Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM J. on Optimization*, 6(4):1025–1039, April 1996.

[14] K.L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

[15] Samuel A Danziger, S Joshua Swamidass, Jue Zeng, Lawrence R Dearth, Qiang Lu, Jonathan H Chen, Jianlin Cheng, Vinh P Hoang, Hiroto Saigo, Ray Luo, et al. Functional census of mutation sequence spaces: the example of p53 cancer rescue mutants. *IEEE/ACM transactions on computational biology and bioinformatics*, 3(2):114–125, 2006.

[16] Alexandre d'Aspremont, Cristobal Guzman, and Martin Jaggi. Optimal affine-invariant smooth minimization algorithms. *SIAM Journal on Optimization*, 28(3):2384–2405, 2018.

[17] W.C. Davidon. Variable metric method for minimization. *Technical Report ANL 5990 (revised), Argonne National Laboratory, Argonne, Il*, 1959.

[18] W.C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1:1–17, 1991.

[19] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[20] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[21] Ron S. Dembo and Trond Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, Jun 1983.

[22] V. F. Demyanov and A. M. Rubinov. Approximate methods in optimization problems. *Modern Analytic and Computational Methods in Science and Mathematics*, 1970.

[23] Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

[24] Jelena Diakonikolas and Lorenzo Orecchia. Conjugate gradients and accelerated methods unified: The approximate duality gap view. *arXiv preprint arXiv:1907.00289*, 2019.

[25] Joseph C Dunn. Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. *SIAM Journal on Control and Optimization*, 17(2):187–211, 1979.

[26] Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3052–3060, Cambridge, MA, USA, 2015. MIT Press.

[27] Haw-ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.

[28] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317?322, 1970.

[29] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[30] Futoshi Futami, Zhenghang Cui, Issei Sato, and Masashi Sugiyama. Bayesian posterior approximation via greedy particle optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3606–3613, 2019.

[31] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[32] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23?26, 1970.

[33] Vladimir V Goncharov and Grigorii E Ivanov. Strong and weak convexity of closed sets in a hilbert space. In *Operations research, engineering, and cyber security*, pages 259–297. Springer, 2017.

[34] Robert M. Gower, Donald Goldfarb, and Peter Richtárik. Stochastic Block BFGS: Squeezing More Curvature out of Data. In *ICML*, 2016.

[35] Robert Mansel Gower and Jacek Gondzio. Action constrained quasi-newton methods. *arXiv preprint arXiv:1412.8045*, 2014.

[36] Jacques Guélat and Patrice Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 1986.

[37] David H Gutman and Javier F Pena. The condition number of a function relative to a set. *Mathematical Programming*, pages 1–40, 2020.

[38] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[39] Isabelle Guyon, Jiwen Li, Theodor Mader, Patrick A Pletscher, Georg Schneider, and Markus Uhr. Competitive baseline methods set new standards for the nips 2003 feature selection benchmark. *Pattern recognition letters*, 28(12):1438–1444, 2007.

[40] F.J. Henk Don. On the symmetric solutions of a linear matrix equation. *Linear Algebra and its Applications*, 93:1–7, 07 1987.

[41] P. Hennig. Probabilistic interpretation of linear solvers. *SIAM Journal on Optimization*, 25(1):234–260, 2015.

[42] N. J. Higham. The symmetric Procrustes problem. *BIT*, 28, 03 1988.

[43] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th international conference on machine learning*, number CONF, pages 427–435, 2013.

[44] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(2), 2010.

[45] Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Projection-free optimization on uniformly convex sets. 2020.

[46] Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Restarting frank-wolfe. *arXiv preprint arXiv:1810.02429*, 2018.

[47] Nicholas Kushmerick. Learning to remove internet advertisements. In *Proceedings of the third annual conference on Autonomous Agents*, pages 175–181, 1999.

[48] Simon Lacoste-Julien and Martin Jaggi. An affine invariant linear convergence analysis for frank-wolfe algorithms. *arXiv preprint arXiv:1312.7864*, 2013.

[49] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank–Wolfe optimization variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 496–504. Curran Associates, Inc., 2015.

[50] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *International Conference on Machine Learning*, pages 53–61. PMLR, 2013.

[51] Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

[52] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989.

[53] Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

[54] Aryan Mokhtari and Alejandro Ribeiro. Global Convergence of Online Limited Memory BFGS. *Journal of Machine Learning Research*, 16:3151–3181, 2015.

[55] Marco Molinaro. Curvature of feasible sets in offline and online optimization. 2020.

[56] Philipp Moritz, Robert Nishihara, and Michael Jordan. A Linearly-Convergent Stochastic L-BFGS Algorithm. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 249–258, Cadiz, Spain, 09–11 May 2016. PMLR.

[57] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[58] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[59] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[60] Yu E Nesterov. An o (1/k)-rate of convergence method for smooth convex functions minimization. In *Dokl. Acad. Nauk SSSR*, volume 269, pages 543–547, 1983.

[61] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[62] J. Nocedal and S.J. Wright. *Numerical optimization, Second Edition*. Springer Verlag, 1999.

[63] Fabian Pedregosa, Geoffrey Negiar, Armin Askari, and Martin Jaggi. Linearly convergent frank-wolfe with backtracking line-search. In *International Conference on Artificial Intelligence and Statistics*, pages 1–10. PMLR, 2020.

[64] Javier Pena. Generalized conditional subgradient and generalized mirror descent: duality, convergence, and symmetry. *arXiv preprint arXiv:1903.00459*, 2019.

[65] M. J. Powell. How bad are the BFGS and DFP methods when the objective function is quadratic? *Math. Program.*, 34:34–47, 1986.

[66] Jarrid Rector-Brooks, Jun-Kun Wang, and Barzan Mozafari. Revisiting projection-free optimization for strongly convex constraint sets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1576–1583, 2019.

[67] Francesco Rinaldi and Damiano Zeffiro. A unifying framework for the analysis of projection-free first-order methods under a sufficient slope condition. 2020.

[68] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[69] Thorsten Rohwedder and Reinhold Schneider. An analysis for the diis acceleration method used in quantum chemistry calculations. *Journal of mathematical chemistry*, 49(9):1889, 2011.

[70] Mark Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab, 2005.

[71] Robert B Schnabel. Quasi-newton methods using multiple secant equations. Technical report, University of Colorado Boulder, Computer Science Department, 1983.

[72] Nicol N. Schraudolph, Jin Yu, and Simon Gunter. A Stochastic Quasi-Newton Method for Online Convex Optimization. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 436–443, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[73] Damien Scieur, Francis Bach, and Alexandre d'Aspremont. Nonlinear acceleration of stochastic algorithms. In *Advances in Neural Information Processing Systems*, pages 3982–3991, 2017.

[74] Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016.

[75] Damien Scieur, Edouard Oyallon, Alexandre d'Aspremont, and Francis Bach. Online regularized nonlinear acceleration. *arXiv preprint arXiv:1805.09639*, 2018.

[76] D.F. Shanno. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computing*, 24:647–656, 07 1970.

[77] Paul Swoboda and Vladimir Kolmogorov. Map inference via block-coordinate frank-wolfe algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11146–11155, 2019.

[78] Alex Toth and CT Kelley. Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.

[79] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

[80] Henry Wolkowicz. Geometry of optimality conditions and constraint qualifications: The convex case. *Mathematical Programming*, 19(1):32–60, 1980.

[81] Max. A. Woodbury. Inverting modified matrices. *Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ*, 1950.

[82] Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3000–3008. Curran Associates, Inc., 2016.

[83] Yi Xu and Tianbao Yang. Frank-wolfe method is automatically adaptive to error bound condition. 2018.

[84] Junzi Zhang, Brendan O'Donoghue, and Stephen Boyd. Globally convergent type-i anderson acceleration for non-smooth fixed-point iterations. *arXiv preprint arXiv:1808.03971*, 2018.

Annexe A

Acceleration of Frank-Wolfe

A.1. Frank-Wolfe Algorithm and Notations

In this section, we provide a brief introduction to the Frank-Wolfe algorithm and associated useful representations. This method is usually used to solve,

$$\min_x F(x) = f(x) + h(x), \tag{A.1.1}$$

where $f(x)$ is a smooth function, and h the indicator function of a convex set $\mathcal{C}$. More generally, h can be a function whose dual h^* is known [3].

Definition A.1.1. The conjugate function of a convex function h is a function h^* that satisfies

$$h^*(d) = \max_{s \in \mathrm{dom}(h)} d^T s - h(s), \quad \delta h^*(d) \in \arg\max_{s \in \mathrm{dom}(h)} d^T s - h(s),$$

where δh^* is a sub-gradient of h^*.

The conditional gradient method perform formally the following steps,

$$x_{k+1} = (1 - \beta_{k+1}) x_k + \beta_{k+1} \nabla h^*(-\nabla f(x_k)) \quad ; \qquad \beta_{k+1} \in [0, 1]. \tag{A.1.2}$$

In the case where h is an indicator function of a convex set $\mathcal{C}$, i.e.,

$$h(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{cases} \tag{A.1.3}$$

the algorithm can be simplified as follow,

$$\begin{cases} s_{k+1} & = \arg\min_{s \in \mathcal{C}} \nabla f(x_k)^T s, \\ x_{k+1} & = (1 - \beta_{k+1}) x_k + \beta_{k+1} s_{k+1}. \end{cases} \tag{FW}$$

This method is usually called Frank-Wolfe. The point s_{k+1} is called the *Frank-Wolfe corner*, and is found using a *linear minimization oracle* (LMO).

In the general case where f is a smooth function and $\mathcal{C}$ a convex set, the rate of convergence of the Frank-Wolfe algorithm is bounded by $\mathcal{O}(1/k)$. With additional assumptions, it is possible to improve this rate to $\mathcal{O}(1/k^2)$. In what follows, we aim to accelerate FW using duality gap techniques.

A.2. Accelerated Gradient Descent and Duality Gap Technique

In this section, we study the method and proof related to the duality gap technique from [23, 24], where the technique is used to design the accelerated proximal gradient algorithm. Thus, we recall the main result in this section.

Accelerated gradient descent solves the following minimization problem,

$$\min_x F(x) = f(x) + h(x), \tag{A.2.1}$$

where $f(x)$ is smooth and convex. This means, for all x, y in the domain of $f + h$,

$$f(x) + \nabla f(x)^T(y - x) \leq f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L_f}{2}\|y - x\|^2 \tag{A.2.2}$$

The function h is convex, potentially non-smooth, whose *proximal operator* is *simple*,

$$\mathbf{prox}_{\lambda h}(x) = \arg\min_{s \in \mathrm{dom}(h)} \lambda h(s) + \frac{1}{2}\|s - x\|^2.$$

In the case where h is an indicator function of a constrained convex set $\mathcal{C}$, the proximal operator is equal to the projection onto the set $\mathcal{C}$.

The duality gap technique sees AGD as a way to build an upper and lower bound for the function $F(x)$. The upper bound $U(x)$ is given by the smoothness of the function f,

$$U_k(x) = f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{L}{2}\|x - y_k\|^2 + h(x).$$

Then, the lower-bound is build as a weighted average of inequalities from convexity,

$$A_k L_k(x) = \sum_{i=0}^{k} \alpha_i \Big(f(y_i) + \nabla f(y_i)(x - y_i) + h(x) \Big), \quad \alpha_i > 0, \;\; A_k = \sum_{i=0}^{k} \alpha_i.$$

Finally, the algorithm reads

$$\begin{aligned} x_{k+1} &= \arg\min_x U_k(x), \\ v_{k+1} &= \arg\min_x L_k(x) + \frac{1}{2A_k}\|x - x_0\|^2, \\ y_{k+1} &= \frac{A_k x_{x+1} + \alpha_{k+1} v_{k+1}}{A_{k+1}}. \end{aligned} \tag{AGD}$$

The first step is intuitive, and corresponds to gradient descent. The second is a bit more tricky, because of the regularization. However, this regularization is crucial for the algorithm: for instance, in the case where $h(x) = 0$, without regularization, the problem is unbounded as we minimize a linear function over R^n.

The duality gap technique consists in bounding the optimality gap function with a gap G_k:

$$F(x_k) - F^* \leq G_k \triangleq U_k(x_k) - \min_x \left[L_k(x) + \frac{1}{2A_k}\|x - x_0\|^2 \right] + \frac{1}{2A_k}\|x_0 - x^*\|^2.$$

Then, it remains to design the parameters α_i to ensure

$$A_{k+1}G_{k+1} \leq A_k G_k, \quad \text{which implies} \quad f(x_k) - f(x^\star) \leq G_k \leq \frac{A_0 G_0}{A_k}.$$

A.3. Bounds for the Duality Gaps

We consider the acceleration of Frank-Wolfe algorithm over a strongly convex and smooth set $Q \subseteq \mathbb{R}^n$. We have the following property for the iterate before and after a Frank-Wolfe step:

$$\langle \nabla f(T(x_t)), x_t - T(x_t) \rangle, \tag{A.3.1}$$

where $T(x_t)$ denotes the iterate obtained after a Frank-Wolfe step, that is, we have $T(x_t) = x_{t+1}$ such that

$$\begin{aligned} s_t &\in \operatorname*{argmin}_{s_t \in Q} \langle \nabla f(x_t), s \rangle, \\ x_{t+1} &= (1 - \gamma_t)x_t + \gamma_t s_t. \end{aligned} \tag{A.3.2}$$

To make use of the dual averaging technique, we need to carefully select the averaging weights and step-sizes for the linear combinations of Frank-Wolfe corners and the current iterates. In what follows, we note that f is β-smooth and μ-strongly convex.

By the technique of estimate functions, we define a sequence of increasing scaling coefficients $\{A_k\}_{k=0}$ such that

$$A_0 = 0, \quad A_k \overset{\text{def}}{=} A_{k-1} + a_k, \quad k \geq 1. \tag{A.3.3}$$

We adopt the following Algorithm 4 to achieve acceleration with only linear minimization oracle available.

Algorithm 4 Accelerated Frank-Wolfe Algorithm (Directly adapted from Nesterov's method)

Require: A strongly convex and strongly smooth set $Q \subseteq \mathbb{R}^n$, initial iterate x_0, number of iterations T,
Set stepsize $\eta \leftarrow T^{-1/2}$
for $k = 0, \ldots, T-1$ **do**
  Determine α_{k+1} by equation $\frac{\alpha_{k+1}^2}{2(A_k+\alpha_{k+1})} = \frac{1+\mu A_k}{\beta}$.
  Solve $v_k = \operatorname{argmin} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \langle \nabla f(x_i), x - x_i \rangle\right)$.
  Set query point $y_k = \frac{A_k x_k + \alpha_k v_k}{A_{k+1}}$.
  $s_k \leftarrow \operatorname{argmin}_{s \in Q} \langle s, \nabla f(y_k) \rangle$.
  $x_{k+1} \leftarrow y_k + \gamma_k (s_k - y_k)$.
end for
Output: x_T

We aim to show a global convergence rate formulated as below

$$f(x_k) - f(x^*) \leq \frac{\|x^0 - x^*\|^2}{2A_k} \tag{A.3.4}$$

via an immediate quantities of estimate functions $M_k := \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \langle \nabla f(x_i), x - x_i \rangle\right)$. We will show that our algorithm maintains recursively the following properties:

$$\mathcal{P}_k^1 : A_k f(x_k) \leq M_k^* = \min_{x \in Q} M_k(x) \tag{A.3.5}$$

$$\mathcal{P}_k^2 : M_k(x) \leq A_k f(x) + \frac{1}{2}\|x - x_0\|^2, \ \forall x \in Q. \tag{A.3.6}$$

A.3.1. Upper and Lower Bounds in the Unconstrained Case

Upper bound U_k: We use y_k constructed from the previous gradient query points $\{x_i\}_{i=0}^k$ and the gradient oracle answers $\{\nabla f(x_i)\}_{i=0}^k$ to give the upper bound, By setting $y_k = x_k - 1/L \nabla f(x_k)$, we have

$$f(y_k) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \triangleq U_k. \tag{A.3.7}$$

Lower bound L_k: Each queried gradient $\nabla f(x_i)$ leads to a lower bound on the function f as follows,

$$f(u) \geq f(x_i) + \langle \nabla f(x_i), u - x_i \rangle + \frac{\mu}{2}\|u - x_i\|^2, \ \forall u \in \mathbb{R}^d \tag{A.3.8}$$

By the scheme of AGDT (Approximate Duality Gap Technique) in [24], we dispatch a measure $a_k > 0$ to each iteration k, and let $A_k = \sum_{i=0}^{k} a_i$. We derive the overall lower bound by averaging the bound for each x_i in (A.3.8) with normalized weights proportional

to a_i as follows,

$$\begin{aligned} f(u) &\geq \sum_{i=0}^{k} \frac{a_i}{A_k} \left(f(x_i) + \langle \nabla f(x_i), u - x_i \rangle + \frac{\mu}{2} \|u - x_i\|^2 \right) \\ &\geq \min_{u \in \mathbb{R}^d} m_k(u) \\ &\triangleq m_k(v_k) \\ &\triangleq L_k. \end{aligned} \tag{A.3.9}$$

where

$$m_k(u) = \sum_{i=0}^{k} a_i \left(f(x_i) + \langle \nabla f(x_i), u - x_i \rangle + \frac{\mu}{2} \|u - x_i\|^2 \right), \tag{A.3.10}$$

$$v_k = \operatorname*{argmin}_{u \in \mathbb{R}^d} m_k(u). \tag{A.3.11}$$

Hence the duality gap at iteration k is defined as $G_k = U_k - L_k \geq f(y_k) - f(x^*)$. In what follows we will show that $A_k G_k$ is non-increasing with the iteration k, which implies that

$$f(y_k) - f(x^*) \leq G_k \leq \frac{A_0 G_0}{A_k}. \tag{A.3.12}$$

The following two lemmas characterize how the duality gap G_k proceeds according to the iteration k, and further provide the convergence certificate.

Lemma A.3.1 (Initial estimate)**.** Let $x_0 \in \mathbb{R}^d$ be an arbitrary initial point and optimality gap estimate follow the notation above, then we have

$$A_0 G_0 \leq \frac{L - \mu_0}{2} \|x_0 - x^*\|^2. \tag{A.3.13}$$

Lemma A.3.2 (Monotonicity of $A_k G_k$)**.** For any $k \geq 1$, we have

$$A_k G_k \leq A_{k_1} G_{k-1}. \tag{A.3.14}$$

Démonstration.

$$m_{k-1}(v_k) = m_{k-1}(v_{k-1}) + \frac{A_{k-1}\mu}{2} \|v_k - v_{k-1}\|^2, \tag{A.3.15}$$

$$m_k(v_k) = m_{k-1}(v_{k-1}) + \frac{A_{k-1}\mu}{2} \|v_k - v_{k-1}\|^2 + a_k \langle \nabla f(x_k), v_k - x_k \rangle + \frac{a_k \mu}{2} \|v_k - x_k\|^2. \tag{A.3.16}$$

As we can explicitly write v_k as

$$\begin{aligned} v_k &= \frac{\mu \sum_{i=0}^{k} a_i x_i - \sum_{i-0}^{k} a_i \nabla f(x_i)}{\mu A_k} \\ &= \frac{A_{k-1}}{A_k} v_{k-1} + \frac{a_k}{A_k} x_k - \frac{a_k}{\mu A_k} \nabla f(x_k), \end{aligned} \tag{A.3.17}$$

we apply Jensen's inequality to (A.3.16) to obtain

$$\begin{aligned} m_k(v_k) &\geq m_{k-1}(v_{k-1}) + \frac{\mu A_k}{2}\left\|v_k - \frac{A_{k-1}}{A_k}v_{k-1} - \frac{a_k}{A_k}x_k\right\|^2 + a_k\langle\nabla f(x_k), v_k - x_k\rangle \\ &\overset{(A.3.17)}{=} m_{k-1}(v_{k-1}) + \frac{a_k^2}{2\mu A_k}\|\nabla f(x_k)\|^2 + \frac{a_k A_{k-1}}{A_k}\langle\nabla f(x_k), v_{k-1} - x_k\rangle - \frac{a_k^2}{\mu A_k}\|\nabla f(x_k)\|^2 \\ &= m_{k-1}(v_{k-1}) - \frac{a_k^2}{2\mu A_k}\|\nabla f(x_k)\|^2 + \frac{a_k A_{k-1}}{A_k}\langle\nabla f(x_k), v_{k-1} - x_k\rangle. \end{aligned} \tag{A.3.18}$$

■

A.3.2. Upper and Lower Bounds with Smooth and Strongly Convex Sets

We are interested in solving the following constrained convex smooth optimization with only a linear minimization oracle.

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } x \in Q \subseteq \Omega \end{aligned} \tag{A.3.19}$$

where in our case $\Omega = \mathbb{R}^d$ (d denotes the dimension of the space), Q is a μ_c-strongly convex and L_c-strongly smooth set, *i.e.*, Q admits a representation of a convex function $C : \Omega \to \mathbb{R}$ such that

$$Q = \{x \in \Omega : C(x) \leq 0\}, \tag{A.3.20}$$

and for any $x, y \in Q$, it turns out to satisfy the following function-alike inequalities:

$$C(y) \leq C(y) + \nabla C(y)^\top (x - y) + \frac{L_c}{2}\|x - y\|^2, \tag{A.3.21}$$

$$C(y) \geq C(y) + \nabla C(y)^\top (x - y) + \frac{\mu_c}{2}\|x - y\|^2. \tag{A.3.22}$$

Upper bounds for $f(y_k)$: We define the upper ball $\bar{B}_k$ at iteration k as a relaxation set contained in Q due to the strong smoothness property:

$$\bar{B}_k = \left\{x \in \Omega : \sum_{i=0}^{k} \gamma_i \left(C(x_i) + \nabla C(x_i)^\top (x - x_i) + \frac{L_c}{2}\|x - x_i\|^2\right) \leq 0\right\}, \tag{A.3.23}$$

where $\gamma_i \geq 0$ for $i \in [n]$, and we denote $\Gamma_k = \sum_{i=0}^{k} \gamma_i$. Since f is L_c-strongly smooth, we have for any $k \geq 0$

$$C(x) \leq C(x_i) + \nabla C(x_i)^\top (x - x_i) + \frac{L_c}{2}\|x - x_i\|^2, \tag{A.3.24}$$

which implies $\bar{B}_k \subseteq Q$ for all $k \geq 0$. In fact, let $d_k = \sum_{i=0}^{k} \alpha_i \nabla f(x_i)$, $\bar{B}_k$ characterizes a ball centered at $\bar{O}_k$ with radius $\bar{r}_k$, where

$$\bar{O}_k = \sum_{i=0}^{k} \gamma_i \left(x_i - \frac{1}{L_c} \nabla C(x_i) \right), \tag{A.3.25}$$

$$\bar{r}_k = \sqrt{\frac{1}{2L_c} \left(\sum_{i=0}^{k} \gamma_i (\nabla C(x_i) - L_c x_i) \right)^2 - \sum_{i=0}^{k} \gamma_i \left(C(x_i) - \nabla C(x_i)^\top x_i + \frac{L_c}{2} \|x_i\|^2 \right)}. \tag{A.3.26}$$

By the definition of $\bar{B}_k$ and $C(x)$, we have $x_i - \frac{1}{L_c} \nabla C(x_i) \in Q$. Then we have

$$\begin{aligned} f(y_{k+1}) &= \min_{\substack{v_{k+1} \in Q \\ h \in [0,1]}} f(x_{k+1} + h(v_{k+1} - x_{k+1})) \\ &\leq \min_{\substack{v_{k+1} \in Q \\ h \in [0,1]}} f(x_{k+1}) + h \nabla f(x_{k+1})^\top (v_{k+1} - x_{k+1}) + \frac{h^2 L_f}{2} \|v_{k+1} - x_{k+1}\|^2 \qquad \text{(A.3.27)} \\ &\leq \min_{\substack{v_{k+1} \in Q \\ y_{k+1} \in \bar{B}_k}} f(x_{k+1}) + h \nabla f(x_{k+1})^\top (v_{k+1} - x_{k+1}) + \frac{h^2 L_f}{2} \|v_{k+1} - x_{k+1}\|^2 \qquad \text{(A.3.28)} \end{aligned}$$

Upper bounds (v2): The obtain s_k, the linear minimization oracle (LMO) solves the following linear optimization problem:

$$\min_{C(s) \leq 0} \nabla f(x_k)^\top (s - x_k), \tag{A.3.29}$$

of which the Lagrangian dual gives

$$\max_{\omega \geq 0} \mathcal{L}(\omega) = \max_{\omega \geq 0} \min_{s} \left\{ f(x_k)^\top (s - x_k) + \omega C(s) \right\}. \tag{A.3.30}$$

By the optimality condition, for any fixed ω we obtain the following at s_ω:

$$\nabla f(x_k) + \omega \nabla C(s_\omega) = 0. \tag{A.3.31}$$

By solving the optimality equation we have

$$s_\omega = \nabla C^*(\frac{-\nabla f(x_k)}{\omega}), \tag{A.3.32}$$

where ∇C^* is the conjugate of ∇C with $1/L_c$-strong convexity and $1/\mu_c$-strong smoothness. We denote by ω^* the optimal solution to (A.3.30). Since the set Q has non-empty interior, by Slater's condition [80] ω^* gives the same optimal value as (A.3.29). Hence by plugging (A.3.32) into (A.3.29) we have

$$\nabla f(x_k)^\top (s_k - x_k) = \nabla f(x_k)^\top \left(\nabla C^* \left(\frac{-\nabla f(x_k)}{\omega^*} \right) - x_k \right). \tag{A.3.33}$$

To derive the upper bound, we denote $d = \nabla f(x_k)$ and maximize the left hand side of (A.3.33) over d. Note that $d = -\nabla C(p) \cdot \omega^*$ for some $p \in Q$ by (A.3.31), we have

$$\begin{aligned}\nabla f(x_k)^\top (s_k - x_k) &\leq \max_d d^\top \left(\nabla C^* \left(\frac{-d}{\omega^*} \right) - x_k \right) \\ &\leq \max_p -\omega^* (p - x_k)^\top \nabla C(p). \end{aligned} \tag{A.3.34}$$

For any $k \geq 0$, we have

$$f(y_{k+1}) = \min_{\substack{s_{k+1} \in Q \\ h \in [0,1]}} f(x_{k+1} + h(s_{k+1} - x_{k+1}))$$

$$\leq \min_{\substack{s_{k+1} \in Q \\ h \in [0,1]}} f(x_{k+1}) + h \nabla f(x_{k+1})^\top (s_{k+1} - x_{k+1}) + \frac{h^2 L_f}{2} \|s_{k+1} - x_{k+1}\|^2 \tag{A.3.35}$$

$$\leq f(x_{k+1}) + \min_{h \in [0,1]} \left(h \nabla f(x_{k+1})^\top (s_{k+1} - x_{k+1}) + \frac{h^2 L_f}{2} \|s_{k+1} - x_{k+1}\|^2 \right) \tag{A.3.36}$$

$$\leq f(x_{k+1}) - \frac{\left(\nabla f(x_{k+1})^\top (s_{k+1} - x_{k+1}) \right)^2}{2 L_f \|s_{k+1} - x_{k+1}\|^2}. \tag{A.3.37}$$

By Lemma 1 in [31], we proceed to derive

$$\nabla f(x_{k+1})^\top (s_{k+1} - x_{k+1}) \leq -\frac{\mu_c \|x_{k+1} - s_{k+1}\|^2}{4} \|\nabla f(x_{k+1})\|. \tag{A.3.38}$$

By plugging the above inequality into (A.3.35) we have

$$\begin{aligned} f(y_{k+1}) - f(x_{k+1}) &\leq -\frac{\mu_c^2 \|\nabla f(x_{k+1})\|^2}{32 L_f} \|x_{k+1} - s_{k+1}\|^2 \\ &\overset{(b)}{\leq} -\frac{\mu_c^2 \|\nabla f(x_{k+1})\|^2}{32 L_f L_c^2} \|\nabla C(x_{k+1}) - \nabla C(s_{k+1})\|^2 \\ &\overset{(c)}{\leq} -\frac{\mu_c^2 \|\nabla f(x_{k+1})\|^2}{32 L_f L_c^2} \left\| \nabla C(x_{k+1}) + \frac{1}{\omega^*} \nabla f(x_{k+1}) \right\|^2 \\ &= -\frac{\mu_c^2 \|\nabla f(x_{k+1})\|^2}{32 L_f L_c^2 \omega^{*2}} \left\| \omega^* \nabla C(x_{k+1}) + \nabla f(x_{k+1}) \right\|^2, \end{aligned} \tag{A.3.39}$$

where (b) follows from the L_c-smoothness of set Q, and (c) is due to the optimality conditions. Then we can conclude the improvement on the upper bounds aas follows:

$$\begin{aligned}
A_{k+1}U_{k+1} - A_kU_k &= A_{k+1}f(y_{k+1}) - A_kf(y_k) \\
&= A_{k+1}\left(f(y_{k+1}) - f(x_{k+1})\right) - A_k\left(f(y_k) - f(x_{k+1})\right) + \alpha_{k+1}f(x_{k+1}) \\
&\overset{(A.3.39)}{\leq} -\frac{\mu_c^2 A_{k+1}\|\nabla f(x_{k+1})\|^2}{32L_fL_c^2\omega^{*2}}\left\|\omega^*\nabla C(x_{k+1}) + \nabla f(x_{k+1})\right\|^2 \\
&\quad - A_k\left(f(y_k) - f(x_{k+1})\right) + \alpha_{k+1}f(x_{k+1}) \\
&\leq -\frac{\mu_c^2 A_{k+1}\|\nabla f(x_{k+1})\|^2}{32L_fL_c^2\omega^{*2}}\left\|\omega^*\nabla C(x_{k+1}) + \nabla f(x_{k+1})\right\|^2 \\
&\quad + A_k\nabla f(x_{k+1})^\top(x_{k+1} - y_k) + \alpha_{k+1}f(x_{k+1}), \qquad \text{(A.3.40)}
\end{aligned}$$

where (A.3.40) comes from the convexity of the function f.

Lower bounds for $f(x^*)$: We define the lower ball $\underline{B}_k$ at iteration k as a relaxation set containing Q due to the strong convexity property:

$$\underline{B}_k = \left\{x \in \Omega : \sum_{i=0}^{k}\beta_i\left(C(x_i) + \nabla C(x_i)^\top(x - x_i) + \frac{\mu_c}{2}\|x - x_i\|^2\right) \leq 0\right\}, \qquad \text{(A.3.41)}$$

where $\gamma_i \geq 0$ for $i \in [n]$, and we denote $\Gamma_k = \sum_{i=0}^{k}\gamma_i$. Since f is L_c-strongly smooth, we have for any $k \geq 0$

$$C(x) \leq C(x_i) + \nabla C(x_i)^\top(x - x_i) + \frac{L_c}{2}\|x - x_i\|^2, \qquad \text{(A.3.42)}$$

which implies $\underline{B}_k \subseteq Q$ for all $k \geq 0$. In fact, let $d_k = \sum_{i=0}^{k}\alpha_i\nabla f(x_i)$, $\bar{B}_k$ characterizes a ball centered at $\underline{O}_k$ with radius $\underline{r}_k$, where

$$\underline{O}_k = \sum_{i=0}^{k}\gamma_i\left(x_i - \frac{1}{L_c}\nabla C(x_i)\right), \qquad \text{(A.3.43)}$$

$$\underline{r}_k = \sqrt{\frac{1}{2L_c}\left(\sum_{i=0}^{k}\gamma_i(\nabla C(x_i) - L_cx_i)\right)^2 - \sum_{i=0}^{k}\gamma_i\left(C(x_i) - \nabla C(x_i)^\top x_i + \frac{L_c}{2}\|x_i\|^2\right)}. \qquad \text{(A.3.44)}$$

By the convexity of f we have for any $u \in Q$ and $i \in [n]$,

$$f(u) \geq f(x_i) + \nabla f(x_i)^\top(u - x_i). \qquad \text{(A.3.45)}$$

Similar to the argument in [23], we assign to each each iteration k a weight $\alpha_k > 0$ and denote by $A_k = \sum_{i=0}^{k}\alpha_i$ the cumulative weight of all iterations up to k. Furthermore, we

consider the lower bound by averaging the bound for each $i \in [k]$ with weight $\frac{\alpha_i}{A_k}$:

$$f(u) \geq \frac{1}{A_k} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (u - x_i) \right). \tag{A.3.46}$$

Taking $u = x^*$ on the left-hand side and minimizing over u within the lower ball $\underline{B}_k$ on the right side yields $f(x^*) \geq L_k$, the lower bound at iteration k, where

$$\begin{aligned} L_k &= \frac{1}{A_k} \min_{u \in \underline{B}_k} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (u - x_i) \right) \\ &\triangleq \frac{1}{A_k} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (\underline{v}_k - x_i) \right) \end{aligned} \tag{A.3.47}$$

Thus we have the following monotonicity on $A_k L_k$ for $k \in [n]$:

$$\begin{aligned} A_{k+1} L_{k+1} - A_k L_k &= M_{k+1}(\underline{v}_{k+1}) - M_k(\underline{v}_k) \\ &= M_k(\underline{v}_{k+1}) - M_k(\underline{v}_k) + \alpha_{k+1} \left[f(x_{k+1}) + \nabla f(x_{k+1})^\top (\underline{v}_{k+1} - x_{k+1}) \right] \\ &\overset{(a)}{=} \sum_{i=0}^{k} \alpha_i \nabla f(x_i)^\top (\underline{v}_{k+1} - \underline{v}_k) + \alpha_{k+1} f(x_{k+1}) + \alpha_{k+1} \nabla f(x_{k+1})^\top (\underline{v}_{k+1} - x_{k+1}) \end{aligned} \tag{A.3.48}$$

$$\overset{(b)}{\geq} \sum_{i=0}^{k+1} \alpha_i \nabla f(x_i)^\top (\underline{v}_{k+1} - x_{k+1}) + \alpha_{k+1} f(x_{k+1}), \tag{A.3.49}$$

where (b) uses the minimization property of the Frank-Wolfe oracle. Another way to proceed on this is as follows.

Lower bounds (v2): We define

$$m_k(v) = \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (v - x_i) \right). \tag{A.3.50}$$

By the convexity of f we have for any $u \in Q$ and $i \in [k]$,

$$f(u) \geq f(x_i) + \nabla f(x_i)^\top (u - x_i). \tag{A.3.51}$$

Similar to the argument in [23], we assign to each each iteration k a weight $\alpha_k > 0$ and denote by $A_k = \sum_{i=0}^{k} \alpha_i$ the cumulative weight of all iterations up to k. Furthermore, we consider the lower bound by averaging the bound for each $i \in [k]$ with weight $\frac{\alpha_i}{A_k}$:

$$f(u) \geq \frac{1}{A_k} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (u - x_i) \right). \tag{A.3.52}$$

Taking $u = x^*$ on the left-hand side and minimizing over u set Q on the right-hand side yields

$$\begin{aligned} f(x^*) &\geq \frac{1}{A_k} \min_{u \in Q} \sum_{i=0}^{k} \alpha_i \left(f(x_i) + \nabla f(x_i)^\top (u - x_i) \right) \\ &= \frac{1}{A_k} \min_{u \in Q} m_k(u). \end{aligned} \tag{A.3.53}$$

We also denote the minimizer of $m_k(u)$ over Q by $v_k = \operatorname{argmin}_v m_k(v)$. By definition we have

$$f(x^*) \geq \frac{1}{A_{k+1}} m_{k+1}(v_{k+1}) = \frac{1}{A_{k+1}} \min_{v \in Q} \left\{ m_k(v) + \alpha_{k+1} f(x_{k+1}) + \nabla f(x_{k+1})^\top (v - x_{k+1}) \right\}. \tag{A.3.54}$$

By considering the Lagrangian of the right hand side of (A.3.54), we define

$$\mathcal{L}_{k+1}(v, \lambda) = \frac{1}{A_{k+1}} \left(m_k(v) + \alpha_{k+1} f(x_{k+1}) + \nabla f(x_{k+1})^\top (v - x_{k+1}) \right) + \lambda C(v), \tag{A.3.55}$$

and the economic function will be

$$\begin{aligned} g(\lambda) &= \min_v \left\{ \frac{1}{A_{k+1}} \left(m_k(v) + \alpha_{k+1} f(x_{k+1}) + \nabla f(x_{k+1})^\top (v - x_{k+1}) \right) + \lambda C(v) \right\} \\ &\overset{(a)}{\leq} \frac{1}{A_{k+1}} m_{k+1}(v_{k+1}) \leq f(x^*), \end{aligned} \tag{A.3.56}$$

where (a) comes from the weak duality of Lagrangian formulation. According to the strong convexity of set Q, we proceed to obtain

$$\begin{aligned} g(\lambda) &\geq \min_v \left\{ \frac{1}{A_{k+1}} \left(m_k(v) + \alpha_{k+1} f(x_{k+1}) + \nabla f(x_{k+1})^\top (v - x_{k+1}) \right) \right. \\ &\quad \left. + \frac{\lambda}{A_{k+1}} \sum_{i=0}^{k+1} \alpha_i \left(C(x_i) + \nabla C(x_i)^\top (v - x_i) + \frac{\mu_c}{2} \|v - x_i\|^2 \right) \right\} \\ &\triangleq \frac{1}{A_{k+1}} \min_v m_{k+1}^{f+\lambda C}(v). \end{aligned} \tag{A.3.57}$$

We denote the minimizer of $m_k^{f+\lambda C}(v)$ by $\underline{v}_k = \operatorname{argmin}_v m_k^{f+\lambda C}(v)$, and naturally define a lower bound for $f(x^*)$: $L_k := 1/A_k \min_v m_k^{f+\lambda C}(v)$. By KKT optimality condition we have

$$0 = \nabla_v m_{k+1}^{f+\lambda C}(\underline{v}_{k+1}) = \sum_{i=0}^{k+1} \alpha_i \nabla f(x_i) + \lambda \sum_{i=0}^{k+1} \alpha_i \left(\nabla C(x_i) + \mu_c (\underline{v}_{k+1} - x_i) \right). \tag{A.3.58}$$

Hence

$$\underline{v}_{k+1} = -\frac{1}{\lambda \mu_c A_{k+1}} \sum_{i=0}^{k+1} \alpha_i \left(\nabla f(x_i) + \lambda \left(\nabla C(x_i) - \mu_c x_i \right) \right), \tag{A.3.59}$$

and

$$A_{k+1}\underline{v}_{k+1} = A_k\underline{v}_k - \frac{1}{\lambda\mu_c}\alpha_{k+1}\left(\nabla f(x_{k+1}) + \lambda\left(\nabla C(x_{k+1}) - \mu_c x_{k+1}\right)\right). \tag{A.3.60}$$

By combining (A.3.56) and (A.3.57) we have

$$\begin{aligned} m_{k+1}(v_{k+1}) &\geq m_{k+1}^{f+\lambda C}(\underline{v}_{k+1}) = m_k^{f+\lambda C}(\underline{v}_{k+1}) + \alpha_{k+1}\left(f(x_{k+1}) + \lambda C(x_{k+1})\right) \\ &\quad + \alpha_{k+1}\left(\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\right)^\top(\underline{v}_{k+1} - x_{k+1}) + \frac{\lambda\mu_c\alpha_{k+1}}{2}\|\underline{v}_{k+1} - x_{k+1}\|^2 \end{aligned} \tag{A.3.61}$$

$$\begin{aligned} &\geq m_k^{f+\lambda C}(\underline{v}_k) + \frac{\lambda\mu_c A_k}{2}\|\underline{v}_{k+1} - \underline{v}_k\|^2 + \alpha_{k+1}\left(f(x_{k+1}) + \lambda C(x_{k+1})\right) \\ &\quad + \alpha_{k+1}\left(\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\right)^\top(\underline{v}_{k+1} - x_{k+1}) + \frac{\lambda\mu_c\alpha_{k+1}}{2}\|\underline{v}_{k+1} - x_{k+1}\|^2, \end{aligned} \tag{A.3.62}$$

where (A.3.62) is due to the fact that $m_k^{f+\lambda C}(v)$ is a quadratic function minimized at $\underline{v}_k$ with a total weight of quadratic terms being $\lambda\mu A_k/2$. By Jensen's inequality for the quadratic terms in the right-hand side of (A.3.62) and (A.3.60) we obtain

$$\begin{aligned} &m_{k+1}^{f+\lambda C}(\underline{v}_{k+1}) - m_k^{f+\lambda C}(\underline{v}_k) \\ &\geq \frac{\lambda\mu_c A_{k+1}}{2}\left\|\underline{v}_{k+1} - \frac{A_k}{A_{k+1}}\underline{v}_k - \frac{\alpha_{k+1}}{A_{k+1}}x_{k+1}\right\|^2 \\ &\quad + \alpha_{k+1}\left(f(x_{k+1}) + \lambda C(x_{k+1}) + \left(\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\right)^\top(\underline{v}_{k+1} - x_{k+1})\right) \qquad \text{(A.3.63)} \\ &= -\frac{\alpha_{k+1}^2}{2\lambda\mu_c A_{k+1}}\|\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\|^2 \\ &\quad + \alpha_{k+1}\left(f(x_{k+1}) + \lambda C(x_{k+1}) + \frac{A_k}{A_{k+1}}\left(\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\right)^\top(\underline{v}_k - x_{k+1})\right). \end{aligned} \tag{A.3.64}$$

Therefore, we have the following for the improvement of the lower bounds for any $k \geq 0$:

$$\begin{aligned} A_{k+1}L_{k+1} - A_kL_k &= m_{k+1}^{f+\lambda C}(\underline{v}_{k+1}) - m_k^{f+\lambda C}(\underline{v}_k) \\ &\geq -\frac{\alpha_{k+1}^2}{2\lambda\mu_c A_{k+1}}\|\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\|^2 \\ &\quad + \alpha_{k+1}\left(f(x_{k+1}) + \lambda C(x_{k+1}) + \frac{A_k}{A_{k+1}}\left(\nabla f(x_{k+1}) + \lambda\nabla C(x_{k+1})\right)^\top(\underline{v}_k - x_{k+1})\right). \end{aligned} \tag{A.3.65}$$

A.4. A New Variant of Frank-Wolfe with Adaptive Step-sizes

In this paper, we design an "accelerated" method to solve the minimization problem

$$\min_x f(x) + h(x) \tag{A.4.1}$$

We assume the function f to be smooth and convex. This means, for all x, y in the domain of $f + h$,

The function h is convex, potentially non-smooth, for which we know its *dual function*,

$$h^*(d) = \max_{s \in \mathrm{dom}(h)} s^T d - h(s), \quad \nabla h^*(d) = \arg\max_{s \in \mathrm{dom}(h)} s^T d - h(s), \quad d \in \mathrm{dom}(h^*).$$

In usual cases, the function h is the *indicator function* of a constrained set $\mathcal{C}$. Indeed, the following problem,

$$\min_x f(x) \quad \text{s.t.} \;\; x \in \mathcal{C}, \tag{A.4.2}$$

can be formulated as (A.2.2) using the indication function

$$h(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{otherwise.} \end{cases}$$

Again, in the constrained case, the dual of h is strongly linked to the linear minimization oracle (LMO) of the constrained set, defined as

$$\mathrm{LMO}(d) = \arg\min_{s \in \mathcal{C}} s^T d.$$

Clearly, $\mathrm{LMO}(d) = \nabla h^*(-d)$ and $h^*(d) = \mathrm{LMO}(-d)^T d$. The output of the LMO, which corresponds to the gradient of the dual function h, is called the *Frank-Wolfe (FW) corner*.

We assume the function h^* to be *isotropically smooth and strongly convex*.

Definition A.4.1. Isotropic smoothness and isotropic strong convexity. The function $h^*(x)$ is isotropically smooth and strongly convex if the function is smooth and strongly convex for all inputs of unitary norm, i.e.,

$$\frac{\mu_h}{2}\|d_1 - d_2\| \leq \|\nabla h^*(d_1) - \nabla h^*(d_2)\| \leq \frac{L_h}{2}\|d_1 - d_2\|, \quad \|d_1\| = \|d_2\| = 1.$$

In the next section we give an intuition of this assumption, as well as some examples.

A.4.1. Directional Smoothness and Directional Strong Convexity

Directional smoothness and directional convexity play a central role in the design of the accelerated conditional gradient method.

We first recall the classical accelerated gradient descent (AGD) from [59],

$$
\begin{aligned}
x_{k+1} &= \arg\min_x f(y_k) + \nabla f(y_k)(x - y_k) + \frac{L_f}{2}\|x - y_k\|^2 + h(x), \\
v_{k+1} &= \arg\min_x \sum_{i=0}^{k} \alpha_i \Big(f(y_i) + \nabla f(y_i)(x - y_i) + h(x) \Big) + \frac{1}{2}\|x - x_0\|^2, \\
y_{k+1} &= \beta_{k+1} x_{x+1} + (1 - \beta_{k+1}) v_{k+1},
\end{aligned}
$$

where h is a convex, potentially non-smooth function, whose proximal operator is "simple", i.e., the two first steps can be computed easily. In the proof of convergence, the step in v_{k+1} can be written as

$$
v_{k+1} = \nabla \left(h + \frac{1}{2}\|x - x_0\|^2 \right)^* (d_{k+1}), \quad d_{k+1} = \sum_{i=0}^{k} \alpha_i \nabla f(y_i).
$$

Let $\tilde{h} = h + \frac{1}{2}\|x - x_0\|^2$. Adding a regularization to h makes $\tilde{h}$ strongly convex.

A classical result from convex optimization states that, if a function h is strongly convex, then it's dual is smooth. Thus the dual $\tilde{h}^*$ is a *smooth* version of h^*. This trick is called the smoothing technique, see for instance [58].

In the proof of AGD, an important step consist in analyzing how far v_{k+1} is from v_k by using the smoothness of $\tilde{h}^*$. Indeed, since $d_{k+1} = d_k + \alpha_k \nabla f(y_k)$,

$$
\|v_{k+1} - v_k\| = \|\tilde{h}^*(d_k + \alpha_k \nabla f(y_k)) - \tilde{h}^*(d_k)\| \leq \alpha_k L_{\tilde{h}^*} \|\nabla f(y_k)\|.
$$

In the case of the conditional gradient, we *cannot* use this smoothing technique, as we only have access to h^*. Assuming we can have equivalent smoothness of h, in the case of FW algorithm, we can project on the set $\mathcal{C}$.

This specific assumption overcome a principal limitation of the LMO: the output is independent of the norm of the input, i.e.,

$$
\forall \gamma > 0, \quad LMO(d) = LMO(\alpha d).
$$

In addition, corners are allowed in the ball-based definition.

A.4.2. A New Algorithm on Smooth and Strongly Convex Sets

We denote the iterate sequence by $\{x_k\}_{k=0}^{\infty}$ and the gradient query sequence by $\{y_k\}_{k=0}^{\infty}$. We assign to each gradient query iterate k an associated measure α_k, a resulted sequence of increasing scaling coefficients $\{A_k\}_{k=0}^{\infty}$ is define by $A_0 = 0, A_{k+1} = A_k + \alpha_{k+1}$ for any $k \geq 0$. Let d_k be the accumulated weighted gradients defined by $d_k = 0, d_{k+1} = d_k + \alpha_{k+1} \nabla f(y_{k+1})$.

Algorithm 5 Accelerated Frank-Wolfe Algorithm over strongly smooth and convex sets.

Require: A strongly convex and directional strongly smooth set $\mathcal{C} \subseteq \mathbb{R}^n$; Initial iterate x_0; Number of iterations T; Averaging weights $\{\alpha_k\}_{k=0}^{\infty}$; $y_0 = x_0 = v_0$.
$A_0 = \alpha_0 = 0, d_0 = 0$.
for $k = 0, \ldots, T-1$ **do**
  $\alpha_{k+1} = \frac{\mu_c \|d_k\| + \sqrt{\mu_c^2 \|d_k\|^2 + 4 L_f L_c^2 \mu_c A_k \|d_k\|}}{2 L_f L_c^2}$
  $y_{k+1} = \frac{A_k x_k + \alpha_{k+1} v_k}{A_k + \alpha_{k+1}}$
  $d_{k+1} = d_k + \alpha_{k+1} \nabla f(y_{k+1})$
  $v_{k+1} = \operatorname{argmin}_{v \in \mathcal{C}} d_{k+1}^{\top} v$
  $x_{k+1} = \frac{A_k x_k + \alpha_{k+1} v_{k+1}}{A_k + \alpha_{k+1}}$
  $A_{k+1} = A_k + \alpha_{k+1}$
end for
Output: x_T

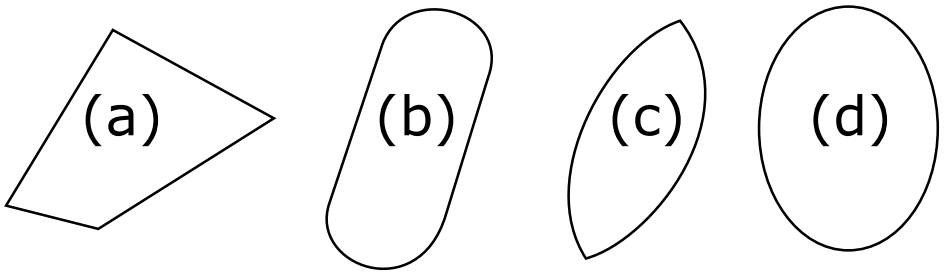


Fig. A.1. The set **(a)** is neither directional smooth and strongly convex, the set **(b)** is only directional strongly convex, **(c)** is only directional smooth. Finally, the set **(d)** is both directional smooth and strongly convex.

By a natural choice of the upper bound sequence $\{B_k\}_{k=0}^{\infty}$ and the lower bound sequence $\{b_k\}_{k=0}^{\infty}$, we obtain the following development on both upper and lower bounds.

$$A_{k+1}B_{k+1} - A_k B_k = A_{k+1} f(x_{k+1}) - A_k f(x_k), \tag{A.4.3}$$

$$\begin{aligned} A_{k+1}b_{k+1} - A_k b_k &= \sum_{i=0}^{k+1} \alpha_i \Big[f(y_i) + \nabla f(y_i)^{\top}(v_{k+1} - y_i) \Big] - \sum_{i=0}^{k} \alpha_i \Big[f(y_i) + \nabla f(y_i)^{\top}(v_k - y_i) \Big] \\ &= \alpha_{k+1} f(y_{k+!}) + \alpha_{k+1} \nabla f(y_k)^{\top}(v_{k+1} - y_{k+1}) + \sum_{i=0}^{k} \alpha_i \nabla f(y_i)^{\top}(v_{k+1} - v_k). \end{aligned} \tag{A.4.4}$$

Recall that f is convex, and the LMO of $\mathcal{C}$ can be computed efficiently, which means we have access to the *dual* of the indicator function, *i.e.*the support function, of the set $\mathcal{C}$,

$$\mathbf{I}_{\mathcal{C}}(x) = \begin{cases} \infty & \text{if } x \notin \mathcal{C} \\ 0 & \text{otherwise} \end{cases} \quad ; \quad \mathbf{I}_{\mathcal{C}}^{\star}(d) = \max_{x \in \mathcal{C}} d^T x. \tag{A.4.5}$$

To simplify the notations, we define the dual (support function) relative to a point $\mathbf{I}^{\star}_{\mathcal{C},\,x}$ and the FW mapping $\mathcal{M}$ as follow,

$$\mathbf{I}^{\star}_{\mathcal{C},\,x}(d) = \max_{s\in\mathcal{C}} d^T(s-x), \qquad \mathcal{M}_{\beta}(x,d) = (1-\beta)x + \beta \operatorname*{argmax}_{s\in\mathcal{C}} d^T s. \tag{A.4.6}$$

In this way, the Frank-Wolfe steps in Algorithm 5 can be written as

$$y_{k+1} = \mathcal{M}_{\alpha_{k+1}/A_{k+1}}(x_k, -d_k), \tag{A.4.7}$$

$$x_{k+1} = \mathcal{M}_{\alpha_{k+1}/A_{k+1}}(x_k, -d_{k+1}). \tag{A.4.8}$$

By defining

$$s^*(d) = \operatorname*{argmax}_{s\in\mathcal{C}} d^\top s, \tag{A.4.9}$$

we also obtain the following connection between the gradient of the support function and the Frank-Wolfe corner,

$$\nabla_d \mathbf{I}^{\star}_{\mathcal{C},\,x}(d) = s^*(d) - x. \tag{A.4.10}$$

Lemma A.4.2. For a μ_c-directionally strongly convex and L_c-directionally smooth set, we have

$$\frac{\mu_c\|d_k\|}{2}\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2 \leq \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1}-v_k) \leq \frac{L_c\|d_k\|}{2}\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2. \tag{A.4.11}$$

Démonstration. See Section A.4.5 for a detailed proof. ■

A.4.3. Theoretical Results and Analysis

In this section, we present a local rate of convergence when the optimum of the unconstrained objective is outside (on) the constraint set. We first impose some mild assumptions on the objective and the constraint set.

Assumption A.4.3. Suppose that for minimizer $x^* \in \mathcal{C}$, there exist a constant $g > 0$ such that $\nabla f(x^*) \geq g$.

In what follows, we show that our Algorithm results in a non-increasing duality gap sequence defined in Section A.2.

Lemma A.4.4. For all $k \geq 0$, the sequence of duality gaps generated by Algorithm 5 satisfies

$$A_{k+1}G_{k+1} \leq A_k G_k. \tag{A.4.12}$$

Démonstration. See Appendix A.4.6 for a detailed proof. ■

The rate of convergence of the algorithm is thus controlled by the rate of growth of A_k. The major difficulty is the dependence of α_k, thus A_k, to $\|d_k\|$. The next lemma shows that $\|d_k\|$ cannot be to small too often.

First, from the explicit formula of α_k, we have the valid lower-bound

$$\alpha_{k+1} \geq \max\left\{\frac{\mu_c\|d_k\|}{L_f L_c^2}, \sqrt{\frac{A_k\mu_c\|d_k\|}{L_f L_c^2}}\right\}$$

Now, we show that if at any moment, the norm of the direction d_k is too small, then $\|d_{k+1}\| \geq \|d_k\|$.

Lemma A.4.5. Assume the norm of the gradient is bounded below on the set, i.e.,

$$G \leq \|\nabla f(x)\| \quad \forall x \in \mathcal{C}.$$

If at any point we have

$$\sqrt{\|d_k\|} \leq \frac{1}{2}\sqrt{\frac{A_k\mu_c}{L_f L_c^2}}\|\nabla f(y_k)\|,$$

then for $t \geq k$ we have

$$\|d_{t+1}\| \geq \frac{1}{2}\sqrt{\frac{A_t\mu_c}{L_f L_c^2}}\|\nabla f(y_t)\| \cdot \sqrt{\|d_t\|}$$

until

$$\sqrt{\|d_t\|} \geq \frac{1}{2}\sqrt{\frac{A_t\mu_c}{L_f L_c^2}}\|\nabla f(y_t)\|.$$

Démonstration. See Appendix A.4.4 for a detailed proof. ■

A.4.4. Proof of Lemma A.4.5

Démonstration. By the triangle inequality,

$$\|d_{k+1}\| > \alpha_{k+1}\|\nabla f(y_k)\| - \|d_k\|.$$

Using the lower bound on α_k,

$$\|d_{k+1}\| \geq \sqrt{\frac{A_k\mu_c\|d_k\|}{L_f L_c^2}}\|\nabla f(y_k)\| - \|d_k\|.$$

Which is equal to

$$\|d_{k+1}\| \geq \sqrt{\|d_k\|}\left(\sqrt{\frac{A_k\mu_c}{L_f L_c^2}}\|\nabla f(y_k)\| - \sqrt{\|d_k\|}\right).$$

Asking the parenthesis to be at least equal to $\sqrt{\|d_k\|}$, i.e.,

$$\left(\sqrt{\frac{A_k\mu_c}{L_f L_c^2}}\|\nabla f(y_k)\| - \sqrt{\|d_k\|}\right) \geq \sqrt{\|d_k\|} \quad \Leftrightarrow \quad \frac{1}{2}\sqrt{\frac{A_k\mu_c}{L_f L_c^2}}\|\nabla f(y_k)\| \geq \sqrt{\|d_k\|},$$

which leads to $\|d_{k+1}\| \geq \|d_k\|$. ■

A.4.5. Proof of Lemma A.4.2

Démonstration. By the definition of the directional smoothness we have

$$\mathbf{I}^{\star}_{\mathcal{C},\, v_{k+1}}\left(-\frac{d_k}{\|d_k\|}\right) - \mathbf{I}^{\star}_{\mathcal{C},\, v_{k+1}}\left(-\frac{d_{k+1}}{\|d_{k+1}\|}\right) \geq \nabla \mathbf{I}^{\star}_{\mathcal{C},\, v_{k+1}}\left(-\frac{d_{k+1}}{\|d_{k+1}\|}\right)\left(-\frac{d_{k+1}}{\|d_{k+1}\|} + \frac{d_k}{\|d_k\|}\right) + \frac{\mu_c}{2}\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2, \tag{A.4.13}$$

By (A.4.10) and (A.4.6), we have

$$\nabla \mathbf{I}^{\star}_{\mathcal{C},\, v_{k+1}}\left(-\frac{d_{k+1}}{\|d_{k+1}\|}\right) = v_{k+1} - v_{k+1} = 0 \tag{A.4.14}$$

and

$$\mathbf{I}^{\star}_{\mathcal{C},\, v_{k+1}}\left(-\frac{d_{k+1}}{\|d_{k+1}\|}\right) = 0 \tag{A.4.15}$$

respectively. Plugging them into (A.4.13) we have

$$(v_{k+1} - v_k)^{\top}\frac{d_k}{\|d_k\|} \geq \frac{\mu_c}{2}\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2, \tag{A.4.16}$$

which implies the desired result. ■

A.4.6. Proof of Lemma A.4.4

Démonstration. By the definition of the duality gaps, lower bound and upper bound sequences, it follows that

$$
\begin{aligned}
A_{k+1}G_{k+1} - A_kG_k &= A_{k+1}\Big(f(x_{k+1}) - f(y_{k+1})\Big) + A_k\Big(f(y_{k+1}) - f(x_k)\Big) \\
&\quad - \alpha_{k+1}\nabla f(y_{k+1})^\top(v_{k+1} - y_{k+1}) - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k) \\
&\overset{(a)}{\leq} A_{k+1}\Big(f(x_{k+1}) - f(y_{k+1})\Big) + A_k\nabla f(y_{k+1})^\top(y_{k+1} - x_k) \\
&\quad - \alpha_{k+1}\nabla f(y_{k+1})^\top(v_{k+1} - y_{k+1}) - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k) \\
&\overset{(b)}{\leq} A_{k+1}\nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) + \frac{L_fA_{k+1}}{2}\|x_{k+1} - y_{k+1}\|^2 \\
&\quad + A_k\nabla f(y_{k+1})^\top(y_{k+1} - x_k) - \alpha_{k+1}\nabla f(y_{k+1})^\top(v_{k+1} - y_{k+1}) \\
&\quad - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k) \\
&= A_{k+1}\nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) + \frac{L_fA_{k+1}}{2}\|x_{k+1} - y_{k+1}\|^2 \\
&\quad + \nabla f(y_{k+1})^\top(A_{k+1}y_{k+1} - A_kx_k - \alpha_{k+1}v_k) - \alpha_{k+1}\nabla f(y_{k+1})^\top(v_{k+1} - v_k) \\
&\quad - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k) \\
&\overset{(c)}{=} A_{k+1}\nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) + \frac{L_fA_{k+1}}{2}\|x_{k+1} - y_{k+1}\|^2 \\
&\quad - \alpha_{k+1}\nabla f(y_k)^\top(v_{k+1} - v_k) - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k), \qquad \text{(A.4.17)}
\end{aligned}
$$

where (a) holds by the convexity of f, (b) follows from the L_f-smoothness of the objective f, (c) holds since we set $y_{k+1} = \dfrac{A_kx_k + \alpha_{k+1}v_k}{A_{k+1}}$ in each iteration.

Let $x_{k+1} = (1-\lambda_k)x_k + \lambda_k\Big(v_k + \theta_k(v_{k+1} - v_k)\Big)$ with $\lambda_k, \theta_k \in [0,1]$ for $k \geq 0$, then by plugging in the construction of y_{k+1} we have

$$
\begin{aligned}
&A_{k+1}\nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) \\
&\quad = \nabla f(y_{k+1})^\top\Big[A_{k+1}(1-\lambda_k)x_k + A_{k+1}\lambda_k(1-\theta_k)v_k + A_{k+1}\lambda_k\theta_kv_{k+1} - A_kx_k - \alpha_{k+1}v_k\Big]. \qquad \text{(A.4.18)}
\end{aligned}
$$

By setting $\lambda_k = \alpha_{k+1}/A_{k+1}$, we obtain

$$
A_{k+1}\nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) = \alpha_{k+1}\theta_k\nabla f(y_{k+1})^\top(v_{k+1} - v_k). \qquad \text{(A.4.19)}
$$

Similarly, we have in fact

$$x_{k+1} - y_{k+1} = \frac{\alpha_{k+1}}{A_{k+1}}\theta_k(v_{k+1} - v_k). \tag{A.4.20}$$

By plugging (A.4.20) into (A.4.17) we obtain

$$\begin{aligned} A_{k+1}G_{k+1} - A_kG_k \leq \alpha_{k+1}(\theta_k - 1)\nabla f(y_{k+1})^\top(v_{k+1} - v_k) + \frac{L_f\alpha_{k+1}^2\theta_k^2}{2A_{k+1}}\|v_{k+1} - v_k\|^2 \\ - \sum_{i=0}^{k}\alpha_i\nabla f(y_i)^\top(v_{k+1} - v_k). \end{aligned} \tag{A.4.21}$$

By setting $\theta_k = 1$, we have by Lemma A.4.2

$$A_{k+1}G_{k+1} - A_kG_k \leq \frac{L_f\alpha_{k+1}^2}{2A_{k+1}}\|v_{k+1} - v_k\|^2 - \frac{\mu_c\|d_k\|}{2}\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2 \tag{A.4.22}$$

$$\leq \left(\frac{L_fL_c^2\alpha_{k+1}^2}{2A_{k+1}} - \frac{\mu_c\|d_k\|}{2}\right)\left\|\frac{d_{k+1}}{\|d_{k+1}\|} - \frac{d_k}{\|d_k\|}\right\|^2, \tag{A.4.23}$$

where (A.4.23) holds by the L_c-directional smoothness of set $\mathcal{C}$. Hence when $L_fL_c^2\alpha_{k+1}^2 = \mu_cA_{k+1}\|d_k\|$, that is,

$$\alpha_{k+1} = \frac{\mu_c\|d_k\| + \sqrt{\mu_c^2\|d_k\|^2 + 4L_fL_c^2\mu_cA_k\|d_k\|}}{2L_fL_c^2}, \tag{A.4.24}$$

we are able to achieve

$$A_{k+1}G_{k+1} - A_kG_k \leq 0. \tag{A.4.25}$$

By the definition of duality gaps, for $k \geq 1$ we have

$$f(x_k) - f(x^*) \leq G_k \leq \frac{A_0G_0}{A_k}. \tag{A.4.26}$$

■

Annexe B

Supplemental Material for Chapter 2

B.1. Strong Convexity of Sets with Asymmetric Distance Functions

Before presenting the proof, we introduce the following results, extending known properties from smooth and strongly convex sets.

Proposition B.1.1. If f is strongly convex w.r.t. the distance function ω, then for $\gamma \in [0,1]$ we have

$$f(\gamma x + (1-\gamma)y) + \mu\gamma(1-\gamma)\frac{\gamma\omega^2(x-y) + (1-\gamma)\omega^2(y-x)}{2} \leq \gamma f(x) + (1-\gamma)f(y)$$

Démonstration. Let $z_\gamma = \gamma x + (1-\gamma)y$. We start with the definition,

$$f(z_\gamma) + \langle \nabla f(z_\gamma),\, x - z_\gamma \rangle + \frac{\mu}{2}\omega^2(x - z_\gamma) \leq f(x)$$
$$f(z_\gamma) + \langle \nabla f(z_\gamma),\, y - z_\gamma \rangle + \frac{\mu}{2}\omega^2(y - z_\gamma) \leq f(y)$$

After multiplying by γ and $1-\gamma$ and adding the two inequalities, we have

$$f(z_\gamma) + \mu\frac{\gamma\omega^2(x - z_\gamma) + (1-\gamma)\omega^2(y - z_\gamma)}{2} \leq \gamma f(x) + (1-\gamma)f(y)$$

Since $\omega^2(x - z_\gamma) = (1-\gamma)^2\omega^2(y-x)$, and $\omega^2(y - z_\gamma) = \gamma^2\omega^2(x-y)$, we obtain the desired result. ■

Proposition B.1.2. If f is convex and smooth w.r.t. the distance function ω, then it holds that

$$\frac{1}{2L}\omega_*^2\Big(\nabla f(x) - \nabla f(y)\Big) \leq f(y) - f(x) - \langle \nabla f(x),\, y - x \rangle$$

where ω_* is the dual of the function ω, written

$$\omega_*(v) \stackrel{\text{def}}{=} \max_{s:\omega(s)\leq 1} \langle v, s \rangle.$$

In particular, Proposition B.1.2 implies that, if f has a minimum $x_\star$, then

$$\frac{1}{2L}\omega_*^2\Big(-\nabla f(y)\Big) \leq f(y) - f(x_\star) \tag{B.1.1}$$

DÉMONSTRATION. Let the function $\phi(y) = f(y) - \langle \nabla f(x), y\rangle$. This function is, by construction, smooth. Moreover, $\min_y \phi(y)$ is attained when $y = x$. Since the function is smooth,

$$\min_y \phi(y) \leq \min_y \phi(z) + \langle \nabla\phi(z), y - z\rangle + \frac{L}{2}\omega^2(y - z)$$

Let $\beta u = y - z$, where $\omega(u) = 1$ and $\beta \geq 0$. Then,

$$\min_y \phi(y) \leq \min_{\beta,u} \phi(z) + \beta\langle \nabla\phi(z), u\rangle + \frac{\beta^2 L}{2}$$

The minimum can be split into two minimization problems,

$$\min_y \phi(y) \leq \phi(z) + \min_{\beta\geq 0}\left(\frac{\beta^2 L}{2} - \beta \max_{u:\omega(u)\leq 1}\langle -\nabla\phi(z), u\rangle\right).$$

By definition of the dual of ω,

$$\min_y \phi(y) \leq \phi(z) + \min_{\beta\geq 0}\left(\frac{\beta^2 L}{2} - \beta\omega_*\Big(-\nabla\phi(z)\Big)\right).$$

Now, we can solve over β, which gives us

$$\min_y \phi(y) \leq \phi(z) - \frac{1}{2L}\omega_*^2\Big(-\nabla\phi(z)\Big).$$

Replacing the minimum by $\phi(x)$, and ϕ by its expression, we get

$$f(x) - \langle \nabla f(x), x\rangle \leq f(z) - \langle \nabla f(x), z\rangle - \frac{1}{2L}\omega_*^2\Big(\nabla f(x) - \nabla f(z)\Big).$$

After reorganization, we get the desired result. ■

We can now show that level sets of a smooth and strong convex function are strongly convex sets, when they use the distance function ω.

DÉMONSTRATION. (**Proof of Lemma 1.3.5.**)

Consider the set

$$\mathcal{C} = \{x : f(x) - f_\star \leq R\}$$

Let $x, y \in \mathcal{C}$. Let $z_\gamma = \gamma x + (1-\gamma)y$, and consider the point $z_\gamma + u$. We have that

$$\begin{aligned} f(z_\gamma + u) - f_\star &\leq f(z_\gamma) - f_\star + \langle \nabla f(z_\gamma), u\rangle + \frac{L}{2}\omega^2(u), \\ &\leq f(z_\gamma) - f_\star + \omega(-u) \max_{v:\omega(v)\leq 1} \langle -\nabla f(z_\gamma), v\rangle + \frac{L}{2}\omega^2(u), \\ &= f(z_\gamma) - f_\star + \omega(-u)\omega_*\Big(-\nabla f(z_\gamma)\Big) + \frac{L}{2}\omega^2(u), \\ &\leq f(z_\gamma) - f_\star + \kappa_\omega \omega(u)\sqrt{2L(f(z_\gamma) - f_\star)} + \frac{L}{2}\omega^2(u). \end{aligned}$$

Therefore, to satisfy $f(z_\gamma + u) - f_\star \leq R$, we need to ensure that

$$\underbrace{f(z_\gamma) - f_\star - R}_{=\omega} + \underbrace{\kappa_\omega \sqrt{2L(f(z_\gamma) - f_\star)}}_{=\beta}\omega(u) + \frac{L}{2}\omega^2(u) \leq 0$$

Solving the problem in $\omega(u)$ gives

$$\omega(u) \leq \frac{-\beta + \sqrt{\beta^2 - 2L\omega}}{L}$$

We have that

$$\beta^2 - 2L\omega = 2L\left((f(z_\gamma) - f_\star)(\kappa_\omega^2 - 1) + R\right)$$

Therefore,

$$\omega(u) \leq \sqrt{2}\frac{-\kappa_\omega\sqrt{(f(z_\gamma) - f_\star)} + \sqrt{(f(z_\gamma) - f_\star)(\kappa_\omega^2 - 1) + R}}{\sqrt{L}}$$

However, since the function is strongly convex,

$$f(z_\gamma) - f_\star \leq \underbrace{\gamma f(x) + (1-\gamma)f(y) - f_\star}_{\leq R} - \mu\gamma(1-\gamma)\frac{\gamma\omega^2(x-y) + (1-\gamma)\omega^2(y-x)}{2}$$

Let $D_\gamma = \gamma(1-\gamma)\frac{\gamma\omega^2(x-y)+(1-\gamma)\omega^2(y-x)}{2}$. The inequality now reads

$$f(z_\gamma) - f_\star \leq R - \mu D_\gamma. \tag{B.1.2}$$

Therefore, the condition on ω becomes

$$\omega(u) \leq \sqrt{2}\frac{-\kappa_\omega\sqrt{R - \mu D_\gamma} + \sqrt{(R - \mu D_\gamma)(\kappa_\omega^2 - 1) + R}}{\sqrt{L}}$$

which gives

$$\omega(u) \leq \frac{\kappa_\omega\sqrt{2}}{\sqrt{L}}\left(-\sqrt{R - \mu D_\gamma} + \sqrt{R - \left(1 - \frac{1}{\kappa_\omega^2}\right)\mu D_\gamma}\right) \tag{B.1.3}$$

To simplify the expression in parenthesis, we multiply and divide by the conjugate of the square roots to get:

$$\left(-\sqrt{R-\mu D_\gamma}+\sqrt{R-\left(1-\frac{1}{\kappa_\omega^2}\right)\mu D_\gamma}\right)=\frac{R-\left(1-\frac{1}{\kappa_\omega^2}\right)\mu D_\gamma-(R-\mu D_\gamma)}{\sqrt{R-\mu D_\gamma}+\sqrt{R-\left(1-\frac{1}{\kappa_\omega^2}\right)\mu D_\gamma}}$$
$$\geq \frac{1}{\kappa_\omega^2 2\sqrt{R}}.$$

We can thus strengthen the condition (B.1.3) to:

$$\omega(u) \leq \frac{\mu D_\gamma}{\kappa_\omega\sqrt{2LR}}.$$

As the definition of a strongly convex set requires $\omega(u) \leq \alpha_\omega D_\gamma$, we conclude that the level set is strongly convex with at least the constant $\alpha_\omega = \frac{\mu}{\kappa_\omega\sqrt{2LR}}$. ■

B.1.1. Proof of Theorem 1.4.4

Theorem B.1.3. Consider the function f, smooth w.r.t. the distance function ω, with constant L_ω, and the set $\mathcal{C}$, strongly convex with constant α_ω.
Let $\delta(x) = x - v(x)$, $v(x)$ being the FW corner

$$v(x) \stackrel{\text{def}}{=} \operatorname*{argmin}_{v\in\mathcal{C}} \langle \nabla f(x), v\rangle.$$

Then, if $\omega_*(-\nabla f(x)) > c_\omega$ for all $x \in \mathcal{C}$, the function $f(x)$ is directionally smooth w.r.t. to ω, with constant

$$\mathcal{L}_{f,\delta} \leq \frac{L_\omega}{c_\omega\alpha_\omega}. \tag{B.1.4}$$

Démonstration. We start by the definition of smooth functions between x and $h\delta(x)$ for the distance function ω. We have for all $0 \leq h \leq 1$

$$f(x+h\delta(x)) \leq f(x) + h\langle\nabla f(x), \delta(x)\rangle + \frac{h^2L_\omega}{2}\omega^2(\delta(x))$$

Using the scaling inequality in (1.3.4),

$$\langle -\nabla f(x),\, \delta(x)\rangle \geq \alpha_\omega\omega_*\Big(-\nabla f(x)\Big)\omega(\delta(x))^2.$$

We hence obtain

$$f(x+h\delta(x)) \leq f(x) + h\langle\nabla f(x), \delta(x)\rangle - \frac{h^2L_\omega}{2}\frac{\langle\nabla f(x),\, \delta(x)\rangle}{\alpha_\omega\omega_*\Big(-\nabla f(x)\Big)}.$$

Since $\omega_*(-\nabla f(x)) > c_\omega$ for all $x \in \mathcal{C}$,

$$f(x+h\delta(x)) \leq f(x) + h\langle\nabla f(x), \delta(x)\rangle - \frac{h^2}{2}\frac{L_\omega}{\alpha_\omega c_\omega}\langle\nabla f(x),\, \delta(x)\rangle.$$

which is the definition of directional smoothness. ■

B.2. Missing proofs

B.2.1. Proof of Proposition 1.7.1

Proposition B.2.1. We define the "local Lipchitz constant" $L_{\text{loc}}(x)$, which satisfies

$$L_{\text{loc}}(x) \overset{\text{def}}{=} \mathcal{L}_{f,\delta} \frac{\langle -\nabla f(x), \delta(x) \rangle}{\|\delta(x)\|^2}.$$

Then, assuming that the local Lipchitz constant is "locally constant", the backtracking line-search finds $L_k \leq 2L_{\text{loc}}(x_k)$, and its stepsize $\gamma_\star$ satisfies

$$\min\left\{1, \frac{1}{2\mathcal{L}_{f,\delta}}\right\} \leq \gamma_\star.$$

Démonstration. We start with the definition of directional smoothness,

$$f(x + h\delta(x)) \leq f(x) + h\langle \nabla f(x), \delta(x) \rangle + [\mathcal{L}_{f,\delta} \langle -\nabla f(x), \delta(x) \rangle] \frac{h^2}{2}.$$

Writing $1 = \frac{\|\delta(x)\|_2^2}{\|\delta(x)\|_2^2}$, the upper bound becomes

$$f(x) + h\langle \nabla f(x), \delta(x) \rangle \quad + \left[\frac{\mathcal{L}_{f,\delta} \langle -\nabla f(x), \delta(x) \rangle}{\|\delta(x)\|_2^2}\right] \frac{h^2 \|\delta(x)\|_2^2}{2}.$$

Defining

$$L_{\text{loc}}(x) \triangleq \frac{\mathcal{L}_{f,\delta} \langle -\nabla f(x), \delta(x) \rangle}{\|\delta(x)\|_2^2},$$

we obtain

$$f(x_k + h\delta(x_k)) \leq f(x_k) + h\langle \nabla f(x_k), \delta(x_k) \rangle + L_{\text{loc}}(x_k) \frac{h^2 \|\delta(x_k)\|_2^2}{2}.$$

If we assume that $L_{\text{loc}}(x_k)$ is approximately constant, then Algorithm 6 finds $L_k \leq 2L_{\text{loc}}(x_k)$. Finally, using the definition of $\gamma_\star$ in Algorithm 6, we have

$$\begin{aligned} \gamma_\star &= \min\left\{\frac{-\nabla f(x_k)(v_k - x_k)}{L_{\text{loc}}(x_k)\|v_k - x_k\|^2}, 1\right\} \\ &\geq \min\left\{\frac{1}{2\mathcal{L}_{f,\delta}}, 1\right\}. \end{aligned}$$

■

B.2.2. Proof of Proposition 1.4.3

Proposition B.2.2 (Affine Invariance)**.** If $\delta(x)$ is affine covariant (e.g. the Frank-Wolfe direction $\delta(x) \triangleq v(x) - x$), then the constant $\mathcal{L}_{f,\delta}$ in (1.4.1) is affine invariant. In other words, let

$$\tilde{f}(\cdot) \triangleq f(B\cdot), \quad \tilde{\delta}_{\tilde{\mathcal{C}}}(\cdot) \triangleq \delta_{B\cdot\mathcal{C}}(\cdot),$$

then $\mathcal{L}_{\tilde{f},\tilde{\delta}_{\tilde{\mathcal{C}}}} = \mathcal{L}_{f,\delta}$.

DÉMONSTRATION. We start with the definition of directional smoothness, but with $x \to By$. The upper bound reads

$$f(By) + \left(h - \frac{\mathcal{L}_{f,\delta} h^2}{2}\right) \langle \nabla f(By), \delta(By) \rangle$$

Since we assumed $\delta(By)$ affine covariant,

$$\delta(By) = B\tilde{\delta}_{\tilde{\mathcal{C}}}(y).$$

Therefore,

$$f(By) + \left(h - \frac{\mathcal{L}_{f,\delta} h^2}{2}\right) \langle B^T \nabla f(By), \tilde{\delta}_{\tilde{\mathcal{C}}}(y) \rangle$$

Since $\nabla \tilde{f}(y) = B^T \nabla f(By)$, we have

$$\tilde{f}(\tilde{y} + h\tilde{\delta}_{\tilde{\mathcal{C}}}(y)) \leq \tilde{f}(y) + \left(h - \frac{\mathcal{L}_{f,\delta} h^2}{2}\right) \langle \nabla \tilde{f}(y), \tilde{\delta}_{\tilde{\mathcal{C}}}(y) \rangle$$

This means the function $\tilde{f}$ is directionally smooth with constant $\mathcal{L}_{f,\delta}$, which proves the statement. ■

B.3. Backtracking Line Search for Frank-Wolfe Steps

Algorithm 6 Backtracking line-search for smooth functions [63]

Input: FW corner v_k, point x_k, smoothness estimate L_k, function f.

1: Create the optimal stepsize and next iterate in the function of the Lipchitz estimate

$$\gamma_\star(L) \stackrel{\text{def}}{=} \min\left\{\frac{-\nabla f(x_k)(v_k - x_k)}{L\|v_k - x_k\|^2}, 1\right\}.$$
$$x(L) \stackrel{\text{def}}{=} (1 - \gamma_\star(L)) + \gamma_\star(L) v_k$$

2: Quadratic model of f between x_k and $x(L)$,

$$m(L) \stackrel{\text{def}}{=} f(x_k) + \langle \nabla f(x_k), x(L) - x_k \rangle + \frac{L}{2}\|x(L) - x_k\|^2$$

3: Set the current estimate $\tilde{L} \stackrel{\text{def}}{=} \frac{L_k}{2}$.

4: **while** $f(x(\tilde{L})) > m(\tilde{L})$ (Sufficient decrease not met because $\tilde{L}$ is too small) **do**

5: Double the estimate : $\tilde{L} \leftarrow 2 \cdot \tilde{L}$.

6: **end while**

Output: Estimate $L_{k+1} = \tilde{L}$, iterate $x_{k+1} = x(\tilde{L})$

B.4. Affine Invariant Analysis without Restriction on Optimum Location

In this section, we propose a modification of the directional smoothness defined in Section 1.4. This new assumption is the basis to obtain an affine invariant analysis of Frank-Wolfe on a strongly convex set without restriction on the position of the unconstrained optimum of f, as recently proposed in [31].

Outline. In Theorem B.4.2, we prove a $\mathcal{O}(1/K^2)$ sublinear convergence rate as in [31] when the function is *modified directionally smooth* (Definition B.4.1). In Theorem B.4.4, we prove that when $\mathcal{C}$ is strongly convex, and f is smooth and strongly convex, then f is *modified directionally smooth* for the Frank-Wolfe direction with an affine invariant constant leading to better conditioned convergence rates than in [31]. Finally, in Proposition B.4.5, we show that the constant of modified directional smoothness is affine invariant.

We now define a modification of directional smoothness. It is a structural assumption on f constrained on $\mathcal{C}$ designed at gathering the strong convexity of $\mathcal{C}$, the smoothness, and the strong convexity of f into a single quantity.

Definition B.4.1 (Modified Directional Smoothness)**.** Let $x_0 \in \mathcal{C}$. The function f is called *modified directionally smooth* with direction function $\delta : \mathcal{C} \to \mathbb{R}^N$ if there exists a constant $\tilde{\mathcal{L}}_{f,\delta}(x_0) > 0$ such that $\forall x \in \mathcal{C}$,

$$f\Big(x + h\delta(x)\Big) \leq f(x) + h\langle \nabla f(x), \delta(x)\rangle - \frac{\tilde{\mathcal{L}}_{f,\delta}(x_0)h^2}{2}\langle \nabla f(x), \delta(x)\rangle \sqrt{\frac{f(x_0) - f^*}{f(x) - f^*}}, \quad \text{(B.4.1)}$$

for $0 < h < 1$.

Note that the dependence of x_0 in the definition of the modified directional smoothness is an artifact to obtain a dimensionless constant $\tilde{\mathcal{L}}_{f,\delta}(x_0)$.

As in Section 1.5, the modified directional smoothness constant $\tilde{\mathcal{L}}_{f.\delta}$ is affine invariant in the case where δ is the FW direction. We now derive an affine invariant accelerated sublinear rate of convergence of Frank-Wolfe providing an affine invariant analysis of [31].

Theorem B.4.2 (Affine Invariant Accelerated Sublinear Rates)**.** Let $x_0 \in \mathcal{C}$ and assume f is a convex function and modified directionally smooth with direction function δ and constant $\tilde{\mathcal{L}}_{f,\delta}(x_0)$. Then, the iterates x_k for the Frank-Wolfe Algorithm 1 with stepsize

$$h_{\text{opt}} = \min\left\{1,\ \tfrac{1}{\tilde{\mathcal{L}}_{f,\delta}(x_0)}\sqrt{\tfrac{f(x_k)-f^*}{f(x_0)-f^*}}\right\}, \quad \text{with } \delta = v(x) - x,$$

or with exact line-search, where $v(x)$ is the Frank-Wolfe corner

$$v(x) = \operatorname*{argmin}_{v\in\mathcal{C}} \langle \nabla f(x), v\rangle,$$

satisfy

$$f(x_k) - f^* \leq \frac{4(f(x_0) - f^*)\max\{1,\, 18\tilde{\mathcal{L}}_{f,\delta}^2(x_0)\}}{(k+2)^2} \quad \text{for } k \geq 0.$$

Démonstration. The proof is similar to that of Theorem 1.5.1. We hence start with the modified directional smoothness assumption on f. For $0 < h < 1$,

$$f\Big(x_{k+1}\Big) \leq f(x_k) + \left(h - \frac{\tilde{\mathcal{L}}_{f,\delta} h^2}{2}\sqrt{\frac{f(x_0) - f^*}{f(x_k) - f^*}}\right)\langle \nabla f(x_k), \delta(x_k)\rangle \tag{B.4.2}$$

After minimizing over h, we have two possibilities. The case with exact line-search follows immediately after these two cases. In the following, we use the notation $h_k \overset{\text{def}}{=} f(x_k) - f^*$ for the primal suboptimality at x_k, and $g_k \overset{\text{def}}{=} \langle -\nabla f(x_k), \delta(x_k)\rangle$ for the Frank-Wolfe gap at x_k (and note that $g_k \geq h_k$ by convexity).

Case 1: $h_{\text{opt}} = \frac{1}{\tilde{\mathcal{L}}_{f,\delta}(x_0)}\sqrt{\frac{f(x_k)-f^*}{f(x_0)-f^*}}$. In such case, we obtain (subtract f^* on both sides of the inequality)

$$h_{k+1} \leq h_k - \frac{1}{2\tilde{\mathcal{L}}_{f,\delta}}\sqrt{\frac{h_k}{h_0}}g_k,$$

and since the Frank-Wolfe gap g_k upper bounds the primal suboptimality, we obtain

$$h_{k+1} \leq h_k\Big[1 - \frac{1}{2\tilde{\mathcal{L}}_{f,\delta}\sqrt{h_0}}\sqrt{h_k}\Big].$$

Case 2: With $h_{\text{opt}} = 1$, we have

$$h_{k+1} \leq h_k + \left(1 - \frac{\mathcal{L}_{f,\delta}}{2}\sqrt{\frac{h_0}{h_k}}\right)g_k.$$

In that case, we have that $\frac{1}{\tilde{\mathcal{L}}_{f,\delta}(x_0)}\sqrt{\frac{h_k}{h_0}} \geq 1$. Hence we obtain

$$h_{k+1} \leq h_k - \tfrac{1}{2}g_k \leq \tfrac{1}{2}h_k$$

Finally, we have the following recursive relation on the sequence of primal suboptimality (h_k):

$$\begin{aligned} h_{k+1} &\leq h_k \cdot \max\left\{\frac{1}{2},\, 1 - \frac{1}{2\tilde{\mathcal{L}}_{f,\delta}\sqrt{h_0}}\sqrt{h_k}\right\} \\ &= h_k \cdot \max\left\{\frac{1}{2},\, 1 - M\sqrt{h_k}\right\}, \end{aligned} \tag{B.4.3}$$

with $M \overset{\text{def}}{=} \frac{1}{2\tilde{\mathcal{L}}_{f,\delta}(x_0)\sqrt{h_0}}$. The inequality (B.4.3) is exactly the same recurrence that was analyzed by [31] (see their Equation (7), with the same notation for M), where they have shown a $\mathcal{O}(1/K^2)$ convergence rate. The exact constant is obtained by following the very same proof as [31], *i.e.* proving by induction that there exists C such that $h_k \leq C/(k+2)^2$.

The base case $k = 0$ can be trivially obtained by letting $C \geq 4h_0$.[1] Their induction step was shown by requiring that $C \geq \frac{18}{M^2}$. Thus using $C = \max\{4h_0, \frac{18}{M^2}\}$ (and re-arranging) proves the statement of our theorem. ■

The following lemma will be used in the proof of the bound on the modified directional smoothness.

Lemma B.4.3. Consider a compact convex set $\mathcal{C}$. Assume f is a μ_ω-strongly convex function with respect to ω. Let x^* be the minimum of f on $\mathcal{C}$. Then, for any $x \in \mathcal{C}$, we have

$$\omega_*(\nabla f(x)) \geq \sqrt{\frac{\mu_\omega}{2}}\sqrt{f(x) - f(x^*)}. \tag{B.4.4}$$

Démonstration. Let $x \in \mathcal{C}$. From Definition 1.3.3, we have that

$$f(x) \geq f(x^*) + \langle \nabla f(x^*),\, x - x^* \rangle + \frac{\mu_\omega}{2}\omega^2(x - x^\star).$$

Hence with the optimality conditions, *i.e.* $\langle \nabla f(x^*),\, x - x^* \rangle \geq 0$, we have

$$f(x) - f(x^*) \geq \frac{\mu_\omega}{2}\omega^2(x - x^*). \tag{B.4.5}$$

By convexity of f, we have $\langle x - x^*,\, \nabla f(x) \rangle \geq f(x) - f(x^*)$, and by definition of the Fenchel conjugate, we have

$$\omega(x - x^*) \cdot \omega_*(\nabla f(x)) \geq \langle x - x^*,\, \nabla f(x) \rangle \geq f(x) - f(x^*).$$

Hence by plugging (B.4.5), we obtain (B.4.4). ■

We now prove Theorem B.4.4 that is similar to Theorem 1.4.4. It states that in the case of the FW algorithm, the modified directional smoothness constant is bounded if the function is smooth, strongly convex and the set is strongly convex for any distance function ω. It also provides an explicit upper bound on the modified directional smoothness constant. This bound implies that the convergence rate in Theorem B.4.2 is better conditioned than existing results [31].

Theorem B.4.4 (Bounds on modified directional smoothness)**.** Consider $x_0 \in \mathcal{C}$ and a function f, smooth w.r.t. the distance function ω, with constant L_ω, strongly convex w.r.t. the distance function ω, with constant μ_ω, and the set $\mathcal{C}$, strongly convex with constant α_ω. Let $\delta(x) = x - v(x)$, $v(x)$ being the FW corner. Then, the function $f(x)$ is modified directionally smooth w.r.t. to δ, with constant

$$\tilde{\mathcal{L}}_{f,\delta}(x_0) \leq \frac{\kappa_\omega\sqrt{2}L_\omega}{\alpha_\omega\sqrt{\mu_\omega}}\frac{1}{\sqrt{f(x_0) - f^*}}. \tag{B.4.6}$$

[1]Note that [31] use a different argument for the base case, bounding instead h_1 with $L \cdot \mathrm{diam}(\mathcal{C})^2/2$, using the Lipschitz smoothness of f (and this would become $C_f/2$ in its affine invariant formulation with C_f as defined by [43]). However, we believe that h_0 is usually smaller than C_f in applications, and in any case h_0 appears from $1/M^2$ for us, so using our different base case argument is more meaningful.

DÉMONSTRATION. Let $h \in [0,1]$. With the smoothness of f, we have

$$f(x + h\delta(x)) \leq f(x) - h\langle -\nabla f(x),\, \delta(x)\rangle + \frac{h^2 L_\omega}{2}\omega\Big(\delta(x)\Big)^2.$$

Recall that when $\delta(x)$ is the Frank-Wolfe direction, we have that the Frank-Wolfe gap $g(x)$ is equal to $\langle -\nabla f(x),\, \delta(x)\rangle$. Also, the scaling inequality for strongly convex sets (Lemma 1.3.6) implies that $\omega(\delta(x))^2 \leq g(x)/(\alpha_\omega \omega^\star(-\nabla f(x)))$, so that

$$f(x + h\delta(x)) \leq f(x) - h\langle -\nabla f(x),\, \delta(x)\rangle + \frac{h^2 L_\omega}{2\alpha_\omega}\frac{g(x)}{\omega^\star(-\nabla f(x))}.$$

Now, it is easy to see from the definition of the dual distance ω_* that is has the same bounded asymmetry constant as for ω, and thus $\omega^\star(-\nabla f(x)) \geq \frac{1}{\kappa_\omega}\omega^\star(\nabla f(x))$. Thus we apply (B.4.4) to obtain:

$$f(x + h\delta(x)) \leq f(x) - hg(x) + \frac{h^2}{2}\frac{\kappa_w\sqrt{2}L_\omega}{\alpha_\omega\sqrt{\mu_\omega}\sqrt{f(x_0) - f^*}}\frac{\sqrt{f(x_0) - f^*}}{\sqrt{f(x) - f^*}}g(x),$$

which implies equation (B.4.6). ■

Theorem B.4.4 shows that the conditioning of convergence with the directional smoothness, which does not depend on any norm choice, in Theorem B.4.2 is better than conditioning of other analysis [31]. We now prove that the optimal constant of modified directional smoothness $\tilde{L}_{f,\delta}$ is affine invariant, a result similar to Proposition 1.4.3 for the directional smoothness constant.

Proposition B.4.5 (Affine Invariance of Modified Directional Smoothness)**.** Consider $\mathcal{C}$ a compact convex set and f a convex function on $\mathcal{C}$ that is modified directionally smooth w.r.t. $\delta(x)$ with constant $\tilde{\mathcal{L}}_{f,\delta}(x_0)$ (with $x_0 \in \mathcal{C}$). If for any $x \in \mathcal{C}$, $\delta(x)$ is affine covariant (e.g. the Frank-Wolfe direction $\delta(x) \triangleq v(x) - x$), then the constant $\tilde{\mathcal{L}}_{f,\delta}$ in (B.4.1) is affine invariant. In other words, for an invertible matrix B, let

$$\tilde{f}(\cdot) \triangleq f(B\cdot), \quad \tilde{\delta}_{\tilde{\mathcal{C}}}(\cdot) \triangleq \delta_{B^{-1}\cdot\mathcal{C}}(\cdot),$$

then $\tilde{\mathcal{L}}_{\tilde{f},\tilde{\delta}_{\tilde{\mathcal{C}}}}(x_0) = \tilde{\mathcal{L}}_{f,\delta}(y_0)$, where $y_0 \triangleq B^{-1}x_0$.

DÉMONSTRATION. Let $y \in B^{-1}\cdot\mathcal{C}$. Applying the definition of directional smoothness for f at By, we obtain

$$f\Big(By + h\delta(By)\Big) \leq f(By) + h\langle\nabla f(By), \delta(By)\rangle - \frac{\tilde{\mathcal{L}}_{f,\delta}(x_0)h^2}{2}\langle\nabla f(By), \delta(By)\rangle\sqrt{\frac{f(x_0) - f^*}{f(By) - f^*}}. \tag{B.4.7}$$

Similarly to Proposition 1.4.3, we have that $\nabla\tilde{f}(y) = B^T\nabla f(By)$ and $\delta(By) = B\tilde{\delta}_{\tilde{\mathcal{C}}}(y)$ so that

$$\langle\nabla f(By), \delta(By)\rangle = \langle\nabla f(By), B\tilde{\delta}_{\tilde{\mathcal{C}}}(y)\rangle = \langle B^T\nabla f(By), \tilde{\delta}_{\tilde{\mathcal{C}}}(y)\rangle = \langle\nabla\tilde{f}(y),\, \tilde{\delta}_{\tilde{\mathcal{C}}}(y)\rangle.$$

Hence (B.4.7) and $\tilde{f}^* = f^*$, implies that for any $y \in B^{-1} \cdot \mathcal{C}$

$$\tilde{f}(y + h\tilde{\delta}_{\tilde{\mathcal{C}}}) \leq \tilde{f}(y) + h\langle \nabla \tilde{f}(y),\, \tilde{\delta}_{\tilde{\mathcal{C}}}(y)\rangle - \frac{\tilde{\mathcal{L}}_{f,\delta}(x_0)h^2}{2}\langle \nabla \tilde{f}(y),\, \tilde{\delta}_{\tilde{\mathcal{C}}}(y)\rangle \sqrt{\frac{\tilde{f}(y_0) - \tilde{f}^*}{\tilde{f}(y) - \tilde{f}^*}}.$$

Hence, $\tilde{f}$ is modified directionally smooth on $\tilde{\mathcal{C}} \triangleq B^{-1} \cdot \mathcal{C}$ with respect to $\tilde{\delta}_{\tilde{\mathcal{C}}}$ and $\tilde{L}_{\tilde{f},\tilde{\delta}_{\tilde{\mathcal{C}}}}(y_0) \leq \tilde{\mathcal{L}}_{f,\delta}(x_0)$. A similar reasoning concludes that the two constants are equal. ■

B.5. Related Work Details

[48] propose an affine invariant analysis of the vanilla Frank-Wolfe algorithm when the unconstrained optimum x^* is in the relative interior of the constraint set $\mathcal{C}$ and f is strongly convex. Hence, the analysis applies when the constraint set is a strongly convex set, and the quantity might be defined in our context. However, the affine invariant constant $\mu_f^{(FW)}$ standing for the strong convexity of f is zero whenever the optimum is not in the relative interior of the constraint set $\mathcal{C}$. Indeed, Equation (3) from [48] define the following affine invariant quantity

$$\mu_f^{(FW)} \triangleq \inf_{\substack{x \in \mathcal{C} \setminus \{x^*\}, \gamma \in]0,1] \\ \bar{s} = \bar{s}(x, x^*, \mathcal{C}) \\ y = x + \gamma(\bar{s} - x)}} \frac{2}{\gamma^2}\Big[f(y) - f(x) - \langle \nabla f(x), y - x\rangle\Big],$$

where $\bar{s}(x, x^*, \mathcal{C}) = \mathrm{ray}(x, x^*) \cap \partial\mathcal{C}$. When $x^* \notin \mathcal{C}$, we have $\mu_f^{(FW)} \leq 0$ since there are some point $x \in \partial\mathcal{C}$ such that $x \in \bar{s}(x, x^*, \mathcal{C})$, and thus we can take $\bar{s} = x$ in the inf, yielding $y = x$ with $\gamma > 0$. This means that the above quantity cannot be easily generalized to the setting we studied in Theorem 1.4.4 where the unconstrained optimum is assumed to be *outside* of $\mathcal{C}$.

Annexe C

Supplemental Material for Chapter 3

C.1. Robust Symmetric Multisecant Algorithms

Algorithm 7 Type-I Symmetric Multisecant step

Input: Function f and gradient ∇f, initial approximation of the Hessian $\mathbf{B}_{\text{ref}}$, maximum memory m (can be ∞), relative regularization parameter $\bar{\lambda}$.

1: Compute $g_0 = \nabla f(x_0)$ and perform the initial step $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{B}_{\text{ref}}^{-1}\boldsymbol{g}_0$

2: **for** $t = 1,2,\ldots$ **do**

3: Form the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ using the m last pairs $(x_i,\nabla f(x_i))$.

4: Compute the qN direction $\mathbf{d}$ as $\mathbf{d}_t = -\mathbf{B}^{-1}g_t$, where

$$
\begin{aligned}
\mathbf{B}^{-1} &= \mathbf{E}\left(\mathbf{Z}_1 - \mathbf{Z}_2\mathbf{B}_{\text{ref}}^{-1}\mathbf{Z}_2^T\right)^{-1}\mathbf{E}^T + (\mathbf{I}-\mathbf{P})\mathbf{B}_{\text{ref}}^{-1}(\mathbf{I}-\mathbf{P}),\\
[\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}_1] &= \mathbf{SVD}(\Delta\mathbf{X}, \texttt{'econ'})\\
\mathbf{Z}_1 &= \mathbf{S}\odot\left[\mathbf{V}_1^T\left(\Delta\mathbf{X}\Delta\mathbf{G}^T + \Delta\mathbf{G}\Delta\mathbf{X}^T + \lambda\mathbf{B}_{\text{ref}}\right)\mathbf{V}_1\right],\\
\mathbf{S} &= \frac{1}{\boldsymbol{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\boldsymbol{\Sigma}^2 + \lambda\mathbf{1}\mathbf{1}^T},\\
\mathbf{P} &= \mathbf{V}_1\mathbf{V}_1^T,\\
\mathbf{Z}_2 &= (\boldsymbol{\Sigma}^2+\lambda\mathbf{I})^{-1}\mathbf{V}_1^T(\Delta\mathbf{X}\Delta\mathbf{G}^T + \lambda\mathbf{Z}_{\text{ref}})(\mathbf{I}-\mathbf{P})\\
\mathbf{E} &= \mathbf{V}_1 - (\mathbf{I}-\mathbf{P})\mathbf{Z}_{\text{ref}}^{-1}\mathbf{Z}_2^T.
\end{aligned}
$$

5: Perform an approximate-line search: $\mathbf{x}_{t+1} = \mathbf{x}_t + h_t\mathbf{d}_t, \quad h_t \approx \operatorname{argmin}_h f\left(\mathbf{x}_t + h_t\mathbf{d}_t\right)$.

6: **end for**

Algorithm 8 Type-II Symmetric Multisecant step

Input: Function f and gradient ∇f, initial approximation of the Hessian $\mathbf{H}_{\text{ref}}$, maximum memory m (can be ∞), relative regularization parameter $\bar{\lambda}$.

1: Compute $g_0 = \nabla f(x_0)$ and perform the initial step $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{H}_{\text{ref}}\boldsymbol{g}_0$

2: **for** $t = 1,2,\ldots$ **do**

3: Form the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ using the m last pairs $(x_i,\nabla f(x_i))$.

4: Compute the qN direction $\mathbf{d}$ as $\mathbf{d}_t = -\mathbf{H}^{-1}g_t$, where

$$\begin{aligned}
\mathbf{H} &= \mathbf{V}_1\mathbf{Z}_1\mathbf{V}_1^T + \mathbf{V}_1\mathbf{Z}_2 + \mathbf{Z}_2^T\mathbf{V}_1^T + (\mathbf{I}-\mathbf{P})\mathbf{H}_{\text{ref}}(\mathbf{I}-\mathbf{P}),\\
[\mathbf{U},\Sigma,\mathbf{V}_1] &= \mathbf{SVD}(\Delta\mathbf{G}^T, \texttt{'econ'}),\\
\mathbf{Z}_1 &= \mathbf{S}\odot\left[\mathbf{V}_1^T\left(\Delta\mathbf{G}\Delta\mathbf{X}^T + \Delta\mathbf{X}\Delta\mathbf{G}^T + \lambda\mathbf{H}_{\text{ref}}\right)\mathbf{V}_1\right],\\
\mathbf{S} &= \frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2 + \lambda\mathbf{1}\mathbf{1}^T},\\
\mathbf{P} &= \mathbf{V}_1\mathbf{V}_1^T,\\
\mathbf{Z}_2 &= (\Sigma^2+\lambda\mathbf{I})^{-1}\mathbf{V}_1^T(\Delta\mathbf{G}\Delta\mathbf{X}^T + \lambda\mathbf{Z}_{\text{ref}})(\mathbf{I}-\mathbf{P})
\end{aligned}$$

5: Perform an approximate-line search: $\mathbf{x}_{t+1} = \mathbf{x}_t + h_t\mathbf{d}_t, \quad h_t \approx \operatorname{argmin}_h f\left(\mathbf{x}_t + h_t\mathbf{d}_t\right)$.

6: **end for**

C.2. Positive Definite Estimates

C.2.1. Schur Complement and Robust Projection

We quickly discuss here a strategy to make the estimate $\mathbf{H}$ or $\mathbf{B}^{-1}$ positive definite. If we rewrite $\mathbf{Z}$ from Theorem 2.4.1, we have

$$\mathbf{Z}_\star = \underset{\mathbf{Z}=\mathbf{Z}^T}{\operatorname{argmin}} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|_F^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|_F^2,$$

where the matrices $\mathbf{Z}_2$, $\mathbf{Z}_{\text{ref}}$, $\mathbf{V}_1$ are defined in 2.4.1, and the matrix $\mathbf{P} = \mathbf{V}_1\mathbf{V}_1^T$ is a projector. Let $\mathbf{V}_2$ be the orthonormal complement of $\mathbf{V}_1$, i.e., $\mathbf{I} - \mathbf{P} = \mathbf{V}_2\mathbf{V}_2^T$. We can write $\mathbf{Z}_\star$ as follow,

$$\mathbf{Z}_\star = \begin{bmatrix}\mathbf{V}_1|\mathbf{V}_2\end{bmatrix} \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2\mathbf{V}_2 \\ \mathbf{V}_2^T\mathbf{Z}_2^T & \mathbf{V}_2^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2 \end{bmatrix} \begin{bmatrix}\mathbf{V}_1|\mathbf{V}_2\end{bmatrix}^T$$

By the Schur complement, the matrix is positive semi-definite if and only if

$$\mathbf{V}_2^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2 \succeq 0 \quad \text{and} \quad \mathbf{Z}_1 - (\mathbf{Z}_2\mathbf{V}_2)(\mathbf{V}_2^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2)(\mathbf{Z}_2\mathbf{V}_2)^T \succeq 0$$

Since $\mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}$, and because we start with a positive definite $\mathbf{Z}_{\text{ref}}$, the only condition is $\mathbf{Z}_1 \succeq \mathbf{Z}_2\mathbf{Z}_{\text{ref}}\mathbf{Z}_2^T$. The matrix $\mathbf{Z}_1$ is small ($m \times m$) and symmetric, therefore the projection of its eigenvalues to ensure the positive definiteness is cheap.

To project the matrix, let the variable $\boldsymbol{\chi}$ and $\boldsymbol{\chi}_0 = \mathbf{Z}_1 - \mathbf{Z}_2\mathbf{Z}_{\text{ref}}\mathbf{Z}_2^T$. We have to solve

$$\min_{\boldsymbol{\chi}} \|\boldsymbol{\chi} - \boldsymbol{\chi}_0\|_F \quad s.t. \;\; \boldsymbol{\chi} \succeq \sigma\mathbf{I}.$$

This way, we ensure that $\mathbf{Z} \succeq \sigma$. Let $\mathbf{U}\Lambda\mathbf{U}^T$ the eigenvalue decomposition of $\boldsymbol{\chi}_0$. the solution $\boldsymbol{\chi}_\star$ reads

$$\boldsymbol{\chi}_\star = \mathbf{U}\max\{\Lambda, \sigma\mathbf{I}\}\mathbf{U}^T \quad \text{(maximum element-wise)}.$$

We retrieve the modified matrix $\mathbf{Z}_1^+$ as

$$\mathbf{Z}_1^\sigma = \boldsymbol{\chi}_\star + \mathbf{Z}_2\mathbf{Z}_{\text{ref}}\mathbf{Z}_2^T.$$

We call this projection "robust" as we project the matrix s.t. the eigenvalues of $\mathbf{Z}$ are strictly positive, if $\sigma > 0$.

C.2.2. Robust Positive Definite Type-I Multisecant Update

We propose here a Robust version of the Multisecant Type-I update. The major stability problem in the Type-I update is the lack of guarantee that the eigenvalues of $\mathbf{Z}$ (i.e., $\mathbf{B}$) are away from zero. This means, when we will invert $\mathbf{Z}$, the eigenvalues of the matrix can be arbitrarily large. On the other side, large eigenvalues of $\mathbf{Z}$ are not a problem, since after inversion they will be very close to zero. That means we do not need to compute a regularized version of $\mathbf{Z}$, i.e., we do not need to set $\lambda > 0$ to compute $\mathbf{Z}$.

All together, we propose the following strategy: We compute all required matrices to form $\mathbf{Z}_\star^{-1}$, but can replace the matrix $\mathbf{Z}_1$ by $\mathbf{Z}_1^\sigma$. This controls the norm of $\mathbf{Z}_\star^{-1}$, and ensure its positive definiteness. We let the detailed analysis of the robustness of the method for future work.

C.2.3. Robust Positive Definite Type-II Multisecant Update

Here, the idea is simpler. As we already have the robustness property, it suffice to use the matrix $\mathbf{Z}_1^\sigma$ directly in the update formula of $\mathbf{Z}_\star$. Again, we let the detailed analysis of this method for future work.

C.3. Preconditioned Updates

We discuss in this section several strategies for the choice of the preconditioner $\mathbf{W}$, presented in Section 2.3.4. We present here the example for the Type-II method, but everything also applies to the Type-I. We recall that the preconditioner matrix $\mathbf{W}$ is an estimate of the Hessian, and is applies as follow,

$$\mathbf{M} = \mathbf{W}^{\alpha}\mathbf{H}\mathbf{W}^{(1-\alpha)}$$

Then, we solve the problem with $\mathbf{W}^{-\alpha}\Delta\mathbf{X}$ instead of $\Delta\mathbf{X}$, and with $\mathbf{W}^{(1-\alpha)}\Delta\mathbf{G}$ instead of $\Delta\mathbf{G}$. The estimate $\mathbf{H}$ is then recovered by solving $\mathbf{H} = \mathbf{W}^{-\alpha}\mathbf{M}\mathbf{W}^{(\alpha-1)}$.

C.3.1. Last estimate

Since we have computed all matrices $\mathbf{Z}_1, \mathbf{Z}_2, \ldots$ for form $\mathbf{H}_{k-1}$, it is easy to form $\mathbf{W} = \mathbf{H}_{k-1}$ and $\mathbf{W}^{-1} = \mathbf{H}_{k-1}^{-1}$ to create $\mathbf{H}_k$, given Theorem 2.4.1. Since we only have access to $\mathbf{H}$ or $\mathbf{H}^{-1}$, we have to set $\alpha = 1$ or $\alpha = 0$.

C.3.2. Successive Preconditioning

As before, we can use the information stored in the secant equation to compute the preconditioner $\mathbf{W}$. However, instead of using the previous secant equation, we use the current ones. We have two possibilities here: we can either use the Type-I approximation to compute $\mathbf{W}$, or the type-II, then compute $\mathbf{H}$ with this preconditioner. For each of these possibilities, we can use $\mathbf{W}$ on the left, or the right of $\mathbf{H}$. At the end, we have 4 possibilities:

$$\begin{aligned}
\mathbf{W} &= \texttt{Procrustes}(\Delta\mathbf{X}, \Delta\mathbf{G}, \mathbf{H}_{\text{ref}}), \\
\mathbf{H} &= \texttt{Procrustes}(\mathbf{W}^{-1}\Delta\mathbf{G}, \Delta\mathbf{X}, \mathbf{H}_{\text{ref}}\mathbf{W})\mathbf{W} && \text{(Type-I, } \alpha = 0\text{)}, \\
\mathbf{H} &= (\mathbf{W})^{-1}\texttt{Procrustes}(\Delta\mathbf{G}, \mathbf{W}\Delta\mathbf{X}, \mathbf{W}\mathbf{H}_{\text{ref}}) && \text{(Type-I, } \alpha = 1\text{)}, \\
\mathbf{W}^{-1} &= \texttt{InvProcrustes}(\Delta\mathbf{G}, \Delta\mathbf{X}, \mathbf{H}_{\text{ref}}^{-1}), \\
\mathbf{H} &= \texttt{Procrustes}(\mathbf{W}^{-1}\Delta\mathbf{G}, \Delta\mathbf{X}, \mathbf{H}_{\text{ref}}\mathbf{W})\mathbf{W} && \text{(Type-II, } \alpha = 0\text{)}, \\
\mathbf{H} &= (\mathbf{W})^{-1}\texttt{Procrustes}(\Delta\mathbf{G}, \mathbf{W}\Delta\mathbf{X}, \mathbf{W}\mathbf{H}_{\text{ref}}) && \text{(Type-II, } \alpha = 1\text{)}.
\end{aligned}$$

In fact, we can iteratively compute several $\mathbf{W}$ (since the SVD is already computed, it's only a matter of matrix-vector multiplications). We give here the example of the Type-I, $\alpha = 0$ preconditioner,

$$\mathbf{W}_i = \texttt{Procrustes}(\mathbf{W}_{i-1}^{-1}\Delta\mathbf{X}, \Delta\mathbf{G}, \mathbf{H}_{\text{ref}}\mathbf{W}_{i-1})\mathbf{W}_{i-1} \text{ or } \mathbf{W}_i = \mathbf{W}_{i-1}^{-1}\texttt{Procrustes}(\Delta\mathbf{X}, \mathbf{W}_{i-1}\Delta\mathbf{G}, \mathbf{W}_{i-1}\mathbf{H}_{\text{ref}})$$

We do not know if this process is convergent, or if it is useful to do several iteration to find the preconditioner. We let these investigations as future work.

C.3.3. Semi-Implicit Preconditioning

We discuss here a semi-implicit strategy, inspired by the preconditioner of BFGS and DFP. Indeed, we assume that there exist a matrix $\mathbf{W}$ such that

$$\mathbf{W}\Delta\mathbf{X} = \Delta\mathbf{G}.$$

In such case, we have 4 possibilities for the preconditioned secant equations,

$$\begin{aligned}(\mathbf{WH})\Delta\mathbf{G} &= \mathbf{W}\Delta\mathbf{X},\\ (\mathbf{HW})\mathbf{W}^{-1}\Delta\mathbf{G} &= \Delta\mathbf{X},\\ (\mathbf{W}^{-1}\mathbf{B})\Delta\mathbf{X} &= \mathbf{W}^{-1}\Delta\mathbf{G},\\ (\mathbf{BW}^{-1})\mathbf{W}\Delta\mathbf{X} &= \Delta\mathbf{G},\end{aligned}$$

which gives, if we use the implicit property of $\mathbf{W}$,

$$\begin{aligned}(\mathbf{WH})\Delta\mathbf{G} &= \Delta\mathbf{G},\\ (\mathbf{HW})\Delta\mathbf{X} &= \Delta\mathbf{X},\\ (\mathbf{W}^{-1}\mathbf{B})\Delta\mathbf{X} &= \Delta\mathbf{X}.\\ (\mathbf{BW}^{-1})\Delta\mathbf{G} &= \Delta\mathbf{G},\end{aligned}$$

We give here the example when $\mathbf{W}$ multiplies the secant equation on the left. We left the full study for future work.

Theorem C.3.1. The solution of the Type-II semi-implicit preconditioned update is given by

$$\min_{\mathbf{H}=\mathbf{H}^T} \|\mathbf{W}(\mathbf{H}-\mathbf{H}_{\text{ref}})\| \quad \text{s.t.} \quad \mathbf{WH}\Delta\mathbf{G} = \Delta\mathbf{G} \tag{C.3.1}$$

where $\Delta\mathbf{G}$ is a full column-rank matrix and $\mathbf{H}_{\text{ref}}$ a symmetric matrix is given by

$$\mathbf{H} = \mathbf{W}^{-1}\Delta\mathbf{G}\mathbf{T}_1^{-1}\Delta\mathbf{G}^T\mathbf{W}^{-1} + (\mathbf{I}-\mathbf{P}_1)^T\mathbf{H}_{\text{ref}}(\mathbf{I}-\mathbf{P}_1) \tag{C.3.2}$$

where

$$\mathbf{T}_1 = \Delta\mathbf{G}^T\mathbf{W}^{-1}\Delta\mathbf{G}, \qquad \text{and} \quad \mathbf{P}_1 = \Delta\mathbf{G}\mathbf{T}_1^{-1}\Delta\mathbf{G}^T\mathbf{W}^{-1} \text{ is a projector.}$$

The Type-I solves instead

$$\min_{\mathbf{B}=\mathbf{B}^T} \|\mathbf{W}^{-1}(\mathbf{B}-\mathbf{B}_{\text{ref}})\| \quad \text{s.t.} \quad \mathbf{W}^{-1}\mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{X},$$

whose inverse reads

$$\mathbf{B}^{-1} = \Delta\mathbf{X}\mathbf{T}_2^{-1}\Delta\mathbf{X}^T + \mathbf{B}_{\text{ref}}^{-1} - \mathbf{B}_{\text{ref}}^{-1}\mathbf{W}\Delta\mathbf{X}(\Delta\mathbf{X}^T\mathbf{W}\mathbf{B}_{\text{ref}}^{-1}\mathbf{W}\Delta\mathbf{X})^{-1}\Delta\mathbf{X}^T\mathbf{W}\mathbf{B}_{\text{ref}}^{-1},$$

where

$$\mathbf{T}_2 = \Delta\mathbf{X}^T\mathbf{W}\Delta\mathbf{X}.$$

The major problem here is to obtain the matrix $\mathbf{W}$ or $\mathbf{W}^{-1}$, which can be approximated using one of the two techniques presented in the previous subsections. Moreover, it would be interesting to consider a robust version of the preconditioned update.

C.4. Generalized qN step

We describe here the generalized qN update (Algorithm 9) and qN step (Algorithm 10).

Algorithm 9 Generalized qN direction

Input: Matrices $\Delta\mathbf{G}$, $\Delta\mathbf{X}$, regularization λ, reference matrices $\mathbf{H}_{\text{ref}} = \mathbf{B}_{\text{ref}}^{-1}$, direction $\mathbf{w}$.
Parameters: Loss function $\mathcal{L}$, Regularization function $\mathcal{R}$, constraint set $\mathcal{C}$.

1: Solve the problem

$$\mathbf{B} = \operatorname*{argmin}_{\mathbf{B}\in\mathcal{C}} \mathcal{L}\left(\mathbf{B}\Delta\mathbf{X}, \Delta\mathbf{G}\right) + \frac{\lambda}{2}\mathcal{R}\left(\mathbf{B}, \mathbf{B}_{\text{ref}}\right) \qquad \text{(Type-I)}$$

$$\mathbf{H} = \operatorname*{argmin}_{\mathbf{H}\in\mathcal{C}} \mathcal{L}\left(\mathbf{H}\Delta\mathbf{G}, \Delta\mathbf{X}\right) + \frac{\lambda}{2}\mathcal{R}\left(\mathbf{H},\, \mathbf{H}_{\text{ref}}\right) \qquad \text{(Type-II)}$$

Output: qN direction $\mathbf{d} = \mathbf{B}^{-1}\mathbf{w}$ or $\mathbf{d} = \mathbf{H}\mathbf{w}$.

Algorithm 10 Generalized qN step

Input: Sequence of $m+1$ pairs iterates-gradient

$$\{(\mathbf{x}_0, \boldsymbol{g}_0), (\mathbf{x}_1, \boldsymbol{g}_1), \ldots, (\mathbf{x}_m, \boldsymbol{g}_m)\}, \qquad \text{where } \boldsymbol{g}_i = \nabla f(\mathbf{x}_i).$$

Parameters: Matrix of differences $\mathbf{C} \in \mathbb{R}^{m+1,m}$ of rank m, vector of coefficients $\boldsymbol{v} \in \mathbb{R}^{m+1}$, such that

$$\mathbf{1}_{m+1}^T\mathbf{C} = 0, \qquad \boldsymbol{v}^T\mathbf{1}_{n+1} = 1.$$

1: Form the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ as

$$\Delta\mathbf{X} = \mathbf{X}\mathbf{C}, \quad \Delta\mathbf{G} = \mathbf{G}\mathbf{C}.$$

2: Form the gradient direction $\mathbf{w}$ as

$$\mathbf{w} = \mathbf{G}\boldsymbol{v}$$

3: Call Algorithm 9 with $\Delta\mathbf{G}, \Delta\mathbf{X}$,$\mathbf{w}$ (and other parameters), and retrieve the qN direction $\mathbf{d}$.

4: Form the next iterate $\mathbf{x}_+$ using approximate line-search,

$$\mathbf{x}_+ = \mathbf{X}\boldsymbol{v} - h^*\mathbf{d}, \quad \text{where } h^* \approx \operatorname*{argmin}_h f(\mathbf{X}\boldsymbol{v} - h\mathbf{d}).$$

Algorithm 10 is inspired by the fact that, if $\mathbf{Q}$ is the true hessian such that

$$\mathbf{Q}^{-1}\mathbf{G} = \mathbf{X} - \mathbf{X}_\star, \quad \textit{where } \mathbf{X}_\star = \mathbf{x}_\star\mathbf{1}^T,$$

when, if $\mathbf{H} \approx \mathbf{Q}^{-1}$ (equivalently $\mathbf{B}0^{-1} \approx \mathbf{Q}^{-1}$), we have

$$\mathbf{X} - \mathbf{H}\mathbf{G} \approx \mathbf{X}_\star.$$

Multiplying both size by $\boldsymbol{v}$, where $\boldsymbol{v}^T\mathbf{1} = 1$, we have $\mathbf{X}_\star\boldsymbol{v} = \mathbf{x}_\star$ and

$$\underbrace{(\mathbf{X} - \mathbf{HG})\boldsymbol{v} = \mathbf{X}\boldsymbol{v} - \mathbf{Hw}}_{\text{Generalized qN step}} \approx \mathbf{x}_\star.$$

C.5. Convergence analysis on quadratics

We now analyze the convergence speed of the generalized qN step (Algorithm 10) when applied on a quadratic function.

C.5.1. Setting

Objective function. We consider the minimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{x}-\mathbf{x}_\star)^T\mathbf{Q}(\mathbf{x}-\mathbf{x}_\star)+f_\star. \tag{C.5.1}$$

Notice that C.5.1 is equivalent to $f(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}\mathbf{x} + b^T\mathbf{x} + c$, but the notation in (C.5.1) is more convenient. Since the function f is quadratic, we have the following relations,

$$\mathbf{Q}\Delta\mathbf{X} = \Delta\mathbf{G}, \quad \mathbf{Q}(\mathbf{X}-\mathbf{X}_\star) = \mathbf{G}. \tag{C.5.2}$$

Algorithm. We consider the algorithm

$$\mathbf{x}_{k+1} = (\mathbf{X}_k - \mathbf{H}_k\mathbf{G}_k)\boldsymbol{v}_k, \qquad \text{where} \quad \mathbf{X}_k = [x_0, \ldots, x_k], \quad \mathbf{G}_k = [g_0, \ldots, g_k], \quad \boldsymbol{v}_k : \boldsymbol{v}_k^T\mathbf{1}_{k+1} = 1, \tag{C.5.3}$$

and $\mathbf{H}_k$ is formed by Algorithm 9.

Assumptions. We assume

- The spectrum of the true Hessian $\mathbf{Q}$ is bounded by $\ell\mathbf{I} \preceq \mathbf{Q} \preceq L\mathbf{I}$, $0 < \ell < L$.
- (Simplifying assumption) We use only the notation $\mathbf{H}_k$ for the approximation of the inverse of the Hessian at the iteration k, in opposition to making the distinction between $\mathbf{H}_k$ and $\mathbf{B}_k^{-1}$.
- We assume that the qN approximation satisfies *exactly* the secant equations, i.e.,

$$\mathbf{H}_k\Delta\mathbf{G}_k = \Delta\mathbf{X}_k.$$

- The qN method is used with *full memory*, i.e., $\mathbf{X}_k$ contains all iterates from 0 to k and grows indefinitely.
- The matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are full column rank.

C.5.2. Generic formula of H

In the case where $\mathbf{H}$ satisfies exactly the secant equation, the generic formula of $\mathbf{H}$ reads

$$\mathbf{H}_k = \Delta\mathbf{X}_k\Delta\mathbf{G}_k^\dagger + \tilde{\Theta}_k(\mathbf{I}-\mathbf{P}_k), \quad \mathbf{P}_k = \Delta\mathbf{G}_k\Delta\mathbf{G}_k^\dagger, \tag{C.5.4}$$

where Θ_k is a matrix that depends on the initialization $\mathbf{H}_{\text{ref}}$, the constraints set $\mathcal{C}$ and the regularization function $\mathcal{R}$ (but not on the loss since $\mathbf{H}$ satisfies exactly the secant equations). The notation $\Delta\mathbf{G}_k^\dagger$ is any left pseudo-inverse of $\Delta\mathbf{G}$ that satisfies

$$\Delta\mathbf{G}_k^\dagger\Delta\mathbf{G}_k = \mathbf{I}_k,$$

which exists since $\Delta\mathbf{G}_k$ is full column rank. The matrix $\mathbf{P}$ is a projector such that $\mathbf{P}\Delta\mathbf{G} = \Delta\mathbf{G}$ and $\mathbf{P}^2 = \mathbf{P}$, which is *not* symmetric because it's not an orthonormal projection (unlike most projection matrices). Finally, the matrix $\tilde{\mathbf{H}}$ depends on the initialization and constraints of the qN method.

Indeed, if $\mathbf{H}_k$ satisfies (C.5.4), we have that $\mathbf{H}_k$ satisfies the secant equations since

$$\mathbf{H}\Delta\mathbf{G} = \Delta\mathbf{X}_k \underbrace{\Delta\mathbf{G}_k^\dagger \Delta\mathbf{G}}_{=\mathbf{I}} + \Theta_k \underbrace{(\mathbf{I} - \mathbf{P}_k)\Delta\mathbf{G}}_{=0} = \Delta\mathbf{X}.$$

C.5.3. Independence of v

We first show that the generalized qN step (C.5.3) is (surprisingly) *independent* of the choice of $\boldsymbol{v}$. We omit the subscript k in this section for simplicity.

Proposition C.5.1 (Invariance under $\boldsymbol{v}$)**.** Let $\tilde{\mathbf{x}}_+$ and $\mathbf{x}_+$ be formed by (C.5.3) using resp. $\tilde{\boldsymbol{v}}$ and $\boldsymbol{v}$. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

Démonstration. We first write the difference between $\mathbf{x}_+$ and $\tilde{\mathbf{x}}_+$,

$$\mathbf{x}_+ - \tilde{\mathbf{x}}_+ = (\mathbf{X} - \mathbf{H}\mathbf{G}) \underbrace{(\boldsymbol{v} - \tilde{\boldsymbol{v}})}_{\Delta\boldsymbol{v}}.$$

However, $\Delta\boldsymbol{v} = \boldsymbol{v} - \tilde{\boldsymbol{v}}$ is a vector that sum to 0. Since $\mathbf{C}$ is a matrix such that

$$\mathbf{1}^T\mathbf{C} = 0, \quad \mathbf{C} \text{ is full column rank},$$

this means $\mathbf{C}$ is a basis for all vectors that sum to zero. Therefore, there exists a vector of coefficients $\boldsymbol{\alpha}$ such that $\mathbf{C}\boldsymbol{\alpha} = \Delta\boldsymbol{v}$. Rewriting the difference, we obtain

$$\mathbf{x}_+ - \tilde{\mathbf{x}}_+ = (\mathbf{X} - \mathbf{H}\mathbf{G})\mathbf{C}\boldsymbol{\alpha}.$$

However, $\mathbf{G}\mathbf{C} = \Delta\mathbf{G}$ and $\mathbf{X}\mathbf{C} = \Delta\mathbf{X}$. Since $\mathbf{H}\Delta\mathbf{G} = \Delta\mathbf{X}$, the difference is zero, which prove the statement. ■

C.5.4. Krylov subspace structure of the iterates

Before proving the rate of convergence of the qN step, we show that the iterates follows a Krylov structure.

Proposition C.5.2. Assume that, for all $i = 0 \dots k$, we have

$$\mathbf{x}_i \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\nabla f(\mathbf{x}_0), \dots, \nabla f(\mathbf{x}_{i-1})\}.$$

In such case,

$$\mathbf{x}_i - \mathbf{x}_\star \in \mathbf{x}_0 - \mathbf{x}_\star + \text{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star), (\tilde{\mathbf{H}}\mathbf{Q})^2(\mathbf{x}_0 - \mathbf{x}_\star), \dots, (\tilde{\mathbf{H}}\mathbf{Q})^{i-1}(\mathbf{x}_0 - \mathbf{x}_\star)\}$$

Démonstration. We prove the result iteratively. For $i = 0$, we have

$$\mathbf{x}_0 - \mathbf{x}_\star = \mathbf{I}(\mathbf{x}_0 - \mathbf{x}_\star).$$

For $i = 1$,

$$\mathbf{x}_1 - \mathbf{x}_\star \in \mathbf{x}_0 - \mathbf{x}_\star + \tilde{\mathbf{H}}\text{span}\{\nabla f(\mathbf{x}_0)\}$$

Since $\nabla f(\mathbf{x}_0) = \mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star)$,

$$\mathbf{x}_1 - \mathbf{x}_\star \in \mathbf{x}_0 - \mathbf{x}_\star + \tilde{\mathbf{H}}\text{span}\{\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star)\} \in \mathbf{x}_0 - \mathbf{x}_\star + \text{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star)\}.$$

For $i = 2$,

$$\begin{aligned}
\mathbf{x}_1 - \mathbf{x}_\star &\in \mathbf{x}_0 - \mathbf{x}_\star + \tilde{\mathbf{H}}\text{span}\{\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star),\, \mathbf{Q}(\mathbf{x}_1 - \mathbf{x}_\star)\} \\
&\in \mathbf{x}_0 - \mathbf{x}_\star + \tilde{\mathbf{H}}\text{span}\{\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star),\, \mathbf{Q}\left(\mathbf{x}_0 - \mathbf{x}_\star + \text{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star\})\right)\} \\
&\in \mathbf{x}_0 - \mathbf{x}_\star + \tilde{\mathbf{H}}\text{span}\{\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star),\, \mathbf{Q}\left(\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star)\right)\} \\
&\in \mathbf{x}_0 - \mathbf{x}_\star + \text{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star),\, \tilde{\mathbf{H}}\mathbf{Q}\left(\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star)\right)\} \\
&\in \mathbf{x}_0 - \mathbf{x}_\star + \text{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star),\, (\tilde{\mathbf{H}}\mathbf{Q})^2(\mathbf{x}_0 - \mathbf{x}_\star)\}
\end{aligned}$$

We can repeat the process up to i. ■

C.5.5. Rate of convergence

We now analyse the rate of convergence of algorithm (C.5.3) in term of the distance to the solution.

Theorem C.5.3. Assume that, for all $i = 0 \dots k$, we have

$$\mathbf{x}_i \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\nabla f(x_0), \dots, \nabla f(x_{i-1})\}.$$

Moreover, assume that

$$\tilde{\mathbf{H}}\mathbf{Q} \text{ is psd, and } \kappa = \frac{\|\tilde{\mathbf{H}}\mathbf{Q}\|}{\|(\tilde{\mathbf{H}}\mathbf{Q})^{-1}\|} \text{ is bounded.}$$

In such case, the accuracy of the $k - th$ qN step is bounded by

$$\|\mathbf{x}_k - \mathbf{x}_\star\| \leq \|\mathbf{I} - \mathbf{H}_k\mathbf{Q}\| \left(\frac{1 - \sqrt{\kappa^{-1}}}{1 + \sqrt{\kappa^{-1}}}\right)^k \|\mathbf{x}_0 - \mathbf{x}_\star\|$$

Démonstration. If we expand the expression, we obtain

$$\begin{aligned}
\mathbf{x}_{k+1} &= (\mathbf{X}_k - \mathbf{H}_k\mathbf{G}_k)\, \boldsymbol{v} - \mathbf{x}_\star, \\
&= (\mathbf{I} - \mathbf{H}_k\mathbf{G}_k)\, \boldsymbol{v}_k, \\
&= (\mathbf{I} - \mathbf{H}_k\mathbf{Q})\, (\mathbf{X}_k - \mathbf{X}_\star)\, \boldsymbol{v}_k. \qquad \text{(C.5.5)}
\end{aligned}$$

By Proposition C.5.1, we can take any $\boldsymbol{v}_k$ such that $\boldsymbol{v}^T\mathbf{1} = 1$. In particular, we chose $\boldsymbol{v}_k = \boldsymbol{v}_k^\star$ such that

$$\boldsymbol{v}_k^\star \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{v}:\boldsymbol{v}^T\mathbf{1}=1} \|\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}\|_2^2$$

Therefore,

$$\|\nabla f(\mathbf{x}_{k+1})\| \leq \|\mathbf{I} - \mathbf{H}_k\mathbf{Q}\|\|\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k\| = \|\mathbf{I} - \mathbf{H}_k\| \cdot \min_{\boldsymbol{v}:\boldsymbol{v}^T\mathbf{1}=1} \|\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}\|.$$

By definition of $(\mathbf{X}_k - \mathbf{X}_\star)$, we have

$$\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k = \left(\sum_{i=0}^{k} \boldsymbol{v}_i \left(\mathbf{x}_0 - \mathbf{x}_\star + \operatorname{span}\{\nabla f(\mathbf{x}_0), \ldots, \nabla f(\mathbf{x}_i)\}\right)\right).$$

Since $\boldsymbol{v}$ sum to one,

$$\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k = \mathbf{x}_0 - \mathbf{x}_\star + \left(\sum_{i=0}^{k} \boldsymbol{v}_i \operatorname{span}\{\nabla f(\mathbf{x}_0), \ldots, \nabla f(\mathbf{x}_i)\}\right).$$

By definition of a span,

$$\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k \in \mathbf{x}_0 - \mathbf{x}_\star + \operatorname{span}\{\nabla f(\mathbf{x}_0), \ldots, \nabla f(\mathbf{x}_i)\}.$$

By Proposition C.5.2,

$$\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k \in \mathbf{x}_0 - \mathbf{x}_\star + \operatorname{span}\{\tilde{\mathbf{H}}\mathbf{Q}(\mathbf{x}_0 - \mathbf{x}_\star), (\tilde{\mathbf{H}}\mathbf{Q})^2(\mathbf{x}_0 - \mathbf{x}_\star), \ldots, (\tilde{\mathbf{H}}\mathbf{Q})^{i-1}(\mathbf{x}_0 - \mathbf{x}_\star)\}.$$

Notice that, because $\mathbf{G}$ is full rank the span is a basis, therefore there is a one-to-one correspondence between the span and $\boldsymbol{v}_k$ (i.e., there exists a unique vector $\boldsymbol{v}_k$ such that $\boldsymbol{v}_k^T\mathbf{1} = 1$ such that $(\mathbf{X}_k - \mathbf{X}_\star)\boldsymbol{v}_k$ is a vector of the span). Using the definition of the span,

$$\left(\mathbf{X}_k - \mathbf{X}_\star\right)\boldsymbol{v}_k = \Pi_k(\tilde{\mathbf{H}}\mathbf{Q})(\mathbf{x}_0 - \mathbf{x}_\star), \quad \Pi_k \text{ is a polynomial of degree at most } k, \text{ such that } \Pi_k(0) = 1.$$

Therefore,

$$\|\nabla f(\mathbf{x}_{k+1})\| \leq \|\mathbf{I} - \mathbf{H}_k\mathbf{Q}\| \cdot \min_{\Pi:\deg(\Pi)\leq k,\, \Pi(0)=1} \left\|\Pi_k(\tilde{\mathbf{H}}\mathbf{Q})(\mathbf{x}_0 - \mathbf{x}_\star)\right\|$$

Now, assume that $\tilde{\mathbf{H}}\mathbf{Q}$ is symmetric, p.s.d., and let κ be its condition number, i.e.,

$$\kappa = \frac{\|\tilde{\mathbf{H}}\mathbf{Q}\|}{\|(\tilde{\mathbf{H}}\mathbf{Q})^{-1}\|}.$$

Then, standard result from Krylov subspace gives the bound

$$\min_{\Pi:\deg(\Pi)\leq k,\, \Pi(0)=1} \left\|\Pi_k(\tilde{\mathbf{H}}\mathbf{Q})(\mathbf{x}_0 - \mathbf{x}_\star)\right\| \leq \left(\frac{1 - \sqrt{\kappa^{-1}}}{1 + \sqrt{\kappa^{-1}}}\right)^k \|\mathbf{x}_0 - \mathbf{x}_\star\|,$$

for $k \leq d$, and converges exactly to 0 when $k \geq d$, which prove the statement. ■

C.5.6. Example of qN method satisfying the assumptions

We show here that standard qN method satisfies the assumptions of Theorem C.5.3. We first show a simpler condition for the method that ensure it satisfies the assumptions of Theorem C.5.3.

Proposition C.5.4. Let $\mathbf{H}$ be any matrix that satisfies the secant equation, which means

$$\mathbf{H} = \Delta\mathbf{X}\Delta\mathbf{G}^\dagger + \Theta(\mathbf{I} - \mathbf{P}), \quad \Delta\mathbf{G}^\dagger : \Delta\mathbf{X}\Delta\mathbf{G}^\dagger\Delta\mathbf{G} = \Delta\mathbf{X}, \quad \mathbf{P} : \mathbf{P}\Delta\mathbf{G} = \Delta\mathbf{G}.$$

If

$$\Theta(\mathbf{I} - \mathbf{P})\mathbf{G}\boldsymbol{v} \in \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\},$$

then $\mathbf{x}_+ \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}$. Moreover, if $\tilde{\mathbf{H}}$ is symmetric positive definite then the method satisfies the assumption of Theorem C.5.3.

Démonstration. We start by expanding the generalized qN step,

$$\begin{aligned}
\mathbf{x}_+ &= \mathbf{X}\boldsymbol{v} - \Delta\mathbf{X}\Delta\mathbf{G}^\dagger\mathbf{G}\boldsymbol{v} - \Theta(\mathbf{I} - \mathbf{P})\mathbf{G}\boldsymbol{v} \\
&= \mathbf{X}\underbrace{\left(\mathbf{I} - \mathbf{C}\Delta\mathbf{G}^\dagger\mathbf{G}\right)\boldsymbol{v}}_{=\mathbf{w}} - \Theta(\mathbf{I} - \mathbf{P})\mathbf{G}\boldsymbol{v} \\
&= \mathbf{X}\mathbf{w} - \Theta(\mathbf{I} - \mathbf{P})\mathbf{G}\boldsymbol{v}.
\end{aligned}$$

Notice that $\mathbf{1}^T\mathbf{w} = 1$, since

$$\mathbf{1}^T\mathbf{w} = \mathbf{1}^T\left(\mathbf{I} - \mathbf{C}\Delta\mathbf{G}^\dagger\mathbf{G}\right)\boldsymbol{v} = \underbrace{\mathbf{1}^T\boldsymbol{v}}_{=1} - \underbrace{\mathbf{1}^T\mathbf{C}}_{=0}\Delta\mathbf{G}^\dagger\mathbf{G}\boldsymbol{v}.$$

We now show the property recursively. The property is true at $\mathbf{x}_0$, and assume it's true up to k. Therefore,

$$\mathbf{X}\mathbf{w} = \mathbf{X}_k\mathbf{w}_k = \sum_{i=0}^{k}\mathbf{w}_i\mathbf{x}_i \in \underbrace{\sum_{i=0}^{k}\mathbf{w}_i}_{=1}\mathbf{x}_0 + \sum_{i=0}^{k}\mathbf{w}_i\tilde{\mathbf{H}}\text{span}\{\mathbf{G}_{i-1}\} \quad \text{(recursivity assumption)},$$

Which means $\mathbf{X}\mathbf{w} \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\mathbf{G}_{k-1}\}$. Therefore, if $\Theta(\mathbf{I} - \mathbf{P})\mathbf{G}\boldsymbol{v} \in \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}$, we have $\mathbf{x}_+ \in \mathbf{x}_0 + \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}$. ■

C.5.6.1. Multisecant Broyden Type-I.

TL;DR. The method satisfies Theorem C.5.3 if $\mathbf{B}_{\text{ref}}$ is symmetric positive definite.

The Multisecant Broyden Type-I reads

$$\mathbf{B}^{-1} = \mathbf{B}_0^{-1} + (\Delta\mathbf{X} - \mathbf{B}_0^{-1}\Delta\mathbf{G})(\Delta\mathbf{X}^T\mathbf{B}_0^{-1}\Delta\mathbf{G})^{-1}\Delta\mathbf{X}^T\mathbf{B}_0^{-1}$$

After reorganization,

$$\mathbf{B}^{-1} = \Delta\mathbf{X}\Delta\mathbf{G}^\dagger + \mathbf{B}_0^{-1}(\mathbf{I}\Delta\mathbf{G}\Delta\mathbf{G}^\dagger), \quad \Delta\mathbf{G}^\dagger = (\Delta\mathbf{X}^T\mathbf{B}_0^{-1}\Delta\mathbf{G})^{-1}\Delta\mathbf{X}^T\mathbf{B}_0^{-1}.$$

We clearly identity $\Theta(\mathbf{I}-\mathbf{P}) = \mathbf{B}_{\text{ref}}^{-1}(\mathbf{I}-\Delta\mathbf{G}\Delta\mathbf{G}^{\dagger})$. After expansion,

$$\begin{aligned}\Theta(\mathbf{I}-\mathbf{P})\mathbf{G}\boldsymbol{v} = \mathbf{B}_{\text{ref}}^{-1}(\mathbf{I}-\Delta\mathbf{G}\Delta\mathbf{G}^{\dagger})\mathbf{G}\boldsymbol{v} &= \mathbf{B}_{\text{ref}}^{-1}\mathbf{G}(\mathbf{I}-\mathbf{C}\Delta\mathbf{G}^{\dagger}\mathbf{G})\boldsymbol{v},\\ &= \mathbf{B}_{\text{ref}}^{-1}\mathbf{G}\tilde{\boldsymbol{v}},\\ &\in \mathbf{B}_{\text{ref}}^{-1}\text{span}\{\mathbf{G}\}.\end{aligned}$$

Defining $\tilde{\mathbf{H}} = \mathbf{B}_{\text{ref}}^{-1}$, we have $\Theta(\mathbf{I}-\mathbf{P})\mathbf{G}\boldsymbol{v} \in \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}$. If $\mathbf{H}_{\text{ref}}$ is full rank, symmetric and positive definite, then by Proposition C.5.4 the method satisfies Theorem C.5.3.

C.5.6.2. Multisecant Broyden Type-II.

TL;DR. The method satisfies Theorem C.5.3 if $\mathbf{H}_{\text{ref}}$ is symmetric positive definite.

The Multisecant Broyden Type-II update reads

$$\mathbf{H} = \Delta\mathbf{X}\Delta\mathbf{G}^{\dagger} + \mathbf{H}_{\text{ref}}(\mathbf{I}-\Delta\mathbf{G}\Delta\mathbf{G}^{\dagger}).$$

We clearly identity $\Theta(\mathbf{I}-\mathbf{P}) = \mathbf{H}_{\text{ref}}(\mathbf{I}-\Delta\mathbf{G}\Delta\mathbf{G}^{\dagger})$. After expansion,

$$\begin{aligned}\Theta(\mathbf{I}-\mathbf{P})\mathbf{G}\boldsymbol{v} = \mathbf{H}_{\text{ref}}(\mathbf{I}-\Delta\mathbf{G}\Delta\mathbf{G}^{\dagger})\mathbf{G}\boldsymbol{v} &= \mathbf{H}_{\text{ref}}\mathbf{G}(\mathbf{I}-\mathbf{C}\Delta\mathbf{G}^{\dagger}\mathbf{G})\boldsymbol{v},\\ &= \mathbf{H}_{\text{ref}}\mathbf{G}\tilde{\boldsymbol{v}},\\ &\in \mathbf{H}_{\text{ref}}\text{span}\{\mathbf{G}\}.\end{aligned}$$

Defining $\tilde{\mathbf{H}} = \mathbf{H}_{\text{ref}}$, we have $\Theta(\mathbf{I}-\mathbf{P})\mathbf{G}\boldsymbol{v} \in \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}$. If $\mathbf{H}_{\text{ref}}$ is full rank, symmetric and positive definite, then by Proposition C.5.4 the method satisfies Theorem C.5.3.

C.5.6.3. Multisecant BFGS for quadratics.

TL;DR. The method satisfies Theorem C.5.3 if $\mathbf{H}_{\text{ref}}$ is symmetric positive definite.

The multisecant BFGS for quadratics reads

$$\mathbf{H} = \Delta\mathbf{X}\Delta\mathbf{G}^{\dagger} + \Delta\mathbf{X}(\Delta\mathbf{G}^{\dagger})^{T}(\mathbf{I}-\mathbf{P}) + (\mathbf{I}-\mathbf{P})^{T}\mathbf{H}_{\text{ref}}(\mathbf{I}-\mathbf{P}), \quad \Delta\mathbf{G}^{\dagger} = (\Delta\mathbf{X}^{T}\Delta\mathbf{G})^{-1}\Delta\mathbf{X}^{T},$$

which is symmetric if and only if $\Delta\mathbf{X}^{T}\Delta\mathbf{G}$ is a symmetric matrix. Notice that this reduces to the standard BFGS update when $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are vectors. We identify $\Theta(\mathbf{I}-\mathbf{P})$ as

$$\Theta(\mathbf{I}-\mathbf{P}) = \left(\Delta\mathbf{X}(\Delta\mathbf{G}^{\dagger})^{T} + (\mathbf{I}-\mathbf{P})^{T}\mathbf{H}_{\text{ref}}\right)(\mathbf{I}-\mathbf{P}).$$

After expanding $\mathbf{P}$,

$$\Theta(\mathbf{I}-\mathbf{P}) = \left(\mathbf{H}_{\text{ref}} + \Delta\mathbf{X}\left((\Delta\mathbf{G}^{\dagger})^{T} - \Delta\mathbf{G}^{\dagger}\mathbf{H}_{\text{ref}})^{T}\right)\right)(\mathbf{I}-\mathbf{P}).$$

Since $\Delta\mathbf{X}$ already belong to the span, it suffices to show

$$\mathbf{H}_{\text{ref}}(\mathbf{I}-\mathbf{P})\mathbf{G}\boldsymbol{v} \in \tilde{\mathbf{H}}\text{span}\{\mathbf{G}\}.$$

Following the same technique as before, we have $\tilde{\mathbf{H}} = \mathbf{H}_{\text{ref}}$. Therefore, the methods satisfies the assumptions if $\mathbf{H}_{\text{ref}}$ is symmetric and positive definite.

C.6. Symmetric Procrustes Problem

Consider the following problem, known as Symmetric Procrustes.

Theorem C.6.1. Consider the Regularized Symmetric Procrustes (RSP) problem

$$\mathbf{Z}_\star = \underset{\mathbf{Z}=\mathbf{Z}^T}{\operatorname{argmin}} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|^2, \tag{RSP}$$

where $\mathbf{Z}_{\text{ref}}$ is symmetric (otherwise, take the symmetric part of $\mathbf{Z}_{\text{ref}}$), $\mathbf{Z}, \mathbf{Z}_{\text{ref}} \in \mathbb{R}^{d\times d}$, and $\mathbf{A}, \mathbf{D} \in \mathbb{R}^{d\times m}$, $m \leq d$, $\lambda > 0$. Then, the solution $\mathbf{Z}_\star$ is given by

$$\mathbf{Z}_\star = \mathbf{V}_1\mathbf{Z}_1\mathbf{V}_1^T + \mathbf{V}_1\mathbf{Z}_2 + \mathbf{Z}_2^T\mathbf{V}_1^T + (\mathbf{I}-\mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I}-\mathbf{P}) \tag{Sol-RSP}$$

where

$$\begin{aligned}
[\mathbf{U},\mathbf{\Sigma},\mathbf{V}_1] &= \mathbf{SVD}(\mathbf{A}^T, \texttt{'econ'}), \text{ (economic SVD)}\\
\mathbf{Z}_1 &= \mathbf{S} \odot \left[\mathbf{V}_1^T\left(\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T + \lambda\mathbf{Z}_{\text{ref}}\right)\mathbf{V}_1\right],\\
\mathbf{S} &= \frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + \lambda\mathbf{1}\mathbf{1}^T},\\
\mathbf{P} &= \mathbf{V}_1\mathbf{V}_1^T,\\
\mathbf{Z}_2 &= (\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}_1^T(\mathbf{A}\mathbf{D}^T + \lambda\mathbf{Z}_{\text{ref}})(\mathbf{I}-\mathbf{P})
\end{aligned}$$

The fraction in $\mathbf{S}$ stands for the element-wise inversion (Hadamard inverse). The inverse $\mathbf{Z}_\star^{-1}$ reads

$$\begin{aligned}
\mathbf{Z}_\star^{-1} &= \mathbf{E}\left(\mathbf{Z}_1 - \mathbf{Z}_2\mathbf{Z}_{\text{ref}}^{-1}\mathbf{Z}_2^T\right)^{-1}\mathbf{E}^T + (\mathbf{I}-\mathbf{P})\mathbf{Z}_{\text{ref}}^{-1}(\mathbf{I}-\mathbf{P})\\
\mathbf{E} &= \mathbf{V}_1 - (\mathbf{I}-\mathbf{P})\mathbf{Z}_{\text{ref}}^{-1}\mathbf{Z}_2^T.
\end{aligned} \tag{Inv-RSP}$$

Démonstration. We begin by deriving the solution of (RSP). By taking the transposition of the matrices inside the Frobenius norm of the first term in (RSP), we obtain the equivalent problem

$$\min_{\mathbf{Z}=\mathbf{Z}^T\in\mathbb{R}^{d\times d}} \|\mathbf{A}^T\mathbf{Z} - \mathbf{D}^T\|^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|_F^2. \tag{C.6.1}$$

We write the (full) singular value decomposition of $\mathbf{A}^T$ as

$$\mathbf{U}\begin{bmatrix}\mathbf{\Sigma} & 0\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{V}_1^T\\ \mathbf{V}_2^T\end{bmatrix}}_{=\mathbf{V}}, \tag{C.6.2}$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{V} \in \mathbb{R}^{d\times d}$ are orthogonal matrices, $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with nonnegative entries, and $\mathbf{V}_1 \in \mathbb{R}^{m\times d}, \mathbf{V}_2 \in \mathbb{R}^{d-m\times d}$. Thus, we obtain another problem

equivalent to (RSP), that reads

$$\min_{\tilde{\mathbf{Z}}=\tilde{\mathbf{Z}}^T\in\mathbb{R}^{d\times d}} \|[\mathbf{\Sigma},0]\tilde{\mathbf{Z}}-\tilde{\mathbf{D}}^T\|^2+\frac{\lambda}{2}\|\tilde{\mathbf{Z}}-\tilde{\mathbf{Z}_{\text{ref}}}\|_F^2, \tag{C.6.3}$$

$$\begin{aligned}\text{where}\quad &\tilde{\mathbf{Z}}=\mathbf{V}\mathbf{Z}\mathbf{V}^T,\\ &\tilde{\mathbf{D}}=\mathbf{U}^T\mathbf{D}^T\mathbf{V},\\ &\tilde{\mathbf{Z}_{\text{ref}}}=\mathbf{V}^T\mathbf{Z}_{\text{ref}}\mathbf{V}.\end{aligned}$$

Equation (C.6.3) is equivalent to (RSP) after multiplying the inside of the norm bu $\mathbf{U}^T$ on the left, and $\mathbf{V}$ on the right, since the Frobenius norm is invariant to orthonormal transformation. We now decompose the matrices in blocks as follow,

$$\tilde{\mathbf{Z}}=\begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}}_D\\ \tilde{\mathbf{Z}}_D^T & \tilde{\mathbf{Z}}_2\end{bmatrix}\qquad \tilde{\mathbf{D}}=\begin{bmatrix}\tilde{\mathbf{D}}_1 & \mathbf{D}_2\end{bmatrix}\qquad \tilde{\mathbf{Z}_{\text{ref}}}=\begin{bmatrix}(\tilde{\mathbf{Z}_{\text{ref}}})_1 & (\tilde{\mathbf{Z}_{\text{ref}}})_D\\ (\tilde{\mathbf{Z}_{\text{ref}}})_D^T & (\tilde{\mathbf{Z}_{\text{ref}}})_2\end{bmatrix} \tag{C.6.4}$$

where $\mathbf{Z}_1$, $(\tilde{\mathbf{Z}_{\text{ref}}})_1, \tilde{\mathbf{D}}_1 \in \mathbb{R}^{m\times m}$, $\mathbf{Z}_2$, $(\tilde{\mathbf{Z}_{\text{ref}}})_2 \in \mathbb{R}^{d-m\times d-m}$, $\mathbf{Z}_D$, $(\tilde{\mathbf{Z}_{\text{ref}}})_D$, $\mathbf{D}_2 \in \mathbb{R}^{m\times d-m}$. Hence, we can problem (C.6.3) as

$$\begin{aligned}&\min_{\tilde{\mathbf{Z}}=\tilde{\mathbf{Z}}^T\in\mathbb{R}^{d\times d}} \|[\mathbf{\Sigma},0]\tilde{\mathbf{Z}}-\tilde{\mathbf{D}}^T\|^2+\frac{\lambda}{2}\|\tilde{\mathbf{Z}}-\tilde{\mathbf{Z}_{\text{ref}}}\|_F^2,\\ &\quad=\min_{\tilde{\mathbf{Z}}_1=\tilde{\mathbf{Z}}_1^T,\tilde{\mathbf{Z}}_2=\tilde{\mathbf{Z}}_2^T,\mathbf{Z}_D} \|[\mathbf{\Sigma}\tilde{\mathbf{Z}}_1,\mathbf{\Sigma}\tilde{\mathbf{Z}}_D]-[\tilde{\mathbf{D}}_1,\tilde{\mathbf{D}}_2]\|^2\\ &\quad+\frac{\lambda}{2}\left(\|\tilde{\mathbf{Z}}_1-(\tilde{\mathbf{Z}_{\text{ref}}})_1\|^2+2\|\tilde{\mathbf{Z}}_D-(\tilde{\mathbf{Z}_{\text{ref}}})_D\|^2+\|\tilde{\mathbf{Z}}_2-(\tilde{\mathbf{Z}_{\text{ref}}})_2\|^2\right)\\ &\quad=\min_{\tilde{\mathbf{Z}}_1=\tilde{\mathbf{Z}}_1^T}\|\mathbf{\Sigma}\tilde{\mathbf{Z}}_1-\mathbf{D}_1\|^2+\frac{\lambda}{2}\|\tilde{\mathbf{Z}}_1-(\tilde{\mathbf{Z}_{\text{ref}}})_1\|^2 &\text{(i)}\\ &\qquad+\min_{\mathbf{Z}_D}\|\mathbf{\Sigma}\tilde{\mathbf{Z}}_D-\mathbf{D}_2\|^2+\lambda\|\tilde{\mathbf{Z}}_D-(\tilde{\mathbf{Z}_{\text{ref}}})_D\|^2 &\text{(ii)}\\ &\qquad+\min_{\tilde{\mathbf{Z}}_21=\tilde{\mathbf{Z}}_2^T}\frac{\lambda}{2}\|\tilde{\mathbf{Z}}_2-(\tilde{\mathbf{Z}_{\text{ref}}})_2\|^2 &\text{(iii)}\end{aligned}$$

Hence, we derive the solution to (RSP) by minimizing three independent terms as below.

Term (iii)**:** The term

$$\operatorname*{argmin}_{\tilde{\mathbf{Z}}_2=(\tilde{\mathbf{Z}}_2)^T}\frac{\lambda}{2}\|\tilde{\mathbf{Z}}_2-(\tilde{\mathbf{Z}_{\text{ref}}})_2\|^2$$

imposes the constraint $\tilde{\mathbf{Z}}_2=(\tilde{\mathbf{Z}_{\text{ref}}})_2$. In other words, we have

$$\tilde{\mathbf{Z}}_2=\mathbf{V}_2^T\mathbf{Z}_0\mathbf{V}_2. \tag{C.6.5}$$

Term (ii)**:** The term

$$\min_{\mathbf{Z}_D}\|\mathbf{\Sigma}\tilde{\mathbf{Z}}_D-\mathbf{D}_2\|^2+\lambda\|\tilde{\mathbf{Z}}_D-(\tilde{\mathbf{Z}_{\text{ref}}})_D\|^2$$

is a simple regularized least-square, which can be solved by setting the derivative to zero. Therefore,

$$\tilde{\mathbf{Z}}_D = (\mathbf{\Sigma}^T\mathbf{\Sigma} + \lambda\mathbf{I})^{-1}(\mathbf{D}_2 + \lambda(\tilde{\mathbf{Z}_{\text{ref}}})_D) \tag{C.6.6}$$

Term (i): In what follows, we solve the problem (similar to the one in [42])

$$\min_{\tilde{\mathbf{Z}}_1=(\tilde{\mathbf{Z}}_1)^T\in\mathbb{R}^{m\times m}} \|\mathbf{\Sigma}\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{D}}_1\|^2 + \lambda\|\tilde{\mathbf{Z}}_1 - (\tilde{\mathbf{Z}}_0)_1\|^2,$$

We first rewrite the optimization problems in terms of the entries in $\tilde{\mathbf{Z}}$ as below, using the fact that $\tilde{\mathbf{Z}}_1$ is symmetric,

$$\begin{aligned}\min_{\tilde{\mathbf{Z}}=\mathbf{Z}^T\in\mathbb{R}^{m\times m}} &\sum_{i=1}^{m}(\sigma_i(\tilde{\mathbf{Z}}_1)_{ii} - (\tilde{\mathbf{D}}_1)_{ii})^2 + \sum_{i=1}^{m}\sum_{j=i+1}^{m}\left(\left(\sigma_i(\tilde{\mathbf{Z}}_1)_{ij} - (\tilde{\mathbf{D}}_1)_{ij}\right)^2 + \left(\sigma_j(\mathbf{Z}_1)_{ij} - (\tilde{\mathbf{D}}_1)_{ji}\right)^2\right) \\ &+ \lambda\left(\sum_{i=1}^{m}\left((\tilde{\mathbf{Z}}_1)_{ii} - (\tilde{\mathbf{Z}_{\text{ref}}})_{ii}\right)^2 + \sum_{i=1}^{m}\sum_{j=i+1}^{m}\left(\left((\tilde{\mathbf{Z}}_1)_{ij} - (\tilde{\mathbf{Z}_{\text{ref}}})_{ij}\right)^2 + \left((\tilde{\mathbf{Z}}_1)_{ij} - (\tilde{\mathbf{Z}_{\text{ref}}})_{ji}\right)^2\right)\right).\end{aligned}$$

By setting the derivative w.r.t. z_{ij}, we obtain for $\lambda > 0$

$$(\tilde{\mathbf{Z}}_1)_{ij} = \frac{\sigma_i(\tilde{\mathbf{D}}_1)_{ij} + \sigma_j(\tilde{\mathbf{D}}_1)_{ji} + \lambda((\tilde{\mathbf{Z}_{\text{ref}}})_{ij} + (\tilde{\mathbf{Z}_{\text{ref}}})_{ji})}{\sigma_i^2 + \sigma_j^2 + 2\lambda},$$

Since $\mathbf{\Sigma}\tilde{\mathbf{D}}^T = \mathbf{\Sigma}\mathbf{U}^T\mathbf{D}^T\mathbf{V}_1^T = \mathbf{V}_1\mathbf{A}\mathbf{D}^T\mathbf{V}_1^T$, We can equivalently write

$$\tilde{\mathbf{Z}}_1 = \left(\frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2 + 2\lambda\mathbf{1}\mathbf{1}^T}\right) \odot \mathbf{V}_1^T\left(\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T + \lambda(\mathbf{Z}_{\text{ref}} + \mathbf{Z}_{\text{ref}}^T)\right)\mathbf{V}_1, \tag{C.6.7}$$

$$\tag{C.6.8}$$

where $\odot$ is the Hadamard product computing the product element-wise.

Summing the terms together. From equations (C.6.5), (C.6.6) and (C.6.7), the solution can be written as

$$\begin{aligned}\mathbf{Z}_\lambda &= \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}}_D \\ (\tilde{\mathbf{Z}}_D)^T & \tilde{\mathbf{Z}}_2\end{bmatrix}\begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}^T \\ &= \mathbf{V}_1\tilde{\mathbf{Z}}_1\mathbf{V}_1^T + \mathbf{V}_1\tilde{\mathbf{Z}}_D\mathbf{V}_2^T + \mathbf{V}_2\tilde{\mathbf{Z}}_D^T\mathbf{V}_1^T + \mathbf{V}_2\tilde{\mathbf{Z}}_2\mathbf{V}_2^T \\ &= \mathbf{Z}_1 + \mathbf{Z}_D + \mathbf{Z}_D^T + (\mathbf{I}-\mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I}-\mathbf{P}),\end{aligned} \tag{C.6.9}$$

where $\mathbf{P} = \mathbf{V}_1\mathbf{V}_1^T = \mathbf{I} - \mathbf{V}_2\mathbf{V}_2^T$ and $\mathbf{Z}_D = \mathbf{V}_1\Big(\mathbf{\Sigma}^T\mathbf{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\Big(\mathbf{A}\mathbf{D}^T + 2\lambda(\mathbf{Z}_0)_D\Big)(\mathbf{I}-\mathbf{P})$, and $\mathbf{Z}_1 = \mathbf{V}_1\tilde{\mathbf{Z}}_1\mathbf{V}_1^T$.

Below we compute the inverse of $\mathbf{Z}_*$. Since

$$\begin{aligned}\mathbf{Z}_* &= \mathbf{V}\begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}_D} \\ \tilde{\mathbf{Z}_D}^T & \tilde{\mathbf{Z}}_2\end{bmatrix}\mathbf{V}^T \\ &= \mathbf{V}\tilde{\mathbf{Z}}\mathbf{V}^T,\end{aligned} \tag{C.6.10}$$

we can write

$$\mathbf{Z}_*^{-1} = \mathbf{V}\tilde{\mathbf{Z}}^{-1}\mathbf{V}^T.$$

By the Woodbury matrix identity [81], we have

$$\tilde{\mathbf{Z}}^{-1} = \begin{bmatrix} \mathbf{M}_1 & -\mathbf{M}_1\tilde{\mathbf{Z}}_D\tilde{\mathbf{Z}}_2^{-1} \\ -\tilde{\mathbf{Z}}_2^{-1}\tilde{\mathbf{Z}}_D^T\mathbf{M}_1 & \tilde{\mathbf{Z}}_2^{-1} + \tilde{\mathbf{Z}}_2^{-1}\tilde{\mathbf{Z}}_D^T\mathbf{M}_1\tilde{\mathbf{Z}}_D\tilde{\mathbf{Z}}_2^{-1} \end{bmatrix}, \tag{C.6.11}$$

with $\mathbf{M}_1 = (\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{Z}_D}\tilde{\mathbf{Z}_2}^{-1}\tilde{\mathbf{Z}}_D^T)^{-1}$. Hence $\mathbf{Z}_*^{-1} = \mathbf{V}\tilde{\mathbf{Z}}^{-1}\mathbf{V}^T$ can be rewritten as

$$\begin{aligned} \mathbf{Z}_*^{-1} &= \mathbf{V}_1\mathbf{M}_1\mathbf{V}_1^T + \mathbf{V}_2\tilde{\mathbf{Z}_2}^{-1}\tilde{\mathbf{Z}}_D^T\mathbf{M}_1\tilde{\mathbf{Z}_D}\tilde{\mathbf{Z}_2}^{-1}\mathbf{V}_2^T \\ &\quad + \mathbf{V}_2\tilde{\mathbf{Z}_2}^{-1}\mathbf{V}_2^T - \mathbf{V}_1\mathbf{M}_1\tilde{\mathbf{Z}_D}\tilde{\mathbf{Z}_2}^{-1}\mathbf{V}_2^T - \mathbf{V}_2\tilde{\mathbf{Z}_2}^{-1}\tilde{\mathbf{Z}_D}^T\mathbf{M}_1\mathbf{V}_1^T \\ &= \mathbf{Q}\mathbf{M}\mathbf{Q}^T + (\mathbf{I}-\mathbf{P})\mathbf{Z}_0^{-1}(\mathbf{I}-\mathbf{P}), \end{aligned} \tag{C.6.12}$$

where $\mathbf{M} = \left(\mathbf{Z}_1 - \mathbf{Z}_D\mathbf{Z}_0^{-1}\mathbf{Z}_D^T\right)^{-1}$ and $\mathbf{Q} = \mathbf{V}_1 - (\mathbf{I}-\mathbf{P})\mathbf{Z}_0^{-1}\mathbf{Z}_D^T$. ■

C.7. Proof of Proposition 2.4.3

In this section, we divide the proof of Proposition 2.4.3 into Lemma C.7.1 and Lemma C.7.3, which correspond to the effect of nonzero λ for (2.4.3) and the perturbation of $\mathbf{A}$ and $\mathbf{D}$ for (2.4.4), respectively.

C.7.1. Effect of regularization

Lemma C.7.1. Let

$$\mathbf{Z}_* = \lim_{\lambda \to 0} \operatorname*{argmin}_{\mathbf{Z}=\mathbf{Z}^T \in \mathbb{R}^{d\times d}} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|_F^2 + \lambda \|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|_F^2 \tag{C.7.1}$$

be the solution to the procrustes problem with λ going to 0, and $\mathbf{Z}_\lambda$ be the solution to (RSP) given $\lambda > 0$. Then, it holds that

$$\|\mathbf{Z}_\lambda - \mathbf{Z}_\star\|_F \leq \frac{5\lambda \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F}{\sigma_{\min}^2(\mathbf{A}) + \lambda}. \tag{C.7.2}$$

Démonstration. We rewrite (C.6.9) for $\mathbf{Z}_\lambda$ and $\mathbf{Z}_*$ respectively,

$$\mathbf{Z}_\lambda = (\mathbf{Z}_\lambda)_1 + (\mathbf{Z}_\lambda)_D + (\mathbf{Z}_\lambda)_D^T + (\mathbf{I} - \mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I} - \mathbf{P}), \tag{C.7.3}$$

$$\mathbf{Z}_* = (\mathbf{Z}_*)_1 + (\mathbf{Z}_*)_D + (\mathbf{Z}_*)_D^T + (\mathbf{I} - \mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I} - \mathbf{P}). \tag{C.7.4}$$

With such notations, we have by triangle inequality,

$$\|\mathbf{Z}_\lambda - \mathbf{Z}_*\|_F \leq \underbrace{\|(\mathbf{Z}_\lambda)_1 - (\mathbf{Z}_*)_1\|_F}_{\text{(i)}} + 2\underbrace{\|(\mathbf{Z}_\lambda)_D - (\mathbf{Z}_*)_D\|_F}_{\text{(ii)}}. \tag{C.7.5}$$

To simplify notations, we define $\max|\mathbf{X}|$ and $\min|\mathbf{X}|$ as the maximum and minimum entry with the absolute value of matrix $\mathbf{X}$, respectively.

For term (i), by (C.6.7) and the symmetry of $\mathbf{Z}_{\text{ref}}$ we have

$$\begin{aligned}
\|(\mathbf{Z}_\lambda)_1 - (\mathbf{Z}_*)_1\|_F &= \left\| \left(\frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2 + 2\lambda\mathbf{1}\mathbf{1}^T} - \frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2} \right) \odot (\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T) \right. \\
&\quad \left. + \frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2 + 2\lambda\mathbf{1}\mathbf{1}^T} \odot 2\lambda\mathbf{Z}_{\text{ref}} \right\|_F \\
&= \left\| -2\lambda \cdot \left(\frac{1}{(\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T) \odot (\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2)} \right) \odot (\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T) \right. \\
&\quad \left. + \frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T} \odot 2\lambda\mathbf{Z}_{\text{ref}} \right\|_F \\
&= 2\lambda \left\| \left(\frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T} \right) \odot \left(\frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2} \odot (\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T) \right) \right. \\
&\quad \left. - \left(\frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T} \right) \odot \mathbf{Z}_{\text{ref}} \right\|_F \\
&= 2\lambda \left\| \left(\frac{1}{\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T} \right) \odot \Big((\mathbf{Z}_*)_1 - \mathbf{Z}_{\text{ref}} \Big) \right\|_F \\
&\le 2\lambda \cdot \frac{1}{\min |\mathbf{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\mathbf{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T|} \cdot \Big\| (\mathbf{Z}_*)_1 - \mathbf{Z}_{\text{ref}} \Big\|_F, \qquad \text{(C.7.6)}
\end{aligned}$$

where the computations of matrices are element-wise, and the first three equalities follows from the identity $\mathbf{A} \odot \mathbf{X} + \mathbf{B} \odot \mathbf{X} = (\mathbf{A} + \mathbf{B}) \odot \mathbf{X}$ of the Hadamard product for any matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}$ of the same dimensions. The fourth equality in (C.7.6) holds by the definition of $(\mathbf{Z}_*)_1$, and the last inequality is due to the fact that

$$\|\mathbf{A} \odot \mathbf{B}\|_F \le \max |\mathbf{A}| \cdot \|\mathbf{B}\|_F \qquad \text{(C.7.7)}$$

for any two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same dimensions.

For the term (ii), note that $(\mathbf{Z}_*)_D = \mathbf{V}_1\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{D}^T\mathbf{V}_2\mathbf{V}_2^T = \mathbf{V}_1\Big(\mathbf{\Sigma}^\top\mathbf{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2\mathbf{V}_2^T$. Since $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}$, we have

$$\mathbf{V}_1^T(\mathbf{Z}_*)_D\mathbf{V}_2 = \Big(\mathbf{\Sigma}^\top\mathbf{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2. \qquad \text{(C.7.8)}$$

Furthermore, by using the unitary invariance of the orthogonal matrix w.r.t. the Frobenius norm, we obtain

$$
\begin{aligned}
\|(\mathbf{Z}_\lambda)_D - (\mathbf{Z}_*)_D\|_F &= \Big\|\mathbf{V}_1\Big(\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\Big(\mathbf{A}\mathbf{D}^T + 2\lambda\mathbf{Z}_{\text{ref}}\Big) - \Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\Big)\mathbf{V}_2\mathbf{V}_2^T\Big\|_F \\
&\overset{\text{(a)}}{=} \Big\|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\Big(\mathbf{A}\mathbf{D}^T + 2\lambda\mathbf{Z}_{\text{ref}}\Big)\mathbf{V}_2 - \Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2\Big\|_F \\
&= \Big\|\Big(\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1} - \Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\Big)^{-1}\Big)\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2 + 2\lambda\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2\Big\|_F \\
&\overset{\text{(b)}}{=} \Big\| - 2\lambda\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2 + 2\lambda\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2\Big\|_F \\
&\overset{\text{(c)}}{=} 2\lambda\Big\|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\Big(\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\Big)^{-1}\mathbf{V}_1^T\mathbf{A}\mathbf{D}^T\mathbf{V}_2 - \mathbf{V}_1^T\mathbf{Z}_{\text{ref}}\mathbf{V}_2\Big)\Big\|_F \\
&\overset{\text{(d)}}{=} 2\lambda\Big\|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\mathbf{V}_1^T\Big((\mathbf{Z}_*)_D - \mathbf{Z}_{\text{ref}}\Big)\mathbf{V}_2\Big\|_F \\
&\leq 2\lambda\max\Big|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\Big| \cdot \Big\|\mathbf{V}_1^T\Big((\mathbf{Z}_*)_D - \mathbf{Z}_{\text{ref}}\Big)\mathbf{V}_2\Big\|_F \\
&\leq 2\lambda\max\Big|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\Big| \cdot \Big\|(\mathbf{Z}_*)_D - \mathbf{Z}_{\text{ref}}\Big\|_F, \qquad \text{(C.7.9)}
\end{aligned}
$$

where (a), (c) and the last equality hold by the unitary invariance of $\mathbf{V}_1$ and $\mathbf{V}_2$ w.r.t. the Frobenius norm, (b) holds since $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}$ and $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}$ are diagonal matrices, (d) follows from (C.7.8), and the first inequality holds since $(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I})^{-1}$ is a diagonal matrix. The last inequality holds since $\mathbf{V}_1\mathbf{V}_1^T$ and $\mathbf{V}_2\mathbf{V}_2^T$ are projections.

Therefore, by combining (C.7.5), (C.7.6) and (C.7.9) we have

$$
\begin{aligned}
\|\mathbf{Z}_\lambda - \mathbf{Z}_*\|_F &\leq \|(\mathbf{Z}_\lambda)_1 - (\mathbf{Z}_*)_1\|_F + 2\|(\mathbf{Z}_\lambda)_D - (\mathbf{Z}_*)_D\|_F \\
&\leq 2\lambda\frac{1}{\min|\boldsymbol{\Sigma}^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\boldsymbol{\Sigma}^2 + 2\lambda\mathbf{1}\mathbf{1}^T|} \cdot \Big\|(\mathbf{Z}_*)_1 - \mathbf{Z}_{\text{ref}}\Big\|_F \\
&\quad + 4\lambda\max\Big|\Big(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} + 2\lambda\mathbf{I}\Big)^{-1}\Big| \cdot \Big\|(\mathbf{Z}_*)_D - \mathbf{Z}_{\text{ref}}\Big\|_F \\
&\leq \frac{\lambda}{\sigma_{\min}^2(\mathbf{A}) + \lambda} \cdot \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F + \frac{4\lambda}{\sigma_{\min}^2(\mathbf{A}) + \lambda} \cdot \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F \\
&= \frac{5\lambda}{\sigma_{\min}^2(\mathbf{A}) + \lambda} \cdot \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F, \qquad \text{(C.7.10)}
\end{aligned}
$$

where the last inequality follows from the definition of the element-wise operator and the facts that $\|(\mathbf{Z}_*)_1 - \mathbf{Z}_{\text{ref}}\|_F \leq \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F$ and $\|(\mathbf{Z}_*)_D - \mathbf{Z}_{\text{ref}}\|_F \leq \|\mathbf{Z}_\star - \mathbf{Z}_{\text{ref}}\|_F$. Hence, we conclude the proof. ■

C.7.2. Perturbation of A and D

We first present a stability analysis result of the regularized least squares (RLS), which is used in the analysis for the perturbation of $\mathbf{A}$ and $\mathbf{D}$ in Lemma C.7.3.

Lemma C.7.2 (Stability analysis of regularized least squares)**.** Let $\mathbf{x}^*$ solve the problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \beta\|\mathbf{x} - \mathbf{x}_0\|_2^2, \tag{C.7.11}$$

where $\mathbf{x}$,$\mathbf{x}_0 \in \mathbb{R}^p$, $\mathbf{A} \in \mathbb{R}^{q\times p}$, $\mathbf{b} \in \mathbb{R}^q$ for some integer p,$q > 0$ and $\beta > 0$. Let $\hat{\mathbf{x}}$ solve

$$\min_{\mathbf{x}} \|(\mathbf{A} + \delta\mathbf{A})\mathbf{x} - (\mathbf{b} + \delta\mathbf{b})\|_2^2 + \beta\|\mathbf{x} - \mathbf{x}_0\|_2^2, \tag{C.7.12}$$

where $\delta\mathbf{A} \in \mathbb{R}^{q\times p}, \delta\mathbf{b} \in \mathbb{R}^q$, and $\|\delta\mathbf{A}\|_2 \ll \|\mathbf{A}\|_2$. Suppose that $\mathbf{rank}(\mathbf{A}) = \mathbf{rank}(\mathbf{A} + \delta\mathbf{A})$, we have

$$\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2 \leq \mathcal{O}\Big(\frac{\|\delta\mathbf{A}\|_2 + \|\delta\mathbf{b}\|_2}{\beta}\Big). \tag{C.7.13}$$

Démonstration. By definition, we have explicitly that

$$\mathbf{x}^* = (\mathbf{A}^T\mathbf{A} + \beta\mathbf{I})^{-1}(\mathbf{A}^T\mathbf{b} + \beta\mathbf{x}_0). \tag{C.7.14}$$

Let $\tilde{\mathbf{A}} = \mathbf{A} + \delta\mathbf{A}$ and $\mathbf{P} = -\mathbf{A}^T\mathbf{A} + \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$, we can write

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}((\mathbf{A} + \delta\mathbf{A})^T(\mathbf{b} + \delta\mathbf{b}) + \beta\mathbf{x}_0). \tag{C.7.15}$$

Hence, we obtain

$$\begin{aligned}
\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2 &\leq \Big\|\Big((\mathbf{A}^T\mathbf{A} + \beta\mathbf{I})^{-1} - (\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\Big)(\mathbf{A}^T\mathbf{b} + \beta\mathbf{x}_0)\Big\|_2 \\
&\quad + \Big\|(\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\Big\|_2\|\delta\mathbf{A}\|_2\|\mathbf{b} + \delta\mathbf{b}\|_2 + \Big\|(\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\Big\|_2\|\mathbf{A}\|_2\|\delta\mathbf{b}\|_2 \\
&= \Big\|(\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\mathbf{P}(\mathbf{A}^T\mathbf{A} + \beta\mathbf{I})^{-1}(\mathbf{A}^T\mathbf{b} + \beta\mathbf{x}_0)\Big\|_2 \\
&\quad + \Big\|(\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\Big\|_2\|\delta\mathbf{A}\|_2\|\mathbf{b} + \delta\mathbf{b}\|_2 + \Big\|(\mathbf{A}^T\mathbf{A} + \mathbf{P} + \beta\mathbf{I})^{-1}\Big\|_2\|\mathbf{A}\|_2\|\delta\mathbf{b}\|_2 \\
&\leq \frac{1}{\beta}\cdot\Big(\|\mathbf{P}\|_2\|\mathbf{x}^*\|_2 + \|\delta\mathbf{A}\|_2\|\mathbf{b}\|_2 + \|\delta\mathbf{b}\|_2\|\mathbf{A}\|_2 + \|\delta\mathbf{b}\|_2\|\delta\mathbf{A}\|_2\Big).
\end{aligned} \tag{C.7.16}$$

Since $\|\delta\mathbf{A}\|_2 \ll \|\mathbf{A}\|_2$, we obtain

$$\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \mathcal{O}\Big(\frac{\|\delta\mathbf{A}\|_2 + \|\delta\mathbf{b}\|_2}{\beta}\Big), \tag{C.7.17}$$

which concludes the proof of the lemma. ■

Now we show the stability analysis with respect to the perturbation of $\mathbf{A}$ and $\mathbf{D}$ below.

Lemma C.7.3. Let $\hat{\mathbf{Z}}$ solve

$$\min_{\mathbf{Z}=\mathbf{Z}^T\in\mathbb{R}^{d\times d}} \|\mathbf{Z}(\mathbf{A} + \delta\mathbf{A}) - (\mathbf{D} + \delta\mathbf{D})\|_F^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|_F^2, \tag{C.7.18}$$

where $\delta\mathbf{A}, \delta\mathbf{D} \in \mathbb{R}^{d\times m}$, $\|\delta\mathbf{A}\|_2 \ll \|\mathbf{A}\|_2$, and $\|\delta\mathbf{D}\|_2 \ll \|\mathbf{D}\|_2$. Also, suppose $\mathbf{Z}_\lambda$ to be the solution to (RSP) given $\lambda > 0$. Then, it holds that

$$\|\hat{\mathbf{Z}} - \mathbf{Z}_\lambda\|_F \leq \mathcal{O}\Big(\frac{\|\delta\mathbf{A}\|_2 + \|\delta\mathbf{d}\|_2}{\lambda}\Big). \tag{C.7.19}$$

Démonstration. We first reduce (RSP) to an unconstrained regularized least squares (RLS) problem as follows. Let $r = md$ and $s = d^2$. We denote by **vec** the operator that stacks the columns of a matrix into a long vector. Then, for any $\mathbf{Z} = \mathbf{Z}^T \in \mathbb{R}^{d\times d}$ it follows that

$$\begin{aligned}\|\mathbf{ZA} - \mathbf{D}\|_F + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{Z}_{\text{ref}}\|_F &= \|\mathbf{vec}(\mathbf{ZA} - \mathbf{D})\|_2 + \frac{\lambda}{2}\|\mathbf{vec}(\mathbf{Z} - \mathbf{Z}_{\text{ref}})\|_2 \\ &= \|(\mathbf{I}_d \otimes \mathbf{A})\mathbf{z} - \mathbf{d}\|_2 + \frac{\lambda}{2}\|\mathbf{vec}(\mathbf{z} - \mathbf{z}_{\text{ref}})\|_2 \end{aligned} \tag{C.7.20}$$

Here $(\mathbf{I}_d \otimes \mathbf{A})_{ij} = \delta_{ij}\mathbf{A} \in \mathbb{R}^{r\times s}$, $\mathbf{z} = \mathbf{vec}(\mathbf{Z}) \in \mathbb{R}^s, \mathbf{z}_{\text{ref}} = \mathbf{vec}(\mathbf{Z}_{\text{ref}}) \in \mathbb{R}^s$, and $\mathbf{d} = \mathbf{vec}(\mathbf{D}) \in \mathbb{R}^r$. We define $\mathbf{S}_d$ as the matrix where the columns form an orthonormal basis for a $\bar{d}$-dimensional subspace of $\mathbb{R}^s$, where $\bar{d} = \dfrac{d(d+1)}{2}$. By using the symmetry of $\mathbf{X}$, letting $\mathbf{z} = \mathbf{S}_d\mathbf{y}$, $\mathbf{z}_{\text{ref}} = \mathbf{S}_d\mathbf{y}_{\text{ref}}$ and $\mathbf{H} = (\mathbf{I}_d \otimes \mathbf{A})\mathbf{S}_d$, we are able to obtain an regularized LS problem equivalent to (RSP) as follows,

$$\min_{\mathbf{y}\in\mathbb{R}^{\bar{d}}} \|\mathbf{Hy} - \mathbf{d}\|_2 + \frac{\lambda}{2}\|\mathbf{y} - \mathbf{y}_{\text{ref}}\|_2. \tag{C.7.21}$$

Here we have used the fact that for an orthonormal matrix $\mathbf{S}_d$ we have $\|\mathbf{S}_d(\mathbf{y} - \mathbf{y}_{\text{ref}})\|_2 = \|\mathbf{y} - \mathbf{y}_{\text{ref}}\|_2$. Likewise, we can identify the perturbed problem (C.7.18) with perturbations

$$\mathbf{H} \to \mathbf{H} + \delta\mathbf{H}, \quad \mathbf{d} \to \mathbf{d} + \delta\mathbf{d}, \quad \mathbf{y} \to \tilde{\mathbf{y}} \tag{C.7.22}$$

in (C.7.21), where

$$\delta\mathbf{H} = (\mathbf{I}_d \otimes \delta\mathbf{A})\mathbf{S}_d, \quad \delta\mathbf{d} = \mathbf{vec}(\delta\mathbf{D}), \quad \mathbf{S}_d\hat{\mathbf{y}} = \mathbf{vec}(\tilde{\mathbf{Z}}). \tag{C.7.23}$$

Furthermore, the solution to (C.7.21) can be written as

$$\mathbf{y}^* = \Big(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}/2\Big)^{-1}\mathbf{H}^T\mathbf{d}. \tag{C.7.24}$$

Then, the solution is perturbed to

$$\hat{\mathbf{y}} = \mathbf{y} + \delta\mathbf{y} = \Big((\mathbf{H} + \delta\mathbf{H})^T(\mathbf{H} + \delta\mathbf{H}) + \lambda\mathbf{I}/2\Big)^{-1}(\mathbf{H} + \delta\mathbf{H})^T(\mathbf{d} + \delta\mathbf{d}).$$

After the reduction of (RSP) to (C.7.21), we apply Lemma C.7.2 to (C.7.21), which yields

$$\|\hat{\mathbf{y}} - \mathbf{y}^*\|_2 \leq \mathcal{O}\Big(\frac{\|\delta\mathbf{H}\|_2 + \|\delta\mathbf{d}\|_2}{\lambda}\Big). \tag{C.7.25}$$

Also, by the definition of $\mathbf{H}$ we have $\|\delta\mathbf{H}\|_2 = \|\delta\mathbf{A}\|_2$ where $\mathbf{A}$ is defined in the original problem (RSP). Hence, (C.7.25) reads

$$\|\hat{\mathbf{y}} - \mathbf{y}^*\|_2 \leq \mathcal{O}\Big(\frac{\|\delta\mathbf{A}\|_2 + \|\delta\mathbf{d}\|_2}{\lambda}\Big). \tag{C.7.26}$$

Further, we can write

$$\begin{aligned}
\Big\|\hat{\mathbf{Z}} - \mathbf{Z}_\lambda\Big\|_F &= \|\mathbf{vec}\Big(\hat{\mathbf{Z}}\Big) - \mathbf{vec}\Big(\mathbf{Z}_\lambda\Big)\|_2 \\
&= \|\mathbf{S}_d\tilde{\mathbf{y}} - \mathbf{S}_d\mathbf{y}^*\|_2 \\
&= \|\hat{\mathbf{y}} - \mathbf{y}^*\|_2 \\
&\leq \mathcal{O}\Big(\frac{\|\delta\mathbf{A}\|_2 + \|\delta\mathbf{d}\|_2}{\lambda}\Big),
\end{aligned} \tag{C.7.27}$$

where the third equality holds since the columns of $\mathbf{S}_d$ form an orthonormal basis. This concludes the proof. ■

Furthermore, we remark that Lemma C.7.1 and Lemma C.7.3 together concludes the proof of Proposition 2.4.3.

C.8. Numerical Experiments

C.8.1. Datasets

We used several UCI datasets, whose main characteristics are summarized in Table C.1. In the case of the *P53 mutant* dataset, we reduce its size to avoid memory problems. We kept all labels where $y = 1$ (153 instances), and merge them with the 5000 first data points.

Dataset name	Tag	# features	# data points	Section
Madelon [38]	Madelon	500	4400	C.8.7
Internet Advertisements [47]	Ad	1558	3279	C.8.8
QSAR oral toxicity [4]	Qsar	1024	8992	C.8.9
p53 Mutants Data Set [15]	P53 mutant	5406	5000	C.8.10

Tableau C.1. Summary of the datasets used in the numerical experiments.

C.8.2. Setting

We consider the regression problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=0}^{N} \ell(\mathbf{a}_i^T\mathbf{x},\mathbf{b}_i) + \frac{\tau}{2}\|x\|_2^2, \tag{C.8.1}$$

where $\ell(\cdot,\cdot)$ is either a quadratic or a logistic loss. The pair $(\mathbf{A},\mathbf{b})$ is a dataset, where $a_i \in \mathbb{R}^d$ is a data point composed by d features, and b_i is the label of the i^{th} data point. We solve the problem using deterministic and stochastic gradient, whose parameters are described in Table C.2. The optimal value of (C.8.1) are obtained using the MATLAB package `minfunc` from [70].

Parameter	Deterministic setting	Stochastic setting
τ	`1e-9` (ill-conditioned problem)	`1e-2`
Descent direction	Full gradient	SAGA (see [19])
Batch size	Full batch	64
Limited memory m	10 and ∞	25
Line-search	None or approximate dichotomy	None
$\mathbf{B}_{\text{ref}}^{-1}$ and $\mathbf{H}_{\text{ref}}$ (No LS)	$\frac{1}{\|A\|_2^2}$ (quad.), $\frac{1}{4\|A\|_2^2}$ (logistic)	$\frac{1}{3\max_i L_i}$ [19]
$\mathbf{B}_{\text{ref}}^{-1}$ and $\mathbf{H}_{\text{ref}}$ (with LS)	1	N/A.
Rel. reg. $\bar{\lambda}$ (if applicable)	`1e-20` (quad), `1e-10` (logistic)	`1e-2`
Max. iteration	`250` (full batch)	`1e4` (mini-batches)

Tableau C.2. Parameters used to optimize (C.8.1)

C.8.3. Observation

Unitary step VS line search. Most of the presented method present a divergent behavior when we do not apply line search. However, it seems that the Multisecant Type-I method is the most robust one, converging for almost all instances. In fact, it seems that adding a line-search to method slow it down - probably because the optimal stepsize is close to one, but it takes time to have the guarantee. When it comes to line-search methods, there is no clear method whose speed is superior. Surprisingly, in both cases, the Type-II symmetric multisecant method seems to be the worst one (after gradient descent).
Stochastic optimization. As it may be expected, the symmetric multisecant type-I is the fastest method. Indeed, our updates have provably better robustness, and the type-I symmetric multisecant update is the best one amongst all method with unitary step-size. However, its performance are not much different than gradient descent. Moreover, the author indicate that the mini-batch size plays an important role in the convergence of the method, as smaller batches have too much variance. We suspect there is a trade-off to improve the speed of the method, where we should balance the size of the batch and the number of secant equations.

C.8.4. Spectrum Recovery on Madelon (Quadratic Loss)

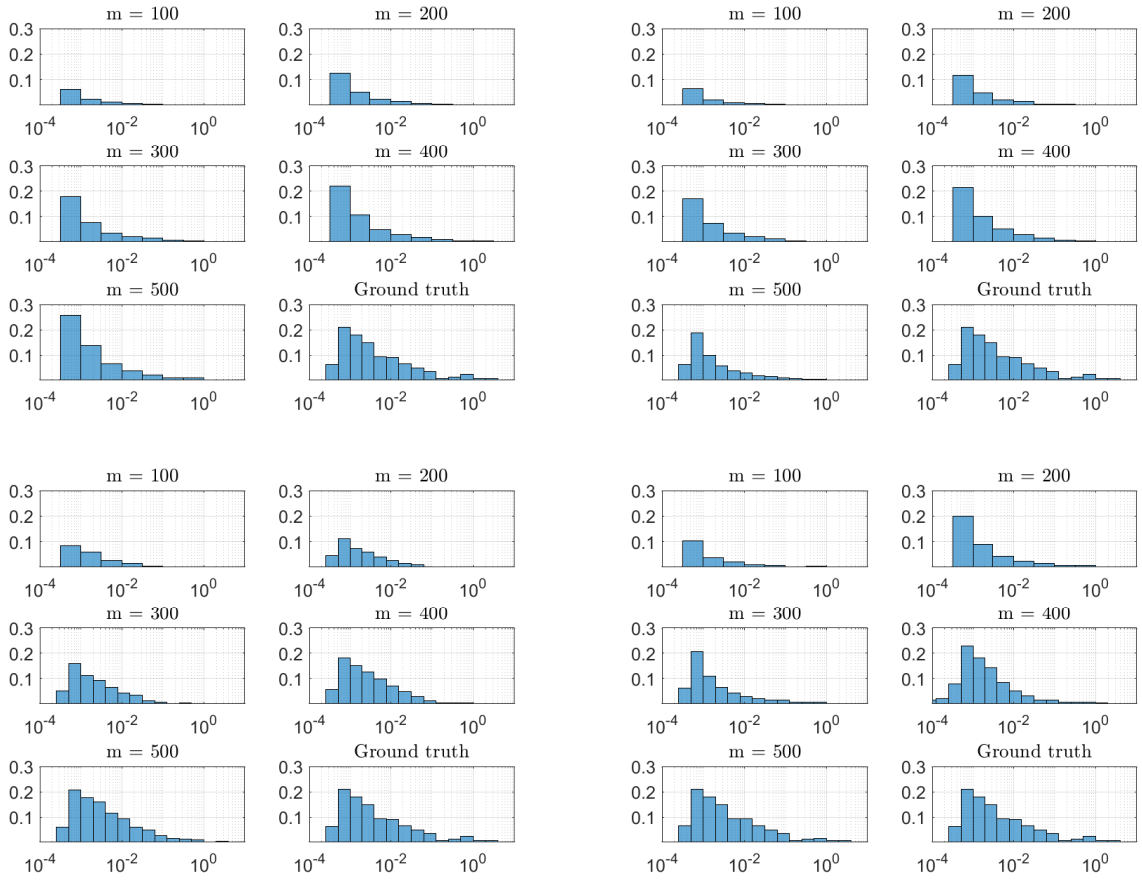


Fig. C.1. Histogram of the eigenvalues of the estimate $\mathbf{H}_k$ or $\mathbf{B}_k^{-1}$ in the function on the iteration counter (i.e., the number of secant equations), when optimizing the square loss on the Madelon dataset without regularization. **Top left:** Multisecant Broyden Type-I, **Top right:** Multisecant Broyden Type-II, **Bottom left:** Type-I symmetric multisecant, **Bottom right:** Type-II symmetric multisecant. For the non-symmetric updates, we took the real part of the eigenvalues. We removed from the histogram the spike of eigenvalues associated to $\mathbf{H}_{\text{ref}}$ or $\mathbf{B}_{\text{ref}}^{-1}$ (initialized at $1/L$, the smoothness constant of the function). It seems that the spectrum converges faster to the ground truth when we use symmetric updates. We did not report BFGS as the method is non-convergent with unitary stepsize.

C.8.5. Organization of figures

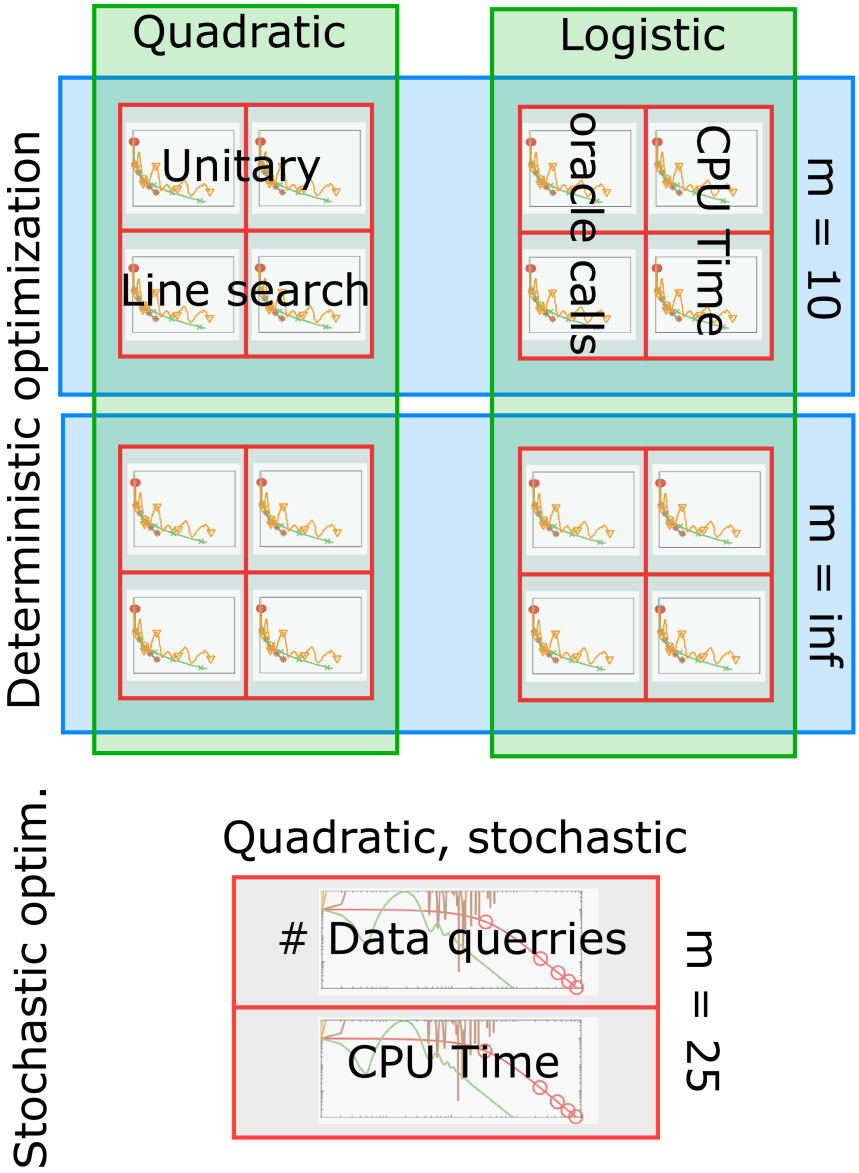
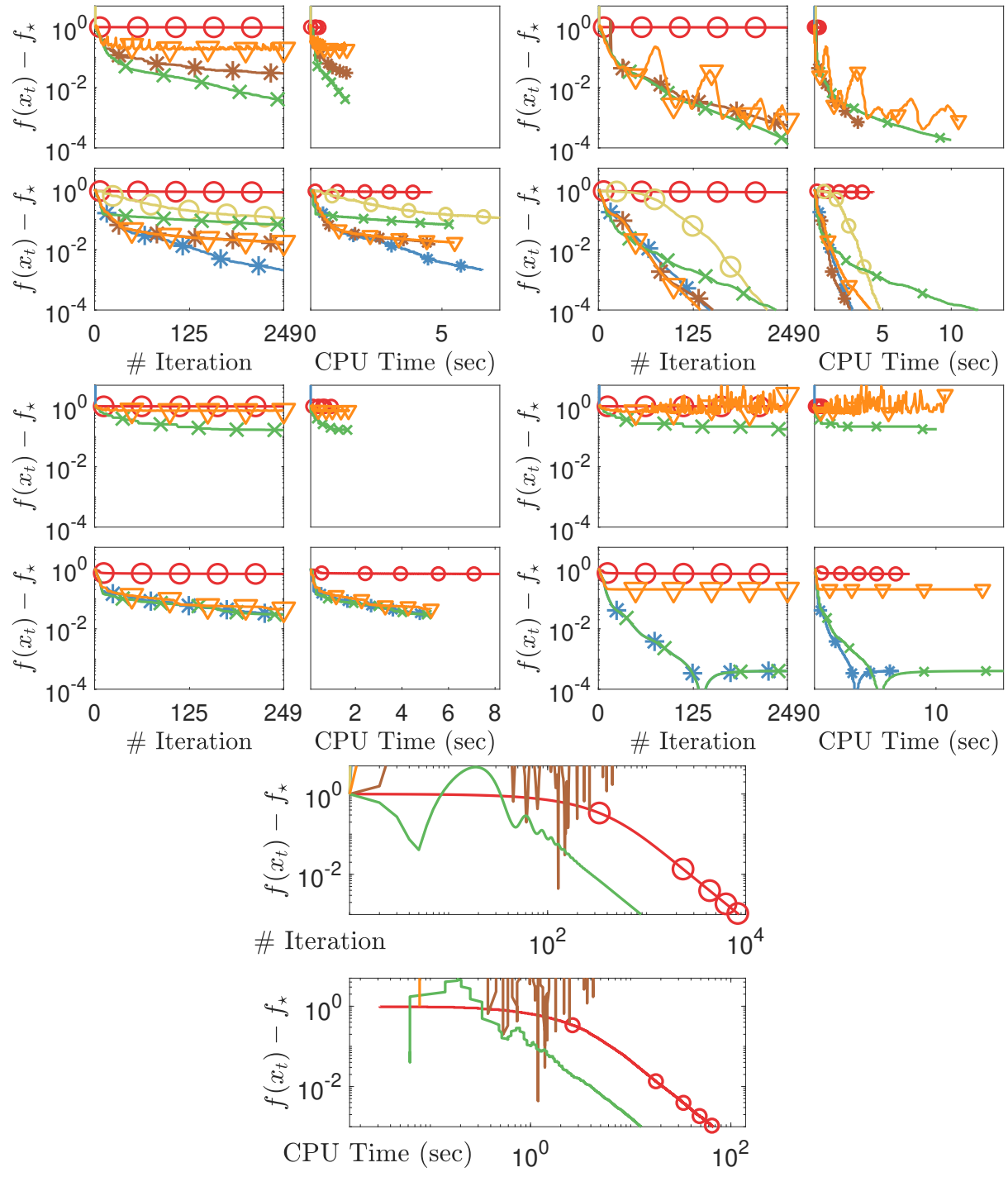


Fig. C.2. Organization of figures for the numerical experiments.

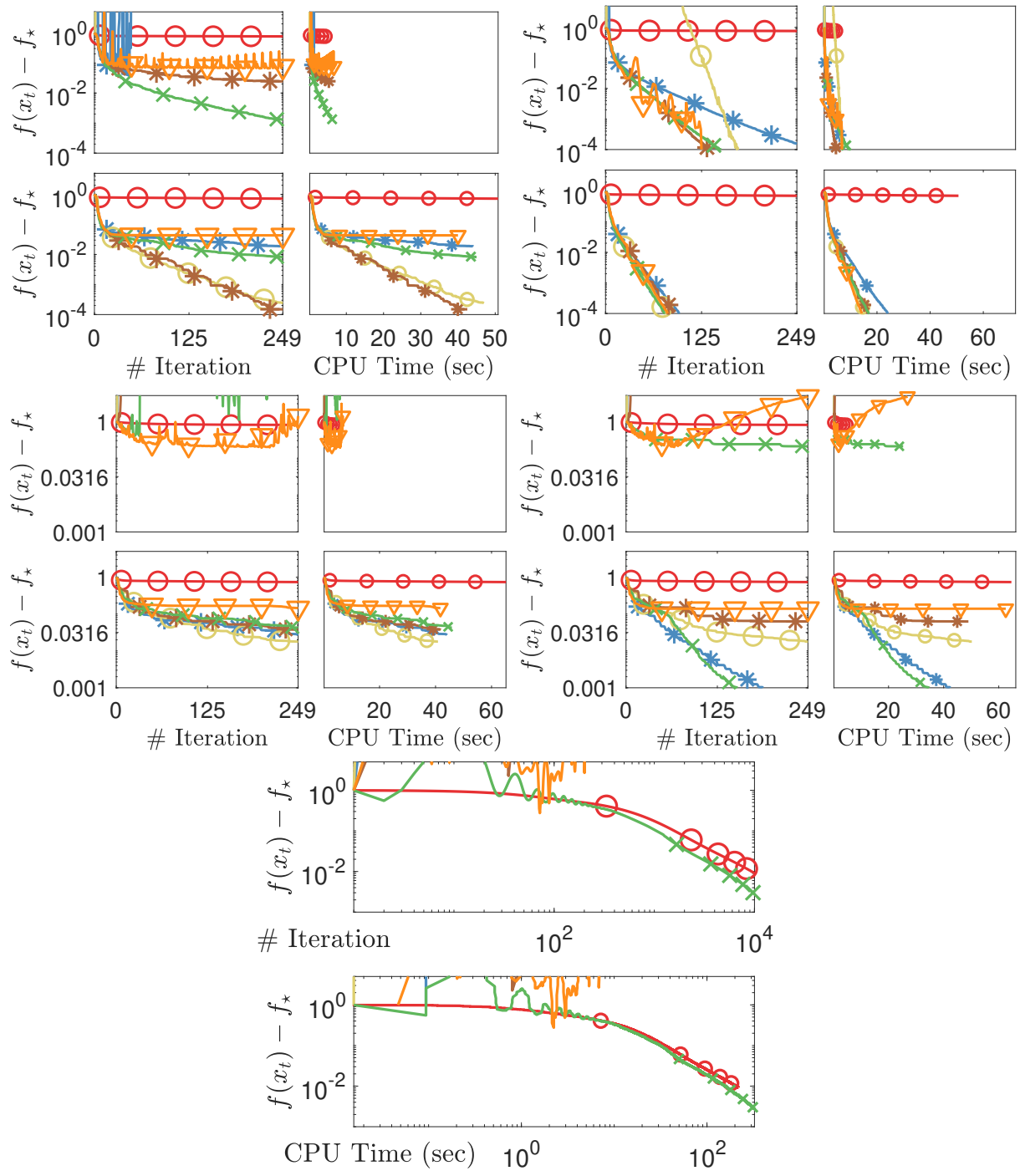
C.8.6. Legend

Fig. C.3. Legend for all subsequent figures

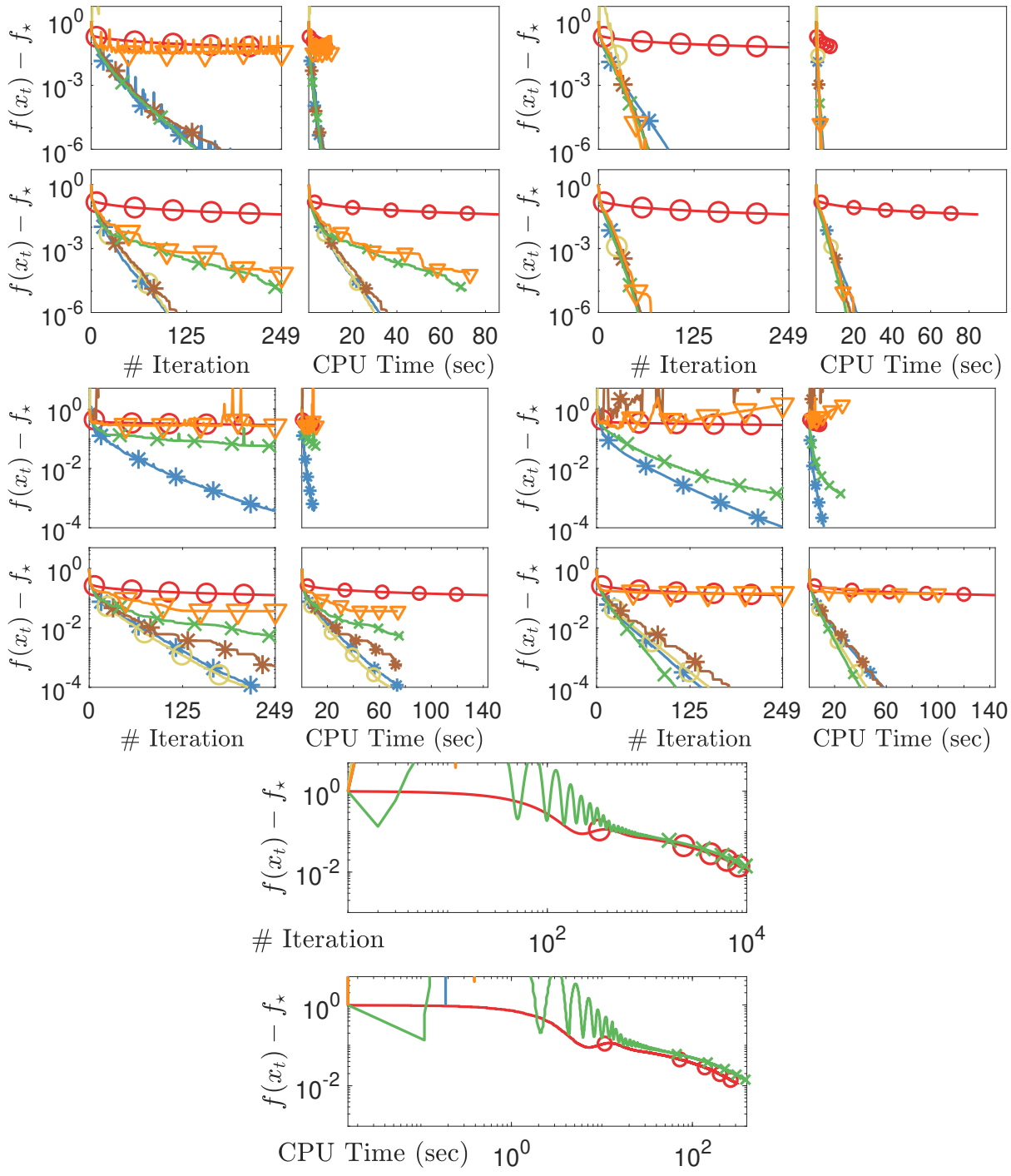
C.8.7. Madelon


$f(x_t) - f_\star$
10^0
10^{-2}
10^{-4}
0
125
2490
249
5
10
2
4
6
8
Iteration
CPU Time (sec)
10^2
10^4

C.8.8. Ad


$f(x_t) - f_\star$
10^0
10^{-2}
10^{-4}
1
0.0316
0.001
0
125
249
10 20 30 40 50
20 40 60
Iteration
CPU Time (sec)
10^2
10^4

C.8.9. Qsar


$f(x_t) - f_\star$
10^0
10^{-3}
10^{-6}
0
125
249
Iteration
20 40 60 80
CPU Time (sec)
10^{-2}
10^{-4}
20 60 100 140
10^2
10^4

C.8.10. P53 Mutant