Mathematics and Mathematical Statistics, 2026

From Accelerated First-Order Methods to Structured Nonconvex Optimization

Analysis and Perspectives

Hoomaan Maskan

UMEÅ UNIVERSITY

Department of Mathematics and Mathematical Statistics
Umeå University
SE-901 87 Umeå, Sweden

This work was supported by the Wallenberg Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundations.

This work is protected by the Swedish Copyright Act (Act 1960:729)
The author retains the copyright of the material in this thesis. Licensing information for published and redistributed papers is provided on the first page of each appended paper.
ISBN: 978-91-8070-910-1 (print)
ISBN: 978-91-8070-911-8 (pdf)
ISSN: 1653-0829
Research report in mathematical statistics no. 80/26
Cover design: The right illustration shows the behavior of the Frank–Wolfe method for a convex optimization problem, while the left illustration depicts a nonconvex function expressed as the difference of two convex functions. Both illustrations are standard in the optimization literature and were created by the author.
Electronic version available at: http://umu.diva-portal.org/
Printed by: Scandinavian Print Group, Skarpnäck, 2026

To my family,

who turned separation into direction.

Abstract

Modern technologies like machine learning and data-driven decision systems, depend on solving large and often highly complex optimization problems. These problems rarely come with simple shapes or smooth surfaces; instead, they can twist and bend in ways that make finding good solutions surprisingly difficult. This thesis explores two ideas that help us navigate such complexity more effectively.

The first idea focuses on acceleration and investigates how optimization algorithms can reach good solutions faster while using only basic information such as function values and gradients. By studying these algorithms through a continuous-time analysis, we show that many of the fastest methods behave like carefully designed dynamical systems. This perspective not only clarifies why acceleration happens, but also allows us to design fast algorithms and understand how they behave in the presence of noisy information.

The second idea focuses on a mathematical structural assumption called difference-of-convex (DC) programming, which captures a remarkably wide range of nonconvex problems. By leveraging this structure, we develop practical algorithms that avoid costly operations like projections or high dimensional gradient evaluations, making them efficient for large-scale applications. Through the connection between DC programming and the classical Expectation–Maximization (EM) algorithm, we develop more scalable EM variants and provide the first general performance guarantees for them.

Keywords: accelerated methods, convex optimization, nonconvex optimization, DC programming, EM algorithm, randomized DC algorithm

Sammanfattning

Moderna teknologier, såsom maskininlärning och datadrivna beslutsstödssystem, är beroende av att lösa stora och ofta mycket komplexa optimeringsproblem. Dessa problem har sällan enkla strukturer eller släta ytor; i stället kan de vara vridna och oregelbundna på sätt som gör det förvånansvärt svårt att hitta bra lösningar. Denna avhandling undersöker två idéer som gör det möjligt att hantera sådan komplexitet på ett mer effektivt sätt.

Den första idén fokuserar på acceleration och studerar hur optimeringsalgoritmer kan nå bra lösningar snabbare genom att endast använda grundläggande information, såsom funktionsvärden och gradienter. Genom att analysera dessa algoritmer ur ett kontinuerligt tidsperspektiv visar vi att många av de snabbaste metoderna beter sig som noggrant konstruerade dynamiska system. Detta perspektiv klargör inte bara varför acceleration uppstår, utan möjliggör även konstruktionen av snabba algoritmer och en förståelse för hur de påverkas av brusig information.

Den andra idén fokuserar på ett matematiskt strukturellt antagande kallat skillnad-av-konvexa (DC) programmering, vilket fångar en anmärkningsvärt bred klass av icke-konvexa problem. Genom att utnyttja denna struktur utvecklar vi praktiska algoritmer som undviker kostsamma operationer såsom projektioner eller gradientberäkningar i hög dimension, vilket gör dem väl lämpade för storskaliga tillämpningar. Genom kopplingen mellan DC-programmering och den klassiska Expectation–Maximization-algoritmen (EM) utvecklar vi mer skalbara varianter av EM och tillhandahåller de första generella prestandagarantierna för dessa metoder.

Nyckelord: accelererade metoder, konvex optimering, icke-konvex optimering, DC programmering, EM algoritmen, randomiserade DC algoritmer

Popular Science Summary

Many modern technologies rely on computers making good decisions in situations that involve large amounts of data. Machine learning systems adjust millions of parameters to improve predictions, statistical models attempt to infer hidden patterns from incomplete observations, and automated decision systems seek to allocate limited resources efficiently. At the heart of all these tasks lies a mathematical problem known as *optimization*: finding the best possible solution among many alternatives.

Although the goal of optimization is conceptually simple, real-world problems are often extremely difficult to solve. Instead of smooth and well-behaved mathematical surfaces that guide algorithms directly to an optimal solution, practical problems tend to be high dimensional, irregular, and filled with competing solutions. Algorithms navigating such landscapes can slow down, become unstable, or get trapped far from the best solution. Understanding how to design methods that are both fast and reliable in these settings is therefore a central challenge in modern optimization.

This thesis investigates how such challenges can be addressed by studying both the behavior of optimization algorithms and the structure of the problems they are applied to. The guiding idea is that difficult problems often become manageable once their underlying mechanisms are properly understood.

One striking observation in optimization is that some algorithms reach good solutions much faster than others, even though they rely on the same basic information. Many methods use gradients, which indicate the direction in which a function increases most rapidly and serve as a local guide for improvement. Yet algorithms that use gradients in slightly different ways can behave very differently in practice. This phenomenon, commonly referred to as *acceleration*, has played a major role in the success of modern optimization methods, but for a long time it lacked a clear conceptual explanation.

To shed light on this behavior, the first part of the thesis adopts a continuous time perspective. Instead of viewing optimization as a sequence of discrete updates, algorithms are interpreted as approximations of smooth dynamical systems, similar to mathematical models used to describe physical motion. From this point of view, an optimization algorithm behaves like a moving object influenced by forces such as inertia and damping.

This interpretation reveals that accelerated algorithms use momentum to carry useful

information forward, allowing them to move quickly across the optimization landscape without becoming unstable. Acceleration is no longer seen as a mysterious trick, but as the result of carefully balancing motion and dissipation. This perspective also makes it possible to compare different accelerated methods within a unified framework and to understand how they behave when the information they rely on is noisy, which is often the case in data-driven applications.

While speed is important, many optimization problems remain difficult even for fast algorithms because of their mathematical structure. In particular, many real world problems are *nonconvex*. Unlike convex problems, which have a single global solution and no misleading local optima, nonconvex problems can contain many competing solutions and flat regions where progress becomes difficult. This makes it challenging to determine whether an algorithm has found a genuinely good solution.

The second part of the thesis addresses this issue through a modeling principle known as *difference of convex programming*. The key idea is that many complex nonconvex objectives can be expressed as the difference of two convex functions. Although the original problem may appear highly irregular, this decomposition exposes structure that can be exploited algorithmically, known as the difference of convex algorithm. By repeatedly solving simpler convex subproblems, it becomes possible to make steady progress even in challenging nonconvex settings.

Building on this idea, the thesis develops optimization algorithms that avoid computationally expensive operations common in high dimensional problems. These methods are particularly well suited for large-scale applications, where memory and computational costs are critical concerns. Importantly, the proposed algorithms come with rigorous guarantees that describe how and when they converge.

This perspective also leads to a deeper understanding of the expectation–maximization algorithm, a classical method used in statistics and machine learning to learn from incomplete data. By viewing expectation-maximization as a special case of difference of convex algorithm, the thesis provides a unified explanation of its behavior and develops new variants that scale more effectively. This connection also enables the establishment of general convergence guarantees, which have historically been difficult to obtain.

To further improve efficiency in very large problems, the thesis explores randomized and block-coordinate update strategies. Instead of updating all variables at once, these methods modify only a small portion of the solution at each step. This significantly reduces computational effort while still ensuring meaningful progress toward a solution.

Overall, this thesis shows that understanding algorithms' behaviour through difficult landscapes and exploiting hidden mathematical structures, make it possible to design methods that are both efficient and reliable. These insights contribute to a deeper understanding of how modern computational systems learn from data and make decisions in complex environments.

Publications

This thesis is based on the following papers:

Paper I. **Maskan, H.**, Zygalakis, K., & Yurtsever, A. (2023). A variational perspective on high-resolution ODEs. Advances in Neural Information Processing Systems, 36, 1589-1600.

Paper II. **Maskan, H.**, Zygalakis, K. C., Eftekhari, A., & Yurtsever, A. (2025). A Unified Model for High-Resolution ODEs: New Insights on Accelerated Methods. arXiv preprint arXiv:2503.15136, (Under Review).

Paper III. **Maskan, H.**, Hou, Y., Sra, S., & Yurtsever, A. (2025). Revisiting Frank-Wolfe for structured nonconvex optimization. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.

Paper IV. **Maskan, H.**, Halvachi, P., Sra, S., & Yurtsever, A. (2025). Randomized block coordinate DC algorithm. EURO Journal on Computational Optimization, 100123.

Author Contributions

This section summarizes the contributions of Hoomaan Maskan (H.M.) to the papers included in this thesis.

Papers I & II. H.M. contributed majorly to the conception of the study, development of the methodology, theoretical analysis, numerical experiments, and the writing of the manuscript.

Paper III. H.M. contributed partly to the conception of the study and development of the methodology. H.M. contributed majorly to theoretical analysis, numerical experiments, and the writing of the manuscript.

Paper IV. H.M. contributed partly to the conception of the study. H.M. contributed majorly to the development of the methodology, theoretical analysis, numerical experiments, and the writing of the manuscript.

This thesis was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

Acknowledgements

With this thesis, I conclude my PhD studies after five and a half years. Throughout this journey, I was blessed with the support, kindness, and generosity of many beautiful souls.

From the very beginning, I felt welcomed at the Department of Mathematics and Mathematical Statistics at Umeå University. I still vividly remember my first day, when my kind co-advisor Jun Yu welcomed me at the Umeå city airport—a gesture that set the tone for the years to come. Over time, I was fortunate to be surrounded by wonderful colleagues and friends. I sincerely thank Konrad Abramowicz, Lars Mattsson, Per Åhag, Per Arnqvist, Anna Ivarsson, Mehdi Moradi, Maryam Sharifzade, Natalia Pya Arnqvist, Jun Yu, Michelle Di Sabato, and Yuwei Jiang for their support, warmth, and friendship.

During the first two and a half years of my PhD, I had the pleasure of working under the supervision of Armin Eftekhari. Although our time together was relatively short, I learned a great deal from him, and I am grateful for his guidance and support. I wish him peace and happiness in all his future endeavors.

When Armin left Umeå, Alp Yurtsever took over my supervision. Alp supported me patiently and wholeheartedly, offering guidance not only in scientific thinking and mathematical analysis, but also in writing, ethics, and navigating both academic and industrial worlds. His vision of scientific life has profoundly shaped my own. Dear Alp, your support means more to me than words can express—thank you.

As a member of the *Wallenberg AI, Autonomous Systems and Software Program (WASP)* community, I had the opportunity to collaborate and exchange ideas with many inspiring researchers. In particular, I would like to thank Saeed Razavi and Amandine Caut, among others, for these valuable interactions.

I am also deeply grateful to my collaborators Yikun Hou and Paniz Halvachi for their dedication and hard work. During my PhD, I had the privilege of visiting Konstantinos Zygalakis at the University of Edinburgh and Suvrit Sra at the Technical University of Munich. On my second visit, I also had the pleasure of meeting Pouria Fatemi, with whom I enjoyed many insightful discussions that led to further collaborations. I sincerely thank them all for their generosity and guidance. Without their support, this journey would not have been the same.

I would also like to thank my master's advisor, Mohammad Hossein Kahaei, who played

a key role in motivating me and introducing me to the world of research.

Throughout my PhD, I formed strong friendships that carried me through difficult moments and sustained me during challenging times. I would like to thank Rouien Zarin, Farah Ranjbarian, Klara Hofer Mattson, Anders Hofer, Azadeh Shirzadi, Mojgan Padyab, Aila Sarfaraz, and Saber Anooshe for bringing light into my life, especially during the long and dark winters.

Finally, I owe my deepest gratitude to my family, and especially to my brother Hooman, who never gave up on me and stood by my side unconditionally, even at my lowest points. Thank you for your endless love, patience, and support.

Hoomaan Maskan, February 2026

Contents

Chapter 1

Introduction

Modern applications in machine learning, statistics, and signal processing often give rise to large-scale optimization problems whose tractability depends critically on the structural properties of the objective function or the feasible set [Candès and Wakin, 2008, Beck and Teboulle, 2009, Bottou et al., 2018]. In many cases, identifying and exploiting such structure is essential for the design of efficient algorithms. For example, blind deconvolution problems in signal processing lead to nonconvex bilinear optimization formulations [Kundur and Hatzinakos, 2002]. Under additional modeling assumptions, these problems can be relaxed into semidefinite programs [Hezave et al., 2020]. Also, sparse MRI recovery naturally admits a convex optimization formulation that exploits sparsity and smoothness [Lustig et al., 2007], while decision-aware resource allocation problems give rise to intrinsically nonconvex optimization tasks arising from predictive–decision interactions [Elmachtoub and Grigas, 2022].

Despite their differences, these examples share a common theme: algorithmic efficiency hinges on understanding and leveraging the underlying structure of the optimization problem. This observation motivates a systematic study of optimization methods that exploit such structural properties. More formally, consider the generic optimization model:

$$\min_{\boldsymbol{x}\in\mathcal{D}} \ \phi(\boldsymbol{x}), \tag{1.1}$$

where $\phi(\boldsymbol{x}) : \mathcal{D} \to \mathbb{R}$ is the objective function and $\mathcal{D} \subseteq \mathbb{R}^n$ is a constraint set. Problem (1.1) in its generic form is intractable and additional structural assumptions on $\phi(\boldsymbol{x})$ and $\mathcal{D}$ are necessary. For instance, the constraint set $\mathcal{D}$ may coincide with the entire space $\mathbb{R}^n$, in which case (1.1) reduces to an *unconstrained optimization* problem. Another example is when the function $\phi(\boldsymbol{x})$ exhibits a particular structure like convexity. In this thesis we mainly focus on *convexity*, *smoothness*, *difference-of-convex* (DC) structure, and *block* or *coordinate-separability* which are discussed in Section 2.1 and chapter 3.

Optimization algorithms aim to leverage such assumptions or structural properties to either improve the quality of the solutions they obtain for (1.1) or reduce the computational cost of the optimization process. At the core of these algorithms lie iterative methods.

These methods begin from an initial point and then repeatedly update the iterates in an attempt to locate a local minimizer of the objective, using information gathered around the previous iterate. When these updates rely solely on first-order information (function values and gradients), the resulting procedure is referred to as a *first-order method* [Nesterov, 2003, Section 1.3].

First-order methods differ in how they exploit the assumptions in problem (1.1). For example, certain algorithms are able to minimize the objective in significantly fewer iterations and therefore achieve a lower computational cost. A class of methods that achieve this goal using only first-order information of the objective, are called *first-order accelerated methods* [Nesterov, 2003, Section 2.2]. A more detailed overview on these methods is provided in Section 1.1.

Another example concerns algorithms tailored to objectives with decomposable structures. In the case of DC functions, the *DC algorithm* (DCA) is well known for its simplicity, versatility, and broad applicability [Le Thi and Pham Dinh, 2018]. A more detailed overview to DC programming and the DCA is given in Section 1.2.

This thesis is concerned with understanding how iterative methods exploit these structural properties. In particular, first it studies acceleration through a continuous-time limit of accelerated methods and reveals hidden mechanisms behind acceleration. Second, it focuses on a geometric and compositional perspective, centered on DC programming, which provides a flexible modeling framework for a broad class of nonconvex problems while preserving convex structure at the algorithmic level.

1.1 Overview of First-Order Accelerated Methods

With the recent surge in data availability across statistics, machine learning, and signal processing, there has been growing interest in first-order methods since they rely solely on evaluations of the objective function and its first-order derivatives [Dvurechensky et al., 2021]. In convex optimization, many such methods arise from a sequential approximation strategy, in which the original problem is approached through a sequence of simpler subproblems. Each subproblem consists of minimizing a surrogate function that approximates the objective in a neighborhood of the current iterate. The simplest method in this category for is known as the *gradient descent* (GD) method. In each iteration of the GD, the next point is determined by minimizing an isotropic quadratic upper bound [Nesterov, 2003, Section 1.3]. In a similar fashion, the Frank-Wolfe (FW) algorithm (introduced in Chapter 3) moves toward the minimizer of the first-order Taylor approximation over the constraint set [Frank and Wolfe, 1956]. Note that here we are addressing unconstrained convex optimization.

Accelerated methods do not necessarily follow the sequential-approximation framework described earlier. Instead of minimizing local surrogate functions at each iteration, these algorithms introduce momentum terms that leverage information from previous iterates

to accelerate convergence. This momentum-based perspective underlies most of the first-order accelerated methods examined in this thesis, and unless stated otherwise, our use of the term first-order accelerated methods refers specifically to this class. A different family of accelerated schemes is based on outer–inner (nested) iterations, where an outer loop seeks to accelerate a base algorithm executed in an inner loop; see [d'Aspremont et al., 2021] for a detailed discussion of such approaches.

Two of the earliest methods that do not follow the sequential-approximation framework, are the Heavy-Ball (HB) [Polyak, 1964] and Nesterov's Accelerated Gradient (NAG) [Nesterov, 1983] algorithms. The HB method shares the same overall structure as NAG in the strongly convex setting; the differences lie in the value of a coefficient and in the fact that HB evaluates the gradient at the current iterate, $\boldsymbol{x}_k$. Despite appearing minor, these distinctions have significant consequences: NAG enjoys global stability, whereas HB does not generally achieve global convergence [Lessard et al., 2016, Maskan et al., 2025c]. In Section 2.1, we present and compare the convergence rates of the GD and the NAG methods.

The original analysis of the NAG method is based on an algebraic construction known as the *estimate sequence*. While this concept plays a central role in establishing the convergence guarantees of NAG, it serves primarily as an analytical tool rather than a mechanism used explicitly in the algorithmic construction [Nesterov, 1983]. As a consequence, the original proof is largely algebraic in nature and provides limited intuitive insight into the underlying mechanism responsible for acceleration. This lack of a clear explanatory principle has motivated the development of several complementary perspectives aimed at elucidating the foundations of Nesterov's method.

To date, four main perspectives have emerged. The first is the numerical *performance estimation approach*, pioneered by Drori and Teboulle [2014]. The second is the variational viewpoint developed by Wibisono et al. [2016]. The third is the continuous-time analysis of accelerated methods [Su et al., 2016, Shi et al., 2021]. The fourth is the dynamical-systems interpretation [Muehlebach and Jordan, 2019, Fazlyab et al., 2018]. In Chapter 4, we propose an extension of the variational perspective that builds a bridge to insights from the continuous-time viewpoint, particularly those of [Shi et al., 2019].

1.2 Overview of DC Programming and DCA

Reconsider our generic optimization problem in (1.1). The generality of this optimization task makes it both attractive and hard to solve. Therefore, it is motivating to find a subclass of functions which bring about a balance between the generality and the resolvability of the problem. One class of functions in this category is the DC function class. DC programming has a rich history of developments and adaptations to various scenarios and applications [Le Thi and Pham Dinh, 2018]. A DC function is representable in the form of the difference of two convex functions (see Definition 3.1). Using DC formulation

in (1.1), our problem takes the form of:

$$\min_{\boldsymbol{x}\in\mathcal{D}} \ \phi(\boldsymbol{x}) := f(\boldsymbol{x}) - g(\boldsymbol{x}), \tag{1.2}$$

where $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are convex functions. Many algorithms have addressed problem (1.2). For example, DCA [Tao and An, 1998], Convex–Concave Procedure (CCCP) [Yuille and Rangarajan, 2003], Proximal-Gradient DC Methods [Bolte et al., 2014], smoothing and penalty approaches [Nesterov, 2005], Stochastic/Online DC Methods [Thi et al., 2023]. Next, we briefly explain DCA.

The DCA exploits the DC structure by iteratively replacing the original objective with a convex upper bound obtained by linearizing the second component around the current iterate. Each iteration then consists of solving a convex optimization problem over the feasible set, producing a new iterate that improves the original objective. We postpone a detailed discussion of convergence guarantees to Section 3.1.

The convergence analysis of the DCA was studied under different sets of assumptions. In particular, the results from [Le Thi et al., 2018] assume that the function $\phi(\boldsymbol{x})$ satisfies a Łojasiewicz gradient inequality (see Definition 3.2) whereas [Yurtsever and Sra, 2022, Abbaszadehpeivasti et al., 2023] only require convexity of f and g as well as differentiability of g.

In Chapter 4, we build a connection between the existing convergence results for the FW method and DC programming. This connection builds the foundation for a projection-free method called Dc-Fw. In addition, in Section 4.3, we propose a block coordinate variant of the DCA for problems with separable structure (see Definition 3.3).

1.3 Thesis Objectives

In Section 1.1, we mentioned multiple perspectives that researchers have attempted to show the essence of acceleration in first-order optimization. The first objective of this thesis is to understand the connection between the variational perspective and the continuous-time perspective. This valuable connection enables us to not only find a unified framework for the accelerated methods in the convex and strongly convex regimes, but also guides us to build new momentum-driven stochastic methods (**paper I**).

Building on this unified framework, we formulate a general representation of first-order accelerated methods that is consistent with their continuous-time counterparts. Using Lyapunov functions, we establish convergence rates that recover the known guarantees for NAG and improve upon existing results for the Quasi-Hyperbolic Momentum (QHM) method [Ma and Yarats, 2019]. We further show that this framework extends naturally to the Triple Momentum (TM) method [Van Scoy et al., 2018] (**paper II**).

As discussed in Section 1.2, DC programming provides a natural modeling framework for a broad class of nonconvex optimization problems, while preserving convex structure at

the level of the algorithmic subproblems. In particular, each iteration of the DC algorithm requires the solution of a constrained convex optimization problem, making it amenable to projection-free methods such as the FW algorithm. While the FW method is classically analyzed for convex objectives, its convergence analysis in nonconvex settings relies on closely related assumptions such as bounded curvature constant and smoothness of the objective [Jaggi, 2013, Lacoste-Julien, 2016, Yurtsever and Sra, 2022]. Since any smooth function admits a DC decomposition, this observation enables the integration of DC programming with the FW method into a projection-free framework for solving constrained smooth nonconvex problems. This perspective underlies the Dc-Fw approach developed in **paper III**.

Finally, we introduce an extension of the DCA in the form of a randomized block coordinate approach [Richtárik and Takáč, 2014] for problems with separable structure. Leveraging the connection between the DCA and the Expectation–Maximization (EM) algorithm [Dempster et al., 1977, Yuille and Rangarajan, 2003], we propose a randomized block-coordinate EM algorithm (Block EM), where convergence guarantees are directly inherited from the DCA as a special case (**paper IV**).

Chapter 2

Acceleration & Continuous-Time Perspective

In this chapter, we provide a brief background on the NAG algorithm and its continuous-time analysis. This material forms the foundation for our main results, which will be presented in Section 4.1. The chapter is organized as follows. After a brief background in Section 2.1, we review the initial works that interpret the NAG algorithm through ODEs in Section 2.2. We then introduce the variational perspective in Section 2.3, which extends these insights to non-Euclidean settings. Finally, we discuss high-resolution ODEs and several related developments in the literature in Section 2.4.

2.1 Preliminaries and Background

Here, we briefly introduce the most important structural properties used in this thesis.

Notation 2.1. Throughout this thesis, Euclidean ℓ_2-norm is denoted with $\|.\|$.

Definition 2.1 (Strong convexity)**.** Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex set. The function $\phi : \mathcal{X} \to \mathbb{R}$ is called μ-strongly convex ($\mu \geq 0$) on $\mathcal{X}$ if, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$,

$$\phi(\boldsymbol{x}) \geq \phi(\boldsymbol{y}) + \langle g, \boldsymbol{x} - \boldsymbol{y} \rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad \forall g \in \partial\phi(\boldsymbol{y}). \tag{2.1}$$

The function ϕ is called μ-strongly concave if $-\phi$ is μ-strongly convex.

Definition 2.2 (L-smoothness)**.** A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if its gradient is Lipschitz continuous with constant L for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, that is:

$$\|\nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \tag{2.2}$$

Notation 2.2. The class of convex and smooth functions with Lipschitz constant L are denoted with $\mathcal{F}_L$. For the class of L-smooth and μ-strongly convex functions we use $\mathcal{F}_{\mu,L}$.

Notation 2.3. From now on, whenever we study functions that are only *convex/strongly convex*, we simply set $\phi(\boldsymbol{x}) = f(\boldsymbol{x})$. Equivalently one could set $g(\boldsymbol{x}) = 0$ in (1.2).

Next, we represent the GD method previously mentioned in Section 1.1. Starting from a point $\boldsymbol{x}_0$ in the domain of f, the update of this method after k iterations is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s\nabla f(\boldsymbol{x}_k), \tag{GD}$$

where $s > 0$ is the step-size. The convergence rate of this method is summarized in the following theorem.

Theorem 2.1 (Gradient Descent)**.** *Consider gradient descent with K iterations for solving (1.1) with optimal value f^* when the constraint set $\mathcal{D} = \mathbb{R}^n$.*

If $f \in \mathcal{F}_L$ and $0 \leq s \leq 1/L$, then

$$f(\boldsymbol{x}_K) - f^* \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{2sK}.$$

If $f \in \mathcal{F}_{\mu,L}$ and $s = 1/L$, then

$$f(\boldsymbol{x}_K) - f^* \leq \left(1 - \frac{\mu}{L}\right)^K (f(\boldsymbol{x}_0) - f^*).$$

Proof. See [Nesterov, 2003, Theorems 2.1.14,2.1.15] for the proof. □

Theorem 2.1 shows that GD achieves a convergence rate of $\mathcal{O}(1/K)$ for convex objectives and $\mathcal{O}((1-\mu/L)^K)$ for μ-strongly convex functions. While the latter corresponds to a linear rate of convergence, the former is sublinear. Importantly, these rates are not optimal: classical complexity lower bounds establish that first-order methods can, in principle, achieve faster convergence for both convex and strongly convex problems [Nesterov, 2003, Theorems 2.1.7, 2.1.13]. In the convex case, the optimal rate is $\mathcal{O}(1/K^2)$, while in the strongly convex case the optimal linear rate depends on the square root of the condition number, namely $\mathcal{O}((1 - \sqrt{1/\kappa})^K)$ with $\kappa = L/\mu$ being the condition number of $f \in \mathcal{F}_{\mu,L}$. These limitations motivate the development of accelerated first-order methods.

The NAG method provably achieves convergence rates that match these lower bounds and strictly improves upon those of GD. In particular, the Nesterov family of accelerated methods has the following form:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{y}_k - s\nabla f(\boldsymbol{y}_k), \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_{k+1} + \beta_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k), \end{aligned} \tag{NAG}$$

where $s \leq 1/L$ and β_k changes depending on the properties of the function $f(\boldsymbol{x})$. The following theorem summarizes the corresponding convergence guarantees for the NAG method. Proofs are omitted for brevity; the interested reader is referred to [Nesterov, 2003, Theorem 2.2.2] for full details.

Theorem 2.2 (Nesterov's Accelerated Gradient)**.** *Consider the NAG method with K iterations for solving (1.1) with optimal value f^* when the constraint set $\mathcal{D} = \mathbb{R}^n$.*

If $f \in \mathcal{F}_L, s = 1/L$, and $\beta_k = {k-1}/{k+2}$ ($\beta_0 = 0$), then

$$f(\boldsymbol{x}_K) - f^* \leq \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{2(K+1)^2}.$$

If $f \in \mathcal{F}_{\mu,L}, \kappa = L/\mu$ and $s = 1/L$, and $\beta_k = {(\sqrt{\kappa}-1)}/{(\sqrt{\kappa}+1)}$, then

$$f(\boldsymbol{x}_K) - f^* \leq \frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{2}\left(1 - \sqrt{\frac{\mu}{L}}\right)^K.$$

2.2 Low-Resolution ODEs

In Section 1.1, we outlined several perspectives for understanding the essence of first-order acceleration. One influential line of research examines the continuous-time behavior of accelerated methods by studying the NAG dynamics in the limit of an infinitesimal step size. This viewpoint was first introduced by Alvarez et al. [2002] and later revitalized by Su et al. [2016]. This analysis reveals that for the unconstrained smooth convex optimization ($\mathcal{D} = \mathbb{R}^n, \phi(\boldsymbol{x}) = f(\boldsymbol{x})$ in (1.2)) the discrete iterates of the (NAG) method, $\boldsymbol{x}_k$, can be approximated using a continuous-time trajectory $X(t)$ ($t := k\sqrt{s}$) as the step-size $s \to 0$, where $X(t)$ satisfies the following second-order ODE:

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0, \tag{2.3}$$

while in the strongly convex case ($f(\boldsymbol{x})$ being μ-strongly convex), one has the following second-order ODE:

$$\ddot{X}_t + 2\sqrt{\mu}\dot{X}_t + \nabla f(X_t) = 0. \tag{Polyak ODE}$$

Interestingly, (Polyak ODE) was assigned to the continuous-time behaviour of both the HB and the NAG algorithms for strongly convex functions. In this thesis, these ODEs will be referred to as *low-resolution* ODEs.

2.3 Variational Perspective

More recently, Wibisono et al. [2016] and Wilson et al. [2021] extended these ODEs to the non-Euclidean setting through a variational perspective. They defined the following Lagrangian using the Bregman divergence instead of the ℓ_2-norm:

$$\mathcal{L}(X_t, V_t, t) = e^{\alpha_t + \gamma_t}\left(D_h(X_t + e^{-\alpha_t}V_t, X_t) - e^{\beta_t}f(X_t)\right), \tag{2.4}$$

where $V_t = \dot{X}_t := {}^{dX_t}/_{dt}$ is velocity, $D_h(X, Y)$ is the Bregman divergence for convex function $h(X)$ defines as:

$$D_h(X, Y) = h(X) - h(Y) - \langle \nabla h(Y), X - Y \rangle,$$

and $\alpha_t, \beta_t, \gamma_t : \mathbb{T} \to \mathbb{R}$ are continuously differentiable functions of time, corresponding to the weight of velocity, the potential function f, and the overall damping, respectively. Using variational calculus, the action for the curves $\{X_t : t \in \mathbb{R}\}$ is defined as the functional:

$$\mathcal{A}(X_t) = \int_{\mathbb{R}} \mathcal{L}(X_t, \dot{X}_t, t) dt.$$

Consider the problem of minimizing $\mathcal{A}(X_t)$. A curve is a stationary point for this problem *if and only if* it satisfies the Euler Lagrange equation [Goldstein et al., 2002, Section 1.4]:

$$\frac{d}{dt}\left\{\frac{\partial \mathcal{L}}{\partial \dot{X}_t}(X_t, \dot{X}_t, t)\right\} = \frac{\partial \mathcal{L}}{\partial X_t}(X_t, \dot{X}_t, t).$$

After calculating this for (2.4), they achieved a non-Euclidean extension of (2.3) for specific choices of $\alpha_t, \beta_t, \gamma_t$. Additionally, these parameters satisfied a set of conditions called the *ideal scaling* as below:

$$\dot{\beta}_t \leq e^{\alpha_t}, \ \dot{\gamma}_t = e^{\alpha_t}. \tag{2.5}$$

In the strongly convex regime, a similar analysis using different Lagrangian recovered (Polyak ODE) [Wilson et al., 2021] which was attributed to the continuous-time trajectory of the NAG and the HB methods.

2.4 High-Resolution ODEs

HB and NAG. Although HB and NAG were both corresponded to the same ODE (Polyak ODE), their practical behavior can differ markedly. Lessard et al. [2016] showed that HB may fail to converge under standard choices of β_k, whereas NAG does not suffer from this issue. Motivated by this discrepancy, Shi et al. [2021] introduced the High-Resolution ODE (HR-ODE) framework, deriving ODEs that more accurately approximate the discrete dynamics of each method. More precisely, they derived the following ODEs:

$$\ddot{X}_t + 2\sqrt{\mu}\dot{X}_t + (1 + \sqrt{\mu s})\nabla f(X_t) = 0, \tag{HB-ODE}$$

$$\ddot{X}_t + 2\sqrt{\mu}\dot{X}_t + \sqrt{s}\nabla^2 f(X_t)\dot{X}_t + (1 + \sqrt{\mu s})\nabla f(X_t) = 0, \tag{NAG-ODE}$$

where the (HB-ODE) corresponds to the HB method and the (NAG-ODE) to the NAG algorithm. These two ODEs are different in a Hessian term, present in (NAG-ODE), which is credited to a *gradient correction* term in the NAG algorithm [Shi et al., 2021]. In

a similar fashion, for the convex case, the following HR-ODE for NAG was derived:

$$\ddot{X}_t + \left(\frac{3}{t} + \sqrt{s}\nabla^2 f(X_t)\right)\dot{X}_t + \left(1 + \frac{3\sqrt{s}}{2t}\right)\nabla f(X_t) = 0.$$

Later, [Shi et al., 2019] discretized these HR-ODEs using various numerical integrators and showed that the Semi-Implicit Euler (SIE) discretization leads to an implementable method with accelerated convergence rate. However, their convergence rate was slower than the NAG's up to a constant. Our proposed framework in Section 4.1.2 resolves this issue.

TM Method. To this point, all the analysis was mainly devoted to the NAG and HB methods. One other accelerated method which is provably faster than the NAG algorithm (up to a constant factor), is the TM method, proposed by Van Scoy et al. [2018]. The state space presentation of the TM method is given by:

$$\begin{cases} \boldsymbol{\epsilon}_{k+1} &= (1+\beta)\boldsymbol{\epsilon}_k - \beta\boldsymbol{\epsilon}_{k-1} - \alpha\nabla f(\boldsymbol{y}_k), \\ \boldsymbol{y}_k &= (1+\gamma)\boldsymbol{\epsilon}_k - \gamma\boldsymbol{\epsilon}_{k-1}, \\ \boldsymbol{x}_k &= (1+\delta)\boldsymbol{\epsilon}_k - \delta\boldsymbol{\epsilon}_{k-1}, \end{cases} \tag{TM-Method}$$

with initialization $\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_{-1} \in \mathbb{R}^d$ and the parameters $\alpha = \frac{1+\rho}{L}$, $\beta = \frac{\rho^2}{2-\rho}$, $\gamma = \frac{\rho^2}{(1+\rho)(2-\rho)}$, and $\delta = \frac{\rho^2}{1-\rho^2}$, where $\rho = 1 - \sqrt{1/\kappa}$ and $\kappa = L/\mu$.

A continuous-time analysis of this method was conducted in [Sun et al., 2020] using $\boldsymbol{y}_k = Y(t_k)$ and $t_k = k\sqrt{\alpha}$. They recovered the following ODE:

$$\ddot{Y}_t + 2\sqrt{M}\dot{Y}_t + \gamma(1+\sqrt{M\alpha})\sqrt{\alpha}\nabla^2 f(Y_t)\dot{Y}_t + (1+\sqrt{M\alpha})\nabla f(Y_t) = 0, \tag{HR-TM}$$

where $M = \left(\frac{1-\beta}{\sqrt{\alpha}(1+\beta)}\right)^2$ is a κ-dependent constant. Although Sun et al. [2020] found a different HR-ODE than the (NAG-ODE), their convergence result did not compare to the convergence rate of the TM method. Our analysis in Section 4.1.2 improves the convergence rate of (HR-TM) and proposes a new HR-ODE for the TM method which achieves the convergence rate corresponding to the TM algorithm.

General HR-ODE. In an effort to demonstrate that acceleration is independent of the choice of integrator, Zhang et al. [2021] showed that an appropriate selection of the HR-ODE together with a suitable discretization scheme can yield an accelerated algorithm. Their analysis was based on a *generalized momentum ODE* (GM-ODE), defined as follows:

$$\begin{cases} \dot{U}_t &= -m'\nabla f(U_t) - n'W_t, \\ \dot{W}_t &= \nabla f(U_t) - q'W_t, \end{cases} \tag{GM-ODE}$$

where $m', n', q' \geq 0$, $U_t = U(t)$, and $W_t = W(t)$; which can be equivalently written as a single line equation:

$$\ddot{U}_t + (q' + m'\nabla^2 f(U_t))\dot{U}_t + (n' + m'q')\nabla f(U_t) = 0. \tag{2.6}$$

Particularly, Zhang et al. [2021] utilized (GM-ODE) to approximate (NAG-ODE) in continuous-time. Later, they discretized (GM-ODE) using the SIE integrator and tried to recover the NAG algorithm. Unfortunately, their recovered ODE had a coefficient mismatch with the original (NAG-ODE). Specifically, the following parameters are selected in their continuous-time analysis:

$$m' = \sqrt{s}, \quad q' = 2\sqrt{\mu}, \quad \text{and} \quad n' = 1.$$

Substituting these values into (2.6) gives

$$\ddot{U}_t + (2\sqrt{\mu} + \sqrt{s}\nabla^2 f(U_t))\dot{U}_t + (1 + 2\sqrt{\mu s})\nabla f(U_t) = 0,$$

which differs from (NAG-ODE) in the coefficient of $\nabla f(U_t)$. Additionally, a different set of parameters, given by

$$m' = \sqrt{s}, \quad q' = 2\sqrt{\mu}, \quad \text{and} \quad n' = 1 - 2\sqrt{\mu s}$$

is used with the SIE discretization to recover the NAG algorithm. This creates a subtle discrepancy between their continuous and discrete-time models. Additionally, their convergence results in both continuous and discrete-time did not match to the NAG's convergence rate. In Section 4.1, we propose a new general framework which resolves the discrepancy between continuous and discrete-time models. This framework achieves faster convergence rates than the state of the art.

Chapter 3

DC Algorithm and Beyond

In Section 1.2, we highlighted the importance and generality of the DC structure and DCA. In this chapter, we focus on two important scenarios: first, when the DC optimization task is constrained on a convex compact set $\mathcal{D}$, and second, when the problem has high dimensions. For the first scenario, when projection is expensive, projection-free methods (e.g. the FW algorithm) are preferred for solving DC subproblems. For the second scenario, if the problem exhibits a block-separable structure (see Definition 3.3), it is appealing to use coordinate-type algorithms to solve the DCA subproblems.

Here, we first present the DCA and an overview of its convergence results in Section 3.1. Then, a background on the FW method and its convergence results is given in Section 3.2. We present our block-separable DC problem setup together with a background on coordinate-methods in DC programming in Section 3.3. Finally, the connection between DCA and the EM algorithm is given in Section 3.4.

3.1 DC Algorithm

An important structural assumption in this work is when the objective is a DC function. This structure entails a wide range of functions, including, but not limited to, convex or L-smooth functions. The formal definition of this class is given below:

Definition 3.1 (DC)**.** A function $\phi(\boldsymbol{x})$ is difference of convex if there exist convex functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ such that $\phi(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x})$.

Here, we briefly explain the DCA and its convergence results . At each iteration k, DCA selects a subgradient $\boldsymbol{u}^k \in \partial g(\boldsymbol{x}^k)$. By the convexity of $g(\boldsymbol{x})$, we have the following upper-bound on the objective $\phi(\boldsymbol{x})$:

$$\hat{\phi}(\boldsymbol{x}) \;:=\; f(\boldsymbol{x}) - g(\boldsymbol{x}^k) - \langle \boldsymbol{u}^k, \boldsymbol{x} - \boldsymbol{x}^k \rangle.$$

Then, DCA solves (1.2) through sequential minimization of the r.h.s. of this inequality,

as below:

$$\boldsymbol{x}^{k+1} \in \arg\min_{x \in \mathcal{D}} \; f(\boldsymbol{x}) - \langle \boldsymbol{u}^k, \boldsymbol{x} \rangle.$$

The non-asymptotic convergence of the DCA to a stationary point is studied in two main tracks. The first track assumes *Łojasiewicz gradient inequality* to hold at all stationary points for the function $\phi(\boldsymbol{x})$ [Le Thi et al., 2018]. The second track relies on more basic assumptions like differentiability and convexity [Yurtsever and Sra, 2022, Abbaszadehpeivasti et al., 2023].

Here, we present Łojasiewicz gradient inequality and a simplified convergence result from [Le Thi et al., 2018] inspired by [Abbaszadehpeivasti et al., 2023].

Definition 3.2 (Łojasiewicz gradient inequality)**.** A differentiable function $\phi(\boldsymbol{x})$ is said to satisfy Łojasiewicz gradient inequality if at a stationary point $\boldsymbol{a}$ there exists constants $\theta \in (0,1), C > 0$ and $\epsilon > 0$ such that:

$$|\phi(\boldsymbol{x}) - \phi(\boldsymbol{a})|^\theta \leq C \|\nabla \phi(\boldsymbol{x})\| \quad \text{if} \quad \|\boldsymbol{x} - \boldsymbol{a}\| \leq \epsilon, \tag{3.1}$$

where the constant θ is called the Łojasiewicz constant.

Theorem 3.1 (Theorems 3.4 and 3.5 in [Le Thi et al., 2018])**.** *Let $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ be proper convex functions such that the domain of ϕ be closed, at least one of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ is strongly convex, and $f(\boldsymbol{x})$ or $g(\boldsymbol{x})$ is differentiable with locally Lipschitz gradient in every critical point of the DC problem. Finally, assume that the DCA-generated sequence $\boldsymbol{x}_k$ is bounded, and let $\boldsymbol{x}_\infty$ be a limit point of $\boldsymbol{x}_k$. Then $\boldsymbol{x}_\infty$ is a stationary point. Moreover, if ϕ satisfies the Łojasiewicz gradient inequality (3.1) at all stationary points, then*

- *if $\theta \in (1/2, 1)$, then $\|\boldsymbol{x}_k - \boldsymbol{x}_\infty\| \leq c k^{\frac{1-\theta}{1-2\theta}}$ for some $c > 0$.*
- *if $\theta \in (0, 1/2)$, then $\|\boldsymbol{x}_k - \boldsymbol{x}_\infty\| \leq c q^k$ for some $c > 0$ and $q \in (0,1)$.*

Many classes of functions satisfy the inequality (3.1). In particular, every real-analytic function satisfies this property locally around each of its critical points [Bolte et al., 2007]. One issue, which is less attended in the literature, is on the value of θ and how to determine it. There exists other convergence results without needing this inequality.

Theorem 3.2 ([Yurtsever and Sra, 2022])**.** *Consider $\boldsymbol{x}_k$ as the sequence generated by DCA when applied on problem (1.2) with stationary point $x^\star$. Then, there exists $\tau \in \{K\}$ such that for any $\boldsymbol{x} \in \mathbb{R}^n$ we have:*

$$f(\boldsymbol{x}_\tau) - f(\boldsymbol{x}) - \langle \nabla g(\boldsymbol{x}_\tau), \boldsymbol{x}_\tau - \boldsymbol{x} \rangle \leq \frac{1}{K} \left(f(\boldsymbol{x}_1) - g(\boldsymbol{x}_1) - (f(\boldsymbol{x}^\star) - g(\boldsymbol{x}^\star)) \right). \tag{3.2}$$

In this thesis, our convergence results follow the same spirit as Theorem 3.2. We generalize the left-hand side of (3.2) to obtain a measure of convergence that characterizes approach to a critical point when the function g is non-differentiable (see Definition 4.1 and Lemma 4.1). This result is new and was not explored in [Yurtsever and Sra, 2022,

Abbaszadehpeivasti et al., 2023].

3.2 Frank-Wolfe Algorithm

For better readability, we restate the problem (1.2) below:

$$\min_{\boldsymbol{x}\in\mathcal{D}} \ \phi(\boldsymbol{x}) := f(\boldsymbol{x}) - g(\boldsymbol{x}),$$

where the constraint set $\mathcal{D}$ is a convex compact set with diameter D. Further, assume $f(\boldsymbol{x})$ to be L-smooth, but $g(\boldsymbol{x})$ may be nonsmooth. In some cases, it is possible to efficiently project onto the constraint set $\mathcal{D}$, in which case projected gradient methods may be employed. For example, when the constraint set is an ℓ_2-norm ball or a box constraint. However, in many applications the projection step is computationally expensive or impractical. Examples of this scenario are nuclear norm balls or flow polytope constraint sets [Combettes and Pokutta, 2021, Table 1]. In such settings, projection-free methods—most notably FW–type algorithms—can offer substantial computational benefits. We now present a brief overview of the FW algorithm. For a comprehensive overview of other developments and variants of this method, we refer the interested reader to [Kerdreux, 2020].

The updates of the FW algorithm are as follows:

$$\begin{aligned} \boldsymbol{s}_k &= \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{D}} \ \langle \nabla\phi(\boldsymbol{x}_k), \boldsymbol{x}\rangle \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \eta_k(\boldsymbol{s}_k - \boldsymbol{x}_k), \end{aligned} \tag{FW}$$

where $\eta_k \in [0,1]$ is the step-size. There are various step-size strategies in the literature, common choices are $\eta_k = 2/(k+1)$, the exact line-search

$$\eta_k = \operatorname*{argmin}_{\eta\in[0,1]} \ \phi(\boldsymbol{x}_k + \eta(\boldsymbol{s}_k - \boldsymbol{x}_k)), \tag{3.3}$$

and the Dem'yanov and Rubinov [1970] step-size

$$\eta_k = \min\Big\{\frac{\langle \nabla\phi(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{s}_k\rangle}{L\|\boldsymbol{x}_k - \boldsymbol{s}_k\|^2}, 1\Big\}. \tag{3.4}$$

The following result from [Jaggi, 2013] formulates the convergence result for solving convex problems (when $g(\boldsymbol{x}) = 0$) using the FW algorithm.

Lemma 3.1 (Theorem 1 in [Jaggi, 2013])**.** *Suppose ϕ is a proper, lower semi-continuous, convex and L-smooth function. Consider the problem of minimizing ϕ over a convex and compact set $\mathcal{D}$ of diameter D. Then, the sequence $\{\boldsymbol{x}_k\}$ generated by the FW algorithm after K iterations satisfies*

$$\phi(\boldsymbol{x}_k) - \phi(\boldsymbol{x}^\star) \leq \frac{2LD^2}{K+1},$$

where $\boldsymbol{x}^\star \in \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{D}} \phi(\boldsymbol{x})$. This result holds for $\eta_k = 2/(k+1)$, the line-search, and the

Demyanov-Rubinov step-size.

In a more general analysis, Lacoste-Julien [2016] established nonasymptotic convergence rates for smooth nonconvex optimization problems. For nonconvex objectives, the conventional suboptimality gap $\phi(\boldsymbol{x})-\phi(\boldsymbol{x}^*)$ is not an appropriate measure of progress, as $\phi(\boldsymbol{x}^*)$ is typically unknown. In unconstrained nonconvex optimization, the gradient norm $\|\nabla\phi(\mathbf{x})\|$ serves as a natural first-order stationarity measure. However, when the optimization problem is constrained to a set, *affine-invariant* measures[1] are preferred [Lacoste-Julien, 2016]. Therefore, the following function, known as the FW gap function, is a proper measure for first-order stationarity in constrained nonconvex optimization:

$$\text{gap}_{\text{FW}}(\boldsymbol{x}_t) := \max_{\boldsymbol{x}\in\mathcal{D}} \Big\{\langle\nabla\phi(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}\rangle\Big\}. \tag{3.5}$$

The reason that (3.5) is called the FW gap function is due to the fact that we solve this problem in the linear minimization step of the FW algorithm (see (FW)). The following result from [Lacoste-Julien, 2016] presents the convergence result of the FW method for constrained nonconvex optimization tasks.

Lemma 3.2 (Theorem 1 in [Lacoste-Julien, 2016])**.** *Consider problem* (1.2) *when* $\phi(\boldsymbol{x})$ *is any nonconvex function (not necessarily DC), continuously differentiable, and* L*-smooth. Then, if we use the* (FW) *algorithm for* K *iterations with the step size* $\eta_k = 2/(k+1)$*, we have:*

$$\min_{0\le k\le K} \text{gap}_{FW}(\boldsymbol{x}_k) \le \frac{\max\{2(\phi(\boldsymbol{x}^0) - \phi(\boldsymbol{x}^\star))), LD^2\}}{\sqrt{K+1}}. \tag{3.6}$$

The result in Lemma 3.2 is valid for any L-smooth noncovnex objective. This means that it does not necessarily incorporate the DC structure attributed to smoothness into the problem. In Section 4.2, we propose an algorithm which benefits from this structure.

Yurtsever and Sra [2022] explored the connection between DC programming and the FW method through an epigraph reformulation of the DC program. Using this connection, they built a convergence rate for the DCA by minimizing concave objectives. Interestingly, their results showed a convergence rate of $\mathcal{O}(1/K)$ for the FW method when dealing with nonsmooth concave objectives (see Theorem 3.2). They used a DC-based gap function as their stationarity measure. We propose a more general form of this gap function in Sections 4.2 and 4.3.

An important extension of the FW method is the conditional gradient sliding (CGS) algorithm [Lan and Zhou, 2016]. Rather than applying FW directly to the original problem, CGS constructs an inexact accelerated projected gradient method and uses FW to approximately solve the projection subproblems. This approach preserves the same linear-minimization oracle complexity as FW, namely $\mathcal{O}(1/\epsilon)$, while achieving a lower gradient oracle complexity of $\mathcal{O}(1/\sqrt{\epsilon})$. We will further explore the relationship between this extension (in the nonconvex setting) and the DCA in Section 4.2.

[1] Measures that remain unchanged under affine transformations of the constraint set.

3.3 Randomized Coordinate DC Programs

When dealing with high-dimensional problems, methods like DCA could suffer from computational costs. In some problems (e.g. nonconvex quadratic programming with negative ℓ_1 regularization [Maskan et al., 2025a]), coordinate analysis of the DC program plays a crucial role in tractability of the optimization task. Here, we address a more general formulation than (1.2) which is the sum of a DC function with a convex block-separable term defined below. Note that when dealing with randomized coordinate methods, we use boldface lowercase letters to denote vectors and boldface uppercase to denote matrices.

Definition 3.3 (block-separability)**.** A function $h(\boldsymbol{x})$ is block or coordinate separable if $h(\boldsymbol{x}) = \sum_i^n h_i(x_i)$ for $h_i : \mathbb{R} \to \mathbb{R}$ and x_i being the i^{th} coordinate of $\boldsymbol{x}$.

Our optimization task is of the form:

$$\min_{\boldsymbol{x}\in\mathcal{D}} \quad \phi(\boldsymbol{x}) := f(\boldsymbol{x}) + h(\boldsymbol{x}) - g(\boldsymbol{x}), \tag{3.7}$$

where f, h, and g are lower semi-continuous convex functions defined on the closed convex set $\mathcal{D} \subseteq \mathbb{R}^n$. We do not impose the smoothness assumption on the terms g and h, yet we assume that the function f is smooth, and that the function h and the set $\mathcal{D}$ exhibit a block-separable structure in relation to the coordinates of $\boldsymbol{x}$:

$$h(\boldsymbol{x}) = \sum_{i=1}^{N_b} h_i(\boldsymbol{D}_i\boldsymbol{x}),$$

where $\boldsymbol{D}_i$ represents $(n_i \times n)$ dimensional row subsets of the identity matrix $\boldsymbol{I}$, serving as the selection operator for a non-overlapping partition of the coordinates into N_b blocks:

$$\sum_{i=1}^{N_b} \boldsymbol{D}_i^\top \boldsymbol{D}_i = \boldsymbol{I} \quad \text{and} \quad \sum_{i=1}^{N_b} n_i = n.$$

Similarly, $\mathcal{D}$ can be decomposed as $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$, where the components $\mathcal{D}_i \in \mathbb{R}^{n_i}$, such that

$$\boldsymbol{x} \in \mathcal{D} \iff \boldsymbol{D}_i\boldsymbol{x} \in \mathcal{D}_i \quad \text{for} \quad i = 1, \ldots, N_b.$$

For the ease of presentation, we define the following coordinate vector notation:

$$x_i = \boldsymbol{D}_i\boldsymbol{x}, \quad \boldsymbol{x}_i = \boldsymbol{D}_i^\top \boldsymbol{D}_i\boldsymbol{x}, \quad \text{and} \quad \bar{\boldsymbol{x}}_i = (\boldsymbol{I} - \boldsymbol{D}_i^\top \boldsymbol{D}_i)\boldsymbol{x}.$$

Here, $x_i \in \mathbb{R}^{n_i}$ represents the i^{th} coordinate block of $\boldsymbol{x} \in \mathbb{R}^n$. The vector $\boldsymbol{x}_i$ is an extension of x_i to $\mathbb{R}^m$ with all other blocks padded with zeros. Conversely, $\bar{\boldsymbol{x}}_i$ is the complement of $\boldsymbol{x}_i$, containing zeros in the i^{th} block and ensuring that $\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i = \boldsymbol{x}$.

Early works on successive upper-bound minimization (which includes DCA) already exploited coordinate-type methods to reduce per-iteration complexity (see [Le Thi and Pham Dinh, 2018, Section 3.3] and [Razaviyayn et al., 2013]), but modern non-asymptotic convergence guarantees are primarily developed through connections to randomized coor-

dinate descent [Richtárik and Takáč, 2014, Nesterov and Stich, 2016]. Recent extensions of these guarantees to nonconvex and DC-type objectives appear in, e.g., [Deng and Lan, 2020, Cai et al., 2023]. From this viewpoint, coordinate descent can be interpreted as a scalable mechanism for approximately solving DCA subproblems, making it particularly well suited for high-dimensional DC problems where full-dimensional convex subproblems are computationally prohibitive. With the exception of [Deng and Lan, 2020], most existing works on block-coordinate nonconvex optimization rely on the standard gradient-norm stationarity measure, which is not appropriate for the constrained nonconvex setting in (3.7). Further details on difference convergence measures used in recent coordinate-DC works can be found in [Maskan et al., 2025a].

In Section 4.3, we study a randomized block-coordinate DC algorithm for solving problem (3.7) and provide a non-asymptotic convergence analysis for the proposed method. Moreover, the coordinate-wise structure considered in our algorithm necessitates a different stationarity measure than earlier works such as [Yurtsever and Sra, 2022]. Motivated by this, we propose a novel block-coordinate separable stationarity measure, formally introduced in Section 4.3.

3.4 DCA and the EM Method

Earlier, we highlighted the generality of DC programs and their associated algorithm, the DCA. In this context, DCA recovers the well-known EM algorithm when applied to exponential-family models [Yuille and Rangarajan, 2003]. This connection is valuable for two reasons. First, it enables the design of randomized block-coordinate DC methods that give rise to new EM variants with reduced per-iteration computational cost. Second, it provides a pathway for establishing concrete non-asymptotic convergence rates for the EM algorithm through its relationship with DCA. These ideas are developed in detail in Section 4.3. In what follows, we briefly recall the EM algorithm and present its connection to DCA.

One of the most prominent methods for computing maximum likelihood estimates in the presence of incomplete data is the EM algorithm. It is widely used in applications such as mixture models, latent variable models, data imputation, and clustering [Dempster et al., 1977, McLachlan and Krishnan, 2007]. Consider negative log-likelihood minimization problem:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(\boldsymbol{\theta}) := -\sum_{\boldsymbol{x}\in\mathcal{X}} \log P(\boldsymbol{x}|\boldsymbol{\theta}). \tag{3.8}$$

Following standard notation in the statistics literature, we make a slight modification: here, $\boldsymbol{x}$ denotes the data points, while the decision variable $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^n$ represents the parameters of the underlying distribution. The dataset $\mathcal{X}$ has support size N, and $\mathcal{L}$ denotes the negative log-likelihood function. The distribution $P(\boldsymbol{x}|\boldsymbol{\theta})$ is defined through a hidden variable $\boldsymbol{y}$:

$$P(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{y}} P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}),$$

where $P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ denotes the likelihood of the complete-data. Due to the hidden variables in (3.8), standard optimization techniques are inapplicable.

The EM algorithm iteratively alternates between two steps – Expectation (E-step) and Maximization (M-step) – to improve the likelihood of the model parameters until convergence:

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k) &= \mathbb{E}_{P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^k)} \left[\log P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\right] && \text{(E-step)} \\ \boldsymbol{\theta}^{k+1} &= \operatorname*{arg\,max}_{\boldsymbol{\theta} \in \Theta} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k) && \text{(M-step)} \end{aligned}$$

Here, $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^k)$ is the posterior distribution of the latent variables given the observed data and the current parameter estimate $\boldsymbol{\theta}^k$. In the E-step, we compute the expected value of the complete-data log-likelihood with respect to $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^k)$. In the M-step, this expected log-likelihood is maximized to obtain an updated parameter estimate.

EM is an instance of DCA for the class of natural exponential family distributions.

Definition 3.4. The natural exponential family of distributions with a parameter $\boldsymbol{\theta} \in \Theta$ has the following probability mass (or density) function:

$$P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = \frac{\varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}{\sum_{\boldsymbol{x}, \boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}, \tag{3.9}$$

where φ is a positive function (base measure) and T denotes the sufficient statistic.

Problem (3.8) with a natural exponential family distribution (3.9) can be written as a DC program through the decomposition of the log-likelihood function as $\mathcal{L}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) - g(\boldsymbol{\theta})$, where the terms are given by

$$f(\boldsymbol{\theta}) = N \log \Big(\sum_{\boldsymbol{x}, \boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle} \Big), g(\boldsymbol{\theta}) = \sum_{x \in \mathcal{X}} \log \Big(\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle} \Big).$$

Both terms are convex and smooth, owing to their log-sum-exp structure. Applying DCA to this problem leads to the following update rule:

$$\boldsymbol{\theta}^{k+1} \in \operatorname*{argmin}_{\boldsymbol{\theta}} \; N \log \left(\sum_{\boldsymbol{x}, \boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle} \right) - \sum_{x \in \mathcal{X}} \frac{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) \langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}.$$

After algebraic simplifications, this reduces to:

$$\boldsymbol{\theta}^{k+1} \in \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{x \in \mathcal{X}} \sum_{\boldsymbol{y}} \frac{\varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}^k, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}{\sum_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}} \times \left[\log \left(\frac{\sum_{\boldsymbol{x}, \boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}}{\varphi(\boldsymbol{x}, \boldsymbol{y}) e^{\langle \boldsymbol{\theta}, T(\boldsymbol{x}, \boldsymbol{y}) \rangle}} \right) \right],$$

which is exactly the EM algorithm for the exponential family.

Due to its widespread applicability, the convergence analysis of the EM algorithm has been an active and important area of research. The results most relevant to this thesis

are [Kumar and Schmidt, 2017, Kunstner et al., 2020]. From a majorization–minimization perspective, Kumar and Schmidt [2017] provided convergence guarantees for EM. Their analysis assumes that the surrogate optimization problems are strongly convex, a condition that may fail to hold in general, even for exponential-family models. Nonetheless, they established that the EM algorithm converges at a rate of $\mathcal{O}(1/K)$ in terms of the squared gradient norm of the negative log-likelihood.

More recently, Kunstner et al. [2020] examined the non-asymptotic convergence of EM for exponential-family distributions by establishing a connection to mirror descent. Their analysis proved a convergence rate of $\mathcal{O}(1/K)$ in Kullback–Leibler (KL) divergence and related this to first-order stationarity through Bregman divergences.

In Section 4.3, we extend these ideas to a block-coordinate DC algorithm. Through this connection, we propose a new block-coordinate EM method with non-asymptotic convergence guarantees to a first-order stationary point.

Chapter 4

Main Results

In this chapter, we build on the background presented in the previous chapters and introduce our new results within each topic. For complete proofs, extended discussions, and numerical experiments, we refer the reader to the original articles appended to this thesis. This chapter is organized as follows. We begin by presenting the connection between the variational perspective and high-resolution ODEs in Section 4.1. This connection enables both faster convergence rates and a new general framework for first-order accelerated methods. Second, we examine constrained DC nonconvex problems solved via the FW method, highlight several new links to existing literature, and propose a new algorithmic variant in Section 4.2. Finally, we introduce a randomized coordinate descent version of the DCA tailored for high-dimensional problems in Section 4.3. Leveraging its relationship with the EM method, we also propose a coordinate-wise EM variant with non-asymptotic convergence guarantees.

4.1 A Variational Perspective on HR-ODEs

In Section 2.3 we presented the variational perspective on accelerated methods. This perspective was mainly dealing with the low-resolution ODEs discussed in Section 2.2. Therefore, it remains an open question if equations like (NAG-ODE) or (HB-ODE) can be seen through the lens of a variational perspective. Furthermore, an intriguing question is whether NAG itself can be recovered through a suitable discretization of the NAG-ODE.

To answer these questions, we propose a novel extension of the variational perspective for HR-ODEs which includes an external non-conservative force F, representing effects such as dissipation or external driving that cannot be captured by a potential. When such a dissipative force is present, the *forced Euler-Lagrange* equation should be used [Goldstein et al., 2002, Section 1.5]:

$$\frac{d}{dt}\left\{\frac{\partial \mathcal{L}}{\partial \dot{X}_t}(X_t, \dot{X}_t, t)\right\} - \frac{\partial \mathcal{L}}{\partial X_t}(X_t, \dot{X}_t, t) = F(X_t, \dot{X}_t, t), \tag{4.1}$$

which itself is the result of integration by parts of Lagrange d'Alembert principle. We

select the external forces to be damped time derivative of the potential function gradients. To recover the existing HR-ODEs in the literature, three different external forces (two for the convex functions and one in the strongly convex regime) are proposed.

In the convex regime, we consider the following Lagrangian:

$$\mathcal{L}(X_t, \dot{X}_t, t) = e^{\alpha_t+\gamma_t}\left(\frac{1}{2}\|e^{-\alpha_t}\dot{X}_t\|^2 - e^{\beta_t}f(X_t)\right). \tag{4.2}$$

where $\alpha_t, \beta_t, \gamma_t : \mathbb{T} \to \mathbb{R}$ are continuously differentiable functions of time, corresponding to the weights of velocity, the potential function f, and the overall damping, respectively. In this work, we study the following possible choices of F for convex functions:

$$F_1 = -\sqrt{s}e^{\gamma_t}\frac{d}{dt}[e^{-\alpha_t}\nabla f(X_t)], \quad F_2 = -\sqrt{s}e^{\gamma_t-\beta_t}\frac{d}{dt}\left[e^{-(\alpha_t-\beta_t)}\nabla f(X_t)\right].$$

Replacing F_1 and F_2 in (4.1) for the Lagrangian in (4.2), gives

$$\ddot{X}_t + (\dot{\gamma}_t - \dot{\alpha}_t)\dot{X}_t + e^{2\alpha_t+\beta_t}\nabla f(X_t) = -\sqrt{s}e^{\alpha_t}\frac{d}{dt}\left[e^{-\alpha_t}\nabla f(X_t)\right], \tag{4.3}$$

and

$$\ddot{X}_t + (\dot{\gamma}_t - \dot{\alpha}_t)\dot{X}_t + e^{2\alpha_t+\beta_t}\nabla f(X_t) = -\sqrt{s}e^{\alpha_t-\beta_t}\frac{d}{dt}[e^{-(\alpha_t-\beta_t)}\nabla f(X_t)], \tag{4.4}$$

respectively. The convergence rate for each of these ODEs requires a dedicated analysis. A key element of our approach is the construction of a suitable Lyapunov function for the corresponding ODE, which acts as a non-increasing energy measure along the trajectory and ensures convergence. See [Siegel, 2019, Shi et al., 2019, Attouch et al., 2020, 2021] for related uses of Lyapunov functions in optimization.

Theorem 4.1 (Convergence for F_1)**.** *Let $f \in \mathcal{F}_L$. If the ideal scaling conditions $\dot{\beta}_t \le \dot{\gamma}_t = e^{\alpha_t}$ hold, then the trajectory X_t from (4.3) satisfies*

$$f(X_t) - f(x^\star) \le \mathcal{O}(e^{-\beta_t}).$$

Proof. See [Maskan et al., 2024]. □

Theorem 4.2 (Convergence for F_2)**.** *Let $f \in \mathcal{F}_L$. If the modified ideal scaling conditions*

$$\dot{\beta}_t \le \dot{\gamma}_t = e^{\alpha_t} \quad \text{and} \quad \ddot{\beta}_t \le e^{\alpha_t}\dot{\beta}_t + 2\dot{\alpha}_t\dot{\beta}_t$$

hold, then the trajectory X_t from (4.4) satisfies

$$f(X_t) - f(x^\star) \le \mathcal{O}\left((e^{\beta_t} + \sqrt{s}e^{-2\alpha_t}\dot{\beta}_t)^{-1}\right).$$

Proof. See [Maskan et al., 2024]. □

In the strongly convex regime, we utilize the following Lagrangian:

$$\mathcal{L}(X_t, \dot{X}_t, t) = e^{\alpha_t+\beta_t+\gamma_t}\left(\frac{Ce^{-\alpha_t}}{2}\|\dot{X}_t\|^2 - f(X_t)\right), \tag{4.5}$$

with $\alpha_t = \alpha$ being a constant. We propose the following external force:

$$F = -C\sqrt{s}e^{\gamma_t}\frac{d}{dt}\left(e^{\beta_t}\nabla f(X_t)\right).$$

Substituting this force into (4.1) for the Lagrangian in (4.5), we get:

$$\ddot{X}_t + (\dot{\gamma}_t + \dot{\beta}_t)\dot{X}_t + \sqrt{s}\nabla^2 f(X_t)\dot{X}_t + \left(\frac{e^{\alpha}}{C} + \sqrt{s}\dot{\beta}_t\right)\nabla f(X_t) = 0. \tag{4.6}$$

The next theorem establishes the convergence guarantees for the trajectory X_t to the unique minimizer $\boldsymbol{x}^\star$.

Theorem 4.3. *Let $f \in \mathcal{F}_{\mu,L}$. We consider two different parameter settings:*
(a) If the following scaling conditions hold:

$$\dot{\gamma}_t = \dot{\gamma}, \quad \textit{such that} \quad 0 \le \dot{\beta}_t \le \dot{\gamma} = e^{\alpha} \le \tfrac{\mu}{C}, \tag{4.7}$$

then the trajectory X_t from (4.6) satisfies

$$f(X_t) - f(x^\star) \le \mathcal{O}(e^{-\beta_t}).$$

(b) If the following modified scaling conditions hold:

$$\dot{\gamma}_t = \dot{\gamma}, \quad \textit{such that} \quad \dot{\gamma} = e^{\alpha} \le 2\dot{\beta} \le \tfrac{2\mu}{C}, \quad \textit{and} \quad C\sqrt{s} \le 2 \tag{4.8}$$

then the trajectory X_t from (4.6) satisfies

$$f(X_t) - f(x^\star) \le \mathcal{O}(e^{-\gamma_t}). \tag{4.9}$$

Proof. See [Maskan et al., 2025c]. □

In what follows, we briefly mention some of the implications of these formulations and results.

4.1.1 Convex Regime

External Force $F = F_1$: Wibisono et al. [2016] showed that discretizing the (2.3) using the rate-matching technique gives an accelerated method:

$$\begin{cases} \boldsymbol{x}_{k+1} = \frac{2}{k+2}\boldsymbol{z}_k + \frac{k}{k+2}\boldsymbol{y}_k, \\ \boldsymbol{y}_k = \boldsymbol{x}_k - s\nabla f(\boldsymbol{x}_k), \\ \boldsymbol{z}_k = \boldsymbol{z}_{k-1} - \frac{1}{2}sk\nabla f(\boldsymbol{y}_k), \end{cases} \tag{4.10}$$

which has a convergence rate of $\mathcal{O}(1/(sk^2))$. In the following proposition, we analyze the behavior of (4.10) in the limit as $s \to 0$.

Proposition 4.1. *The continuous-time behavior of (4.10) is given by*

$$\ddot{X}_t + \left(\frac{3}{t} + \sqrt{s}\nabla^2 f(X_t)\right)\dot{X}_t + \left(1 + \frac{\sqrt{s}}{2t}\right)\nabla f(X_t) = 0. \tag{4.11}$$

We also consider the following modified HR-ODE:

$$\ddot{X}_t + \left(\frac{3}{t} + \sqrt{s}\nabla^2 f(X_t)\right)\dot{X}_t + \left(1 + \frac{\sqrt{s}}{t}\right)\nabla f(X_t) = 0, \tag{4.12}$$

which is obtained by replacing the last term in (4.11) using the approximation $(k+1)/(k+1/2) \approx (k+1)/(k)$.

Proof. See [Maskan et al., 2025c]. □

The ODE (4.12) coincides with (4.3) for

$$\alpha_t = \log(n(t)), \quad \beta_t = \log(q(t)/n(t)), \quad \dot{\gamma}_t = e^{\alpha_t} = n(t), \tag{4.13}$$

$n(t) = p/t$, $q(t) = Cpt^{p-1}$, $p = 2$, and $C = 1/4$.

External Force $F = F_2$: Reformulating (4.4) with the parameters α_t, β_t and γ_t defined as in (4.13) gives:

$$\begin{cases} \dot{X}_t = n(t)(V_t - X_t) - \sqrt{s}\nabla f(X_t) \\ \dot{V}_t = -q(t)\nabla f(X_t) - \sqrt{s}\dfrac{\dot{q}(t)n(t) - \dot{n}(t)q(t)}{n^2(t)q(t)}\nabla f(X_t). \end{cases} \tag{4.14}$$

Applying the SIE discretization to this ODE for $X_t \approx X(t_k)$, $V_t \approx V(t_k)$, and setting $n(t_k) = p/t_k$, $q(t_k) = Cpt_k^{p-1}$ with $p = 2$, $t_k = k\sqrt{s}$, and $C = 1/4$, we obtain

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{2}{k}(\boldsymbol{v}_k - \boldsymbol{x}_{k+1}) - s\nabla f(\boldsymbol{x}_k), \\ \boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \frac{1}{2}sk\nabla f(\boldsymbol{x}_{k+1}) - s\nabla f(\boldsymbol{x}_{k+1}), \end{cases} \tag{4.15}$$

which exactly recovers the NAG algorithm. This precise connection motivated our choice of the Lyapunov function in the next result, leading to an improved convergence rate.

Theorem 4.4. *Let $f \in \mathcal{F}_L$, and consider the algorithm defined in (4.15), initialized with $\boldsymbol{v}_0 = \boldsymbol{x}_1 \in \mathbb{R}^d$ and a step-size $0 < s \leq 1/L$. Then, the sequence $\boldsymbol{x}_k$ satisfies*

$$\frac{K^3 s^2}{12} \min_{0 \leq i \leq K-1} \|\nabla f(\boldsymbol{x}_i)\|^2 \leq \|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2, \tag{4.16}$$

$$\frac{sK(K+2)}{2} f(\boldsymbol{x}_K) - f(\boldsymbol{x}^\star) \leq \|\boldsymbol{x}_1 - \boldsymbol{x}^\star\|^2. \tag{4.17}$$

Proof. See [Maskan et al., 2024]. □

Remark 4.1. The rate in Theorem 4.4 improves upon the previously established rate in [Shi et al., 2021], which is given by

$$\min_{0\le i\le K} \|\nabla f(\boldsymbol{x}_i)\|^2 \le \frac{8568}{(K+1)^3 s^2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2, \tag{4.18}$$

for $0 < s \le 1/(3L)$. This improvement not only reduces the constant but also relaxes the conditions on the step-size.

To address more realistic scenarios, we also studied a stochastic extension of (4.15), where the source of randomness arises from the gradients. We model the noisy gradients by adding i.i.d. noise terms $\boldsymbol{e}_k$ with variance σ^2 to the true gradients. Then, the final update would be:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{2s_k}{t_k}(\boldsymbol{v}_k - \boldsymbol{x}_{k+1}) - \frac{\beta s_k}{\sqrt{L}}(\nabla f(\boldsymbol{x}_k) + \boldsymbol{e}_k), \\ \boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \frac{1}{2}(t_k s_k + \frac{2s_k\beta}{\sqrt{L}})(\nabla f(\boldsymbol{x}_{k+1}) + \boldsymbol{e}_{k+1}) \end{cases} \tag{4.19}$$

with $\beta \ge 2$. This update reduces to (4.15) when $\boldsymbol{e}_k = 0, s_k = \sqrt{s} = \beta/\sqrt{L}, t_k = k\sqrt{s}$. The convergence analysis of this method is given by Maskan et al. [2024, Theorem 5.1]. For numerical experiments as well as other stochastic variants of (4.19) see [Maskan et al., 2024].

4.1.2 Strongly Convex Regime

For simplicity, we reparametrize (4.6) using $\alpha = \log n$, $\beta = qt$, $\gamma = nt$, $m = \sqrt{s}$, and $C = 1/p$, where the constants m, n, p and q are non-negative. The result can be restated in the compact form as follows:

$$\begin{cases} \dot{X}_t = -m\nabla f(X_t) - n(X_t - V_t) \\ \dot{V}_t = -p\nabla f(X_t) - q(V_t - X_t) \end{cases} \tag{GM2-ODE}$$

By analyzing (GM2-ODE), which unifies (HB-ODE) and (NAG-ODE), we obtain convergence rates that improve upon those previously reported in the literature [Shi et al., 2021, Zhang et al., 2021, Shi et al., 2019]. Moreover, (GM2-ODE) captures the continuous-time behavior of several other accelerated methods (see Table 4.1), and yields new HR-ODE formulations for additional algorithms such as the TM method.

By applying SIE discretization to (GM2-ODE), we introduce the following iterative method:

$$\begin{cases} \boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -m\sqrt{s}\nabla f(\boldsymbol{x}_k) - n\sqrt{s}(\boldsymbol{x}_{k+1} - \boldsymbol{v}_k), \\ \boldsymbol{v}_{k+1} - \boldsymbol{v}_k = -p\sqrt{s}\nabla f(\boldsymbol{x}_{k+1}) - q\sqrt{s}(\boldsymbol{v}_k - \boldsymbol{x}_{k+1}). \end{cases} \tag{GM2}$$

The following theorem provides convergence guarantees for (GM2).

Theorem 4.5. *Let $f \in \mathcal{F}_{\mu,L}$, and consider the iterative method described in (GM2) for K iterations, initialized with an arbitrary point $\boldsymbol{x}_0$ and $\boldsymbol{v}_0 = \boldsymbol{x}_0 - \frac{m}{n}\nabla f(\boldsymbol{x}_0)$. If the following conditions hold: $m \le p$, $q/p \le \mu$, $0 \le nps \le m\sqrt{s} \le 1/L$, $0 \le q\sqrt{s} < 1$, $p > 0$, and $n = q$,*

Table 4.1: Various ODEs and their convergence rates from the literature are recovered from (GM2-ODE) and Theorem 4 in [Maskan et al., 2025c].

ODE	Parameters (GM2-ODE)	Convergence Rate ([Maskan et al., 2025c])
Gradient Flow	$n = 0, m = 1,$ any p, q	$\mathcal{O}(e^{-2\mu t})$
(Polyak ODE)	$q = n = \sqrt{\mu}$ $m = 0, p = \frac{1}{\sqrt{\mu}}$	$\mathcal{O}(e^{-\sqrt{\mu}t})$
(HB-ODE)	$n = q = \sqrt{\mu},$ $m = 0, p = \frac{1}{\sqrt{\mu}} + \sqrt{s}$	$\mathcal{O}(e^{-\sqrt{\mu}t})$
(NAG-ODE)	$n = q = \sqrt{\mu},$ $m = \sqrt{s}, p = \frac{1}{\sqrt{\mu}}$	$\mathcal{O}(e^{-\sqrt{\mu}t})$
Continuous H-NAG ($\gamma = \mu$) [Chen and Luo, 2019]	$n = 1, q = \frac{\mu}{\gamma}$ $p = \frac{1}{\gamma}, m = \beta$	$\mathcal{O}(e^{-t})$

then the Lyapunov function

$$\varepsilon(k) = f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star) + \frac{n}{2p}\|\boldsymbol{v}_k - \boldsymbol{x}^\star\|_2^2 - \frac{nps}{2}\|\nabla f(\boldsymbol{x}_k)\|^2, \tag{4.20}$$

decreases as $\varepsilon(k) \leq (1 - q\sqrt{s})^k \varepsilon(0)$. *Furthermore, we have*

$$f(\boldsymbol{x}_K) - f(\boldsymbol{x}^\star) \leq C''_{GM}(1 - q\sqrt{s})^K,$$

where C''_{GM} *is a positive constant.*

Proof. See [Maskan et al., 2024]. □

Remark 4.2. (GM-ODE) was also discretized using SIE in [Zhang et al., 2021]. After recovering accelerated methods such as NAG, HB, and QHM, they analyzed the convergence of these methods. However, their rates required that the condition cumber $\kappa := \frac{L}{\mu} \geq 9$ and did not match the previously known rates for NAG and HB. Moreover, they used slightly different coefficients than their continuous-time analysis to recover these methods. In other words, the SIE discretization of (GM-ODE) was not coefficient consistent. Further details and a comparison of convergence rates are provided in [Maskan et al., 2025c].

For specific parameter choices, (GM2) recovers several well-known optimization algorithms, including HB, NAG, the TM method [Van Scoy et al., 2018], and the QHM method [Ma and Yarats, 2019]. Theorem 4.5 establishes convergence rates using Lyapunov function analysis recovering the known rates for NAG, and improve upon the existing results for QHM. A summary of these results is given in Table 4.2. In what follows, we briefly outline our results for the TM method and the QHM algorithm.

TM Method: The TM method can be recovered through (GM2) with the choice of

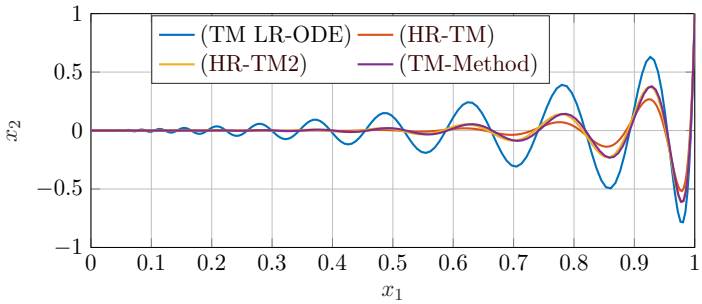

Figure 4.1: Trajectory of TM method and various proposed ODEs for $f(x_1, x_2) = 5 \times 10^{-3}x_1^2 + x_2^2$ with starting point $(x_1(0), x_2(0)) = (1, 1)$. In this figure, (TM LR-ODE) corresponds to the low-resolution ODE of the TM method ((HR-TM2) with $\sqrt{s} = 0$). The step-size was $s = 0.16$.

parameters:

$$m = \frac{1}{\sqrt{L}},\ n = \frac{2\sqrt{\mu L}}{\sqrt{L} - \sqrt{\mu}},\ q = \sqrt{\mu},\ p = \frac{1}{\sqrt{\mu}},\ \text{and } v_k := y_k. \tag{4.21}$$

Inserting these parameters in (GM2-ODE) gives:

$$\ddot{X}_t + \sqrt{\mu}\left(\frac{3 - \sqrt{\frac{\mu}{L}}}{1 - \sqrt{\frac{\mu}{L}}}\right)\dot{X}_t + \sqrt{s}\nabla^2 f(X_t)\dot{X}_t + \left(\frac{2}{1 - \sqrt{\frac{\mu}{L}}} + \sqrt{s\mu}\right)\nabla f(X_t) = 0 \tag{4.22}$$

with $s = 1/L$. To satisfy the parameter conditions in Theorem 4.3, we approximate n in (4.21) as $n \approx 2\sqrt{\mu}$, which leads to the following HR-ODE for the TM method:

$$\ddot{X}_t + 3\sqrt{\mu}\dot{X}_t + \sqrt{s}\nabla^2 f(X_t)\dot{X}_t + (2 + \sqrt{s\mu})\nabla f(X_t) = 0. \tag{HR-TM2}$$

The following corollary of Theorem 4.3 presents the convergence rate of (HR-TM2).

Corollary 4.1. *With the choice of* $\alpha = \log(2\sqrt{\mu})$, $\beta = \sqrt{\mu}t$, $\gamma = 2\sqrt{\mu}t$, $m = \sqrt{s}$, *and* $C = \sqrt{\mu}$, (4.6) *reduces to* (HR-TM2). *Furthermore, invoking Theorem 4.3 with these parameters yields the following convergence guarantee:*

$$f(X_t) - f(\boldsymbol{x}^\star) \le C'_{TM} e^{-2\sqrt{\mu}t},$$

where C'_{TM} *is a positive constant.*

In Figure 4.1, the proposed (HR-TM2) has shown to be more accurate in following the TM method's trajectory than the previous corresponding ODEs.

QHM method: This accelerated method is used in machine learning and training neural networks [Ma and Yarats, 2019, Zhang et al., 2021]. A rescaled version of this numerical

Table 4.2: Comparison of the convergence rates proposed in [Maskan et al., 2025c] (this thesis) with existing results.

	NAG-ODE	HR-TM	NAG Algorithm	QHM
[Shi et al., 2019]	$\mathcal{O}\left(e^{-\frac{\sqrt{\mu}t}{4}}\right)$	-	$\mathcal{O}\left((1+\frac{1}{9}\sqrt{\frac{1}{\kappa}})^{-k}\right)$	-
[Zhang et al., 2021]	$\mathcal{O}\left(e^{-\frac{\sqrt{\mu}t}{2}}\right)$	-	$\mathcal{O}\left((1+\frac{1}{30}\sqrt{\frac{1}{\kappa}})^{-k}\right)$	$\mathcal{O}\left((1+\frac{1}{40}\sqrt{\frac{1}{\kappa}})^{-k}\right), \kappa \geq 9$
[Sun et al., 2020]	-	$\mathcal{O}\left(e^{-\frac{\sqrt{M}t}{2}}\right)$	-	-
[Chen and Luo, 2019]	-	-	$\mathcal{O}\left((1+\sqrt{\frac{\sqrt{\kappa}-1}{\kappa\sqrt{\kappa}}})^{-k}\right)$	-
[Maskan et al., 2025c]	$\mathcal{O}\left(e^{-\sqrt{\mu}t}\right)$	$\mathcal{O}\left(e^{-\frac{4\sqrt{M}t}{3}}\right)$	$\mathcal{O}\left((1-\sqrt{\frac{1}{\kappa}})^{k}\right)$	$\mathcal{O}\left((1-\sqrt{\frac{1}{3\kappa}})^{k}\right)$

scheme is as follows:

$$\begin{cases} \boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -s(1-a)\nabla f(\boldsymbol{x}_k) - sa\boldsymbol{g}_{k+1}, \\ \boldsymbol{g}_{k+1} = b\boldsymbol{g}_k + \nabla f(\boldsymbol{x}_k), \end{cases} \tag{QHM}$$

where $0 \leq a, b \leq 1$. The following corollary, derived from Theorem 4.5, provides new convergence guarantees for the QHM method:

Corollary 4.2. *With the choice of $m = (1-a)\sqrt{s}$, $n = q = \sqrt{a\mu}$, $p = \frac{a}{q} + \sqrt{s}$, $q\sqrt{s} \leq \frac{1}{2}$, and $s \leq \frac{4}{3L}$, for some $0 < a \leq \frac{1}{4}$, algorithm (GM2) reduces to (QHM) with $b = \frac{1-q\sqrt{s}}{1+q\sqrt{s}}$. Furthermore, invoking Theorem 4.5 with these parameters yields the following convergence guarantee:*

$$f(\boldsymbol{x}_K) - f(\boldsymbol{x}^\star) \leq C_{QHM}(1-\sqrt{a\mu s})^K, \tag{4.23}$$

where C_{QHM} is a positive constant.

To the best of our knowledge, the previous strongest convergence guarantee for QHM was $\mathcal{O}\left((1+\sqrt{1/\kappa}/40)^{-K}\right)$, established in [Zhang et al., 2021], but this result required the restrictive assumption $\kappa \geq 9$, where κ denotes the condition number. In Corollary 4.2, we both improve this convergence rate and remove the assumption on κ.

4.2 Revisiting Frank-Wolfe for Structured Nonconvex Optimization

Recall the DC-structured nonconvex problem (1.2):

$$\min_{\boldsymbol{x}\in\mathcal{D}} \ \phi(\boldsymbol{x}) := f(\boldsymbol{x}) - g(\boldsymbol{x}),$$

where the constraint set $\mathcal{D}$ is a convex compact set with diameter D as described above. Further, assume $f(\boldsymbol{x})$ to be L-smooth convex and $g(\boldsymbol{x})$ to be convex, but not necessarily smooth. As explained in Chapter 3, DCA is a suitable method to solve this problem.

Then, the iterates of the DCA would be:

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{D}} \ \hat{\phi}_t(\boldsymbol{x}) \coloneqq f(\boldsymbol{x}) - g(\boldsymbol{x}_t) - \langle \boldsymbol{u}_t, \boldsymbol{x} - \boldsymbol{x}_t\rangle, \tag{4.24}$$

where $\boldsymbol{u}_t \in \partial g(\boldsymbol{x}_t)$ is an arbitrary subgradient of g at $\boldsymbol{x}_t$. Problem (4.24) is a smooth convex optimization problem, making it a natural fit for the FW algorithm, which attains a convergence rate of $\mathcal{O}(1/K)$ as shown in Lemma 3.1. Applying FW to the subproblems of DCA yields the Dc-Fw algorithm (see Algorithm 1). Without loss of generality, we assumed that $g(\mathbf{x})$ is differentiable in Algorithm 1; however, our convergence results for this algorithm remain valid even when $g(\mathbf{x})$ is non-differentiable. To propose our convergence result, we need to define a new gap measure.

Definition 4.1. We measure convergence in terms of the following gap definition:

$$\operatorname{gap}_{\text{DC}}(\boldsymbol{x}_t) := \max_{\boldsymbol{x}\in\mathcal{D}} \min_{\boldsymbol{u}\in\partial g(\boldsymbol{x}_t)} \Big\{ f(\boldsymbol{x}_t) - f(\boldsymbol{x}) - \langle \boldsymbol{u}, \boldsymbol{x}_t - \boldsymbol{x}\rangle \Big\}.$$

When g is differentiable, the subdifferential becomes a singleton, $\partial g(\boldsymbol{x}_t) = \{\nabla g(\boldsymbol{x}_t)\}$, and the gap measure simplifies to

$$\operatorname{gap}_{\text{DC}}(\boldsymbol{x}_t) = \max_{\boldsymbol{x}\in\mathcal{D}} \{ f(\boldsymbol{x}_t) - f(\boldsymbol{x}) - \langle \nabla g(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}\rangle \}$$

.

Lemma 4.1. *The measure* $\operatorname{gap}_{DC}(\boldsymbol{x}_t)$ *is nonnegative for any* $\boldsymbol{x}_t \in \mathcal{D}$*, and it is equal to zero if and only if* $\boldsymbol{x}_t$ *is a critical point satisfying*

$$(\nabla f(\boldsymbol{x}_t) + \mathcal{N}_{\mathcal{D}}(\boldsymbol{x}_t)) \cap \partial g(\boldsymbol{x}_t) \neq \emptyset, \tag{4.25}$$

where $\mathcal{N}_D(\boldsymbol{x}_t)$ *is the normal cone of* $\mathcal{D}$ *at* $\boldsymbol{x}_t$*.*

Moreover, if g *is differentiable (but not necessarily smooth, i.e., its gradients may not be Lipschitz continuous), then the condition* (4.25) *reduces to the characterization of a first-order stationary point. In other words,* $\operatorname{gap}_{DC}(\boldsymbol{x}_t) = 0$ *if and only if* $\boldsymbol{x}_t$ *is a first-order stationary point of problem* (1.2)*.*

Proof. See [Maskan et al., 2025b]. □

Next, we establish the convergence result for Algorithm 1 under the assumption that the DCA subproblems are solved only approximately, up to an ϵ-accurate solution. The theorem guarantees that, even with inexact subproblem solutions, the overall method converges to a stationary point—and to a critical point when g is non-differentiable.

Theorem 4.6. *Suppose that the sequence* $\boldsymbol{x}_t$ *is generated by an inexact-DCA algorithm designed to solve the subproblems described in* (4.24) *approximately, satisfying the inequality* $\hat{\phi}_t(\boldsymbol{x}_{t+1}) - \min_{x\in\mathcal{D}} \hat{\phi}_t(\boldsymbol{x}) \leq \epsilon/2$ *for some* $\epsilon > 0$*. Then, if* $\boldsymbol{x}^\star$ *is a critical/stationary*

Algorithm 1 DC-FW

Input: initial point $\boldsymbol{x}_1 \in \mathcal{D}$, target accuracy $\epsilon > 0$
for $t = 1, 2, \ldots$ **do**
  Initialize $X_{t,1} = \boldsymbol{x}_t$
  for $k = 1, 2, \ldots$ **do**
    $S_{t,k} = \arg\min_{\boldsymbol{x} \in \mathcal{D}} \langle \nabla f(X_{t,k}) - \nabla g(\boldsymbol{x}_t), \boldsymbol{x} \rangle$
    $D_{t,k} = S_{t,k} - X_{t,k}$
    if $-\langle \nabla f(X_{t,k}) - \nabla g(x_t), D_{t,k} \rangle \leq \epsilon/2$ **then**
      set $\boldsymbol{x}_{t+1} = X_{t,k}$ and break
    end if
    $X_{t,k+1} = X_{t,k} + \eta_{t,k} D_{t,k}$ // use $\eta_{t,k} = 2/(k+1)$ or the greedy step-size in (3.3)
  end for
end for

point of (1.2)*, the following bound holds:*

$$\min_{0 \leq \tau \leq t} \operatorname{gap}_{DC}(\boldsymbol{x}_\tau) \leq \frac{\phi(\boldsymbol{x}_0) - \phi(\boldsymbol{x}^\star)}{t+1} + \frac{\epsilon}{2}. \tag{4.26}$$

Proof. See [Maskan et al., 2025b]. □

This result, together with Lemma 3.1, gives the final convergence rate of the DC-FW method.

Corollary 4.3. *DC-FW generates a sequence of solutions that satisfies*

$$\min_{0 \leq \tau \leq t} \operatorname{gap}_{DC}(\boldsymbol{x}_\tau) \leq \epsilon$$

within $\mathcal{O}(1/\epsilon)$ *iterations, requiring at most* $\mathcal{O}(1/\epsilon^2)$ *calls to the linear minimization oracle.*

The choice of DC decomposition in Definition 4.1 determines the notion of stationarity captured by the gap function, thereby enabling a trade-off between computational efficiency and the strength of the resulting stationarity measure. This follows from the fact that $\operatorname{gap}_{\text{DC}}$ preserves the convex component f while linearizing the concave term $-g$, effectively discarding higher-order information in g.

To illustrate this effect, consider $\phi(x) = \sin(\pi x_1)\cos(\pi x_2)$ as a smooth nonconvex function on the domain $[-1, 1]^2$ with $L = \pi^2$. We compute $\operatorname{gap}^L_{\text{PGM}}$, corresponding to the decomposition:

$$f(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2, \;\; g(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2 - \phi(\boldsymbol{x}),$$

and $\operatorname{gap}^L_{\text{PPM}}$, corresponding to the decomposition

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x}) + {}^L\!/_2\|\boldsymbol{x}\|_2^2, \;\; g(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2,$$

over a fine grid across this domain. As shown in Figure 4.2, $\operatorname{gap}^L_{\text{PGM}}$ behaves similarly around all stationary points—local minima, saddle points, and local maxima—whereas

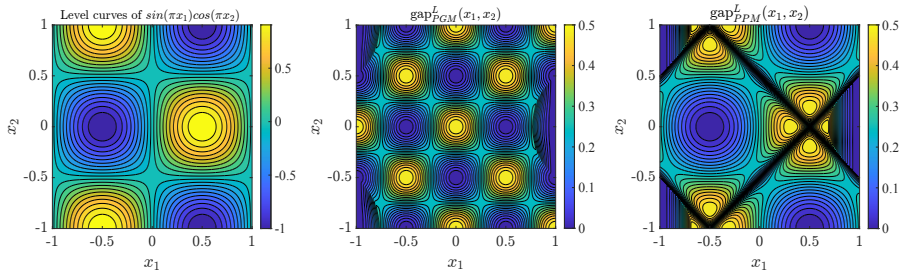

Figure 4.2: Comparison of the DC gap function for different decompositions of $\phi(x_1, x_2) = \sin(\pi x_1)\cos(\pi x_2)$ on the domain $[-1, 1]^2$. *[Left]* Level curves of ϕ. *[Middle]* $\text{gap}^L_{\text{PGM}}$, corresponding to the decomposition $f(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2$, $g(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2 - \phi(\boldsymbol{x})$, which linearizes ϕ and therefore does not distinguish between local minima, saddle points, and local maxima. *[Right]* $\text{gap}^L_{\text{PPM}}$, corresponding to the decomposition $f(\boldsymbol{x}) = \phi(\boldsymbol{x}) + {}^L\!/_2\|\boldsymbol{x}\|_2^2$, $g(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2$, which retains curvature information in ϕ; it is flatter around local minima and sharper around saddle points and local maxima.

$\text{gap}^L_{\text{PPM}}$ clearly differentiates between them: it is flatter near the local minimum and noticeably sharper near saddle points and local maxima. This distinction arises because $\text{gap}^L_{\text{PPM}}$ retains curvature information through f, while $\text{gap}^L_{\text{PGM}}$ removes it by linearizing the objective.

Our analysis has several other important results. Here, we briefly mention some of them:

Recovering CGS: Since the DC decomposition $\phi(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x})$ is not unique, we examined two natural DC decompositions within the standard smooth minimization template. One decomposition was $f(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2$, $g(\boldsymbol{x}) = {}^L\!/_2\|\boldsymbol{x}\|_2^2 - \phi(\boldsymbol{x})$, for an L-smooth nonconvex function $\phi(\boldsymbol{x})$. Using this decomposition in Algorithm 1 leads to an algorithm that reduces gradient computations, improving the gradient complexity from $\mathcal{O}(\epsilon^{-2})$ to $\mathcal{O}(\epsilon^{-1})$. This algorithm found the CGS method (see Section 3.2) as a special case and extend the result from [Lan and Zhou, 2016] to the nonconvex functions.

Improved Convergence: When the problem's constraint set is strongly convex, Dc-Fw improves the linear minimization oracle complexity to $\mathcal{O}(\epsilon^{-3/2})$.

Dc-Fw vs FW: Our results suggest that the DC structure did not limit the FW algorithm. On the contrary, Dc-Fw achieved similar convergence rate to the FW method. Though the similar convergence rate, we demonstrated the effectiveness of Dc-Fw in comparison to the standard FW approach through numerical experiments on quadratic assignment problem and alignment of partially observed embeddings (see [Maskan et al., 2025b, Section 6]).

4.3 Randomized Block Coordinate DC Programming

In Section 3.3, we mentioned expensive DCA subproblems when dealing with high-dimensional problems. Motivated by this issue, we study a block-coordinate DC algorithm. In addition, we investigate the convergence of the proposed method and its connections to a

Algorithm 2 Block Coordinate DC Algorithm (Bdca)

Input: Starting point $\boldsymbol{x}^1 \subseteq \mathcal{D}$, and total number of itreations K.
for $k = 1$ **to** K **do**
  Select a subgradient $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$
  Choose i_k from $\{1, \ldots, N_b\}$ uniformly at random
  Find $x_{i_k}^{k+1} \in \operatorname{argmin}_{x_{i_k} \in \mathcal{D}_{i_k}} f(\boldsymbol{x}_{i_k} + \bar{\boldsymbol{x}}_{i_k}^k) + g_{i_k}(x_{i_k}) - \langle v_{i_k}^k, x_{i_k} \rangle$
  Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}_{i_k}^{k+1} + \bar{\boldsymbol{x}}_{i_k}^k$
end for
Output: $\boldsymbol{x}^{k+1}$

block-coordinate EM algorithm.

Recall the optimization problem

$$\min_{\boldsymbol{x} \in \mathcal{D}} \quad \phi(\boldsymbol{x}) := f(\boldsymbol{x}) + h(\boldsymbol{x}) - g(\boldsymbol{x}),$$

where f, h, and g are lower semi-continuous convex functions defined on the closed convex set $\mathcal{D} \subseteq \mathbb{R}^n$. We do not impose the smoothness assumption on the terms g and h, yet we assume that the function f is smooth, and that the function h and the set $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \ldots \times \mathcal{D}_n$ exhibit a block-separable structure.

We previously introduced the DCA subproblems in (4.24), which involve iterative minimization of a surrogate function. By restricting the minimization of this surrogate to specific coordinate directions, we obtain a block-coordinate variant of DCA. We determine these specific directions by randomly choosing coordinate blocks. The surrogate objective restricted to the i^{th} coordinate block is:

$$\begin{aligned}\hat{\phi}_i(x_i, \boldsymbol{x}^k) :=& \hat{\phi}(D_i^\top x_i + \bar{\boldsymbol{x}}_i^k, \boldsymbol{x}^k) = \hat{\phi}(\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i^k, \boldsymbol{x}^k) \\ =& \underline{f(\boldsymbol{x}_i + \bar{\boldsymbol{x}}_i^k) + g_i(x_i)} - g_i(x_i^k) + g(\boldsymbol{x}^k) - h(\boldsymbol{x}^k) - \langle \underline{\nabla_i h(\boldsymbol{x}^k), x_i} - x_i^k \rangle.\end{aligned}$$

Here, only the underlined terms are relevant for solving the subproblem, as the remaining terms are constant with respect to x_i. Therefore, we propose Block Coordinate DC Algorithm (Bdca), as outlined in Algorithm 2.

Convergence of Bdca: To study the convergence of Algorithm 2, we define our stationarity measure in Definition 4.2.

Definition 4.2. We use the following 'gap' function as a measure of closeness to a first order stationary point:

$$\operatorname{gap}_{\mathcal{D}}^L(\boldsymbol{y}) = \max_{\boldsymbol{x} \in \mathcal{D}} \left\{ \langle \nabla f(\boldsymbol{y}) - \nabla h(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x} \rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \right\}.$$

The next lemma shows that this is a suitable measure for first-order stationarity.

Lemma 4.2. *Let $\mathcal{D} \subseteq \mathbb{R}^m$ be a closed convex set, and let $f, g, h : \mathcal{D} \to \mathbb{R}$ be lower-*

Algorithm 3 Block EM Algorithm

Input: total iterations K, initialize the probability distribution P using vector $\boldsymbol{\theta}^{(0)}$ (P belongs to the exponential family), and number of blocks N_b
for $k = 1$ **to** K **do**
  Randomly choose i_k in $\{1, ..., N_b\}$ with uniform distribution
  Update $\hat{P}^{k+1}(\boldsymbol{y}) = \frac{P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^k)}{\sum_{\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^k)}$
  Update $\theta_{i_k}^{k+1} = \arg\min_{\theta_{i_k}} - \sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}} \hat{P}^{k+1}(\boldsymbol{y}) \log P(\boldsymbol{x},\boldsymbol{y}|\theta_1^k, \ldots, \theta_{i_k}, \ldots, \theta_n^k)$
  Set $\boldsymbol{\theta}^{k+1} = \bar{\boldsymbol{\theta}}_{i_k}^k + \boldsymbol{\theta}_{i_k}^{k+1}$
end for
Output: $\boldsymbol{\theta}^{K+1}$

semicontinuous convex functions, where f is L-smooth and h is continuously differentiable. Then, $\mathrm{gap}_{\mathcal{D}}^L(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y} \in \mathcal{D}$, and $\mathrm{gap}_{\mathcal{D}}^L(\boldsymbol{y}) = 0$ if and only if $\boldsymbol{y}$ is a first-order stationary point of problem (3.7).

Proof. See [Maskan et al., 2025a]. □

We can now present the main convergence result.

Theorem 4.7. *Suppose $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K$ is a sequence generated by BDCA for solving problem (3.7). Then, the following bound holds:*

$$\min_{k\in\{1,\ldots,K\}} \mathbb{E}\left[\mathrm{gap}_{\mathcal{D}}^L(\boldsymbol{x}^k)\right] \leq \frac{N}{K}\left(\phi(\boldsymbol{x}^1) - \phi^\star\right).$$

Proof. See [Maskan et al., 2025a]. □

When h is non-differentiable, we replace $\nabla h(\boldsymbol{y})$ in the definition of the gap by a subgradient $\boldsymbol{v} \in \partial h(\boldsymbol{y})$, and define

$$\mathrm{gap}_{\mathcal{D}}^L(\boldsymbol{y}) = \max_{\boldsymbol{x}\in\mathcal{D}} \min_{\boldsymbol{v}\in\partial h(\boldsymbol{y})} \{\langle \nabla f(\boldsymbol{y}) - \boldsymbol{v}, \boldsymbol{y} - \boldsymbol{x}\rangle + g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\}.$$

This choice ensures the gap remains well-defined, as it is sufficient that the condition holds for one subgradient of h at $\boldsymbol{y}$. When h is differentiable, this definition reduces to Definition 4.2.

The result below extends Lemma 4.2 to the setting where h is non-differentiable; the proof follows similar arguments and can be found in [Maskan et al., 2025a].

Lemma 4.3. *Let $\mathcal{M} \subseteq \mathbb{R}^m$ be a closed convex set, and let $f, g, h : \mathcal{M} \to \mathbb{R}$ be lower-semicontinuous convex functions, where f is L-smooth. Then $\mathrm{gap}_{\mathcal{M}}^L(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y} \in \mathcal{M}$, and $\mathrm{gap}_{\mathcal{M}}^L(\boldsymbol{y}) = 0$ if and only if $\boldsymbol{y}$ is a critical point of problem (3.7), defined by the condition*

$$(\nabla f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) + \mathcal{N}_{\mathcal{M}}(\boldsymbol{y})) \cap \partial h(\boldsymbol{y}) \neq \varnothing. \tag{4.27}$$

Theorem 4.7 continues to hold in the non-differentiable setting. The proof follows identically, except that $\nabla_{i_k} h(\boldsymbol{x}^k)$ replaced by the i_k-th component (or block) of a subgradient $\boldsymbol{v}^k \in \partial h(\boldsymbol{x}^k)$, denoted by $v^k_{i_k}$. Any subgradient can be used, since the gap is defined via a minimization over the subdifferential set. The proof proceeds without further modification.

Block EM: Next, we introduce the Block EM method through the connection of DCA and the EM method discussed in Section 3.3. Notably, we can use the guarantees established in Theorem 4.7 to prove non-asymptotic expected convergence rates for Block EM.

Corollary 4.4. *Suppose there exists a sequence $\boldsymbol{\theta}^k, k = 1 \ldots, K$ generated by Block EM algorithm. Then,*

$$\min_{k \in \{1,\ldots,K\}} \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{\theta}^k)\|^2\right] \leq \frac{2N_b L}{K}(\mathcal{L}(\boldsymbol{\theta}^1) - \mathcal{L}(\boldsymbol{\theta}^\star)).$$

Remark 4.3. The results in Corollary 4.4 are consistent with the $\mathcal{O}(1/K)$ convergence rate guarantee for the standard EM algorithm, as provided by Kumar and Schmidt [2017].

Further discussion and numerical performance of BDCA on logistic regression with nonconvex sparsity regularization, nonconvex quadratic programming with negative ℓ_1 regularization, and one-bit multiple-input-multiple-output channel signal recovery problem are presented in [Maskan et al., 2025a, Section 5, Appendix F].

Chapter 5

Conclusion & Future Directions

In this thesis, we studied optimization algorithms for solving structured optimization problems under specific assumptions and geometric properties.

We began by studying first-order accelerated methods for smooth convex and smooth strongly convex objectives. In **paper I**, we introduced a unified HR-ODE framework for analyzing momentum-based accelerated gradient algorithms. Using the Forced Euler–Lagrange equation, we derived a general HR-ODE that captures the continuous-time behavior of a broad class of classical momentum methods. Within this framework, we were able to recover known HR-ODEs and prove sharper convergence guarantees. We further analyzed the convergence rates of both the proposed HR-ODEs and their discretizations using general Lyapunov-function arguments.

In the strongly convex regime (**paper II**), our framework uncovers a systematic relationship between the proposed general HR-ODE (GM2-ODE) and its SIE discretization. The resulting general method exactly recovers several prominent accelerated algorithms, including HB, NAG, TM, and QHM. Among existing unified formulations, we demonstrated that (GM2-ODE) yields the fastest known convergence rate for each of these recovered methods. In the convex setting, our framework further provides improved gradient-norm minimization rates for NAG. Additionally, our unified perspective subsumes the HR-ODE of the algorithm proposed by [Wibisono et al., 2016].

In the second part of the thesis, we considered nonconvex constrained optimization tasks, focusing on settings where either the Frank–Wolfe method or a coordinate-type algorithm become advantageous.

We studied the Dc-Fw framework, which couples the DCA with the FW algorithm. We showed that Dc-Fw retains the same oracle complexity for linear minimization steps as the standard FW method in nonconvex optimization. At the same time, by choosing the DC decomposition appropriately, Dc-Fw can adapt to the geometry of the underlying problem, providing a flexible approach suitable for a wide range of settings. Within the standard smooth minimization template, we analyzed two natural DC decomposi-

tions. One of them leads to an algorithm that reduces gradient evaluations, improving the gradient complexity from $\mathcal{O}(\epsilon^{-2})$ to $\mathcal{O}(\epsilon^{-1})$. Moreover, when the feasible domain is strongly convex, this decomposition improves the linear minimization oracle complexity to $\mathcal{O}(\epsilon^{-3/2})$. Through numerical experiments on the quadratic assignment problem and the alignment of partially observed embeddings, we demonstrated that DC-FW substantially outperforms the standard FW algorithm (**paper III**).

When the problem is high-dimensional, classical DCA subproblems may become computationally intractable. In **paper IV**, we examined a broad class of DC problems consisting of smooth components together with separable nonsmooth terms, and proposed a randomized block-coordinate DC algorithm (BDCA) tailored for such settings. We established convergence guarantees for BDCA at a rate of $\mathcal{O}(n/K)$, where the method performs K iterations over n coordinate blocks. Leveraging the relationship between DCA and the EM algorithm, we further introduced a randomized block-coordinate EM method (Block EM), whose convergence guarantees follow directly from those of the DCA as a special case. Given the widespread use of DCA and EM in data science and machine learning, scalable variants such as BDCA and Block EM offer substantial potential for practical impact.

5.1 Future Directions

Several directions for future research naturally emerge from the work presented in this thesis. First, our use of the Forced Euler–Lagrange equation has thus far been limited to Euclidean settings. Extending this framework to non-Euclidean geometries would be a natural and important next step, with the potential to broaden its applicability to manifold-constrained optimization problems.

In the strongly convex regime, while the proposed general model (GM2) recovers the TM method, the discrete-time analysis developed here does not yet match the state-of-the-art convergence rate known for TM [Van Scoy et al., 2018]. Faster rates can be obtained through the approximation-based formulation (HR-TM2), but it remains an open question whether comparable guarantees can be established directly for the exact model (4.22) without relying on approximation.

The second part of the thesis illustrates that a simple but deliberate choice of DC decomposition can substantially improve algorithmic efficiency. This observation suggests a broader research direction: investigating how alternative or systematically chosen DC reformulations may enhance performance across different problem classes. The fact that such improvements arise from straightforward reformulations indicates that the potential of this approach is far from exhausted.

Our work on randomized coordinate DC methods also opens several promising avenues. Extending the analysis of BDCA to stochastic environments which are closer to contemporary machine learning practices, offers both theoretical and practical value, especially for large-scale problems where randomness is inherent.

Finally, the design of adaptive DC decompositions that adjust dynamically to the geometry of the optimization landscape presents another compelling direction. Such adaptivity could yield algorithms that are more robust and efficient across a broader range of applications, and may represent a fruitful step toward more automated structure-exploiting optimization methods.

Bibliography

Hadi Abbaszadehpeivasti, Etienne de Klerk, and Moslem Zamani. On the rate of convergence of the difference-of-convex algorithm (DCA). *Journal of Optimization Theory and Applications*, pages 1–22, 2023.

F. Alvarez, H. Attouch, J. Bolte, and P. Redont. A Second-Order Gradient-Like Dissipative Dynamical System with Hessian-Driven Damping.: Application to Optimization and Mechanics. *Journal de Mathématiques Pures et Appliquées*, 81(8):747–779, 2002. ISSN 0021-7824.

Hedy Attouch, Zaki Chbani, Jalal Fadili, and Hassan Riahi. First-Order Optimization Algorithms via Inertial Systems with Hessian Driven Damping. *Mathematical Programming*, pages 1–43, 2020.

Hédy Attouch, Zaki Chbani, Jalal Fadili, and Hassan Riahi. Convergence of Iterates for First-Order Optimization Algorithms with Inertia and Hessian Driven Damping. *Optimization*, pages 1–40, 2021.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1): 459–494, 2014.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Xufeng Cai, Chaobing Song, Stephen Wright, and Jelena Diakonikolas. Cyclic block coordinate descent with variance reduction for composite nonconvex optimization. In *International Conference on Machine Learning*, pages 3469–3494. PMLR, 2023.

Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

Long Chen and Hao Luo. First Order Optimization Methods based on Hessian-Driven Nesterov Accelerated Gradient Flow. *arXiv preprint arXiv:1912.09276*, 2019.

Cyrille W Combettes and Sebastian Pokutta. Complexity of linear minimization and projection on some sets. *Operations Research Letters*, 49(4):565–571, 2021.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

Vladimir Fedorovich Dem'yanov and Aleksandr Moiseevich Rubinov. *Approximate methods in optimization problems*. American Elsevier Publishing Company, 1970.

Qi Deng and Chenghao Lan. Efficiency of coordinate descent methods for structured nonconvex optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 74–89. Springer, 2020.

Yoel Drori and Marc Teboulle. Performance of First-Order Methods for Smooth Convex Minimization: A Novel Approach. *Mathematical Programming*, 145(1):451–482, 2014.

Pavel Dvurechensky, Shimrit Shtern, and Mathias Staudigl. First-order methods for convex optimization. *EURO Journal on Computational Optimization*, 9:100015, 2021.

Alexandre d'Aspremont, Damien Scieur, and Adrien Taylor. Acceleration Methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021. ISSN 2167-3888.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

Mahyar Fazlyab, Alejandro Ribeiro, Manfred Morari, and Victor M. Preciado. Analysis of Optimization Algorithms via Integral Quadratic Constraints: Nonstrongly Convex Problems. *SIAM Journal on Optimization*, 28(3):2654–2689, 2018.

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

Herbert Goldstein, Charles P. Poole, and John L. Safko. *Classical Mechanics*. Addison-Wesley, San Francisco, 3 edition, 2002.

Hoomaan Hezave, Milad Javadzadeh, and Mohammad Hossein Kahaei. Sparse signal reconstruction using blind super-resolution with arbitrary sampling. *IEEE Signal Processing Letters*, 27:615–619, 2020.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

Thomas Kerdreux. *Accelerating conditional gradient methods*. PhD thesis, Université Paris sciences et lettres, 2020.

Raunak Kumar and Mark Schmidt. Convergence rate of expectation-maximization. In *10th NIPS Workshop on Optimization for Machine Learning*, page 98, 2017.

Deepa Kundur and Dimitrios Hatzinakos. Blind image deconvolution. *IEEE signal processing magazine*, 13(3):43–64, 2002.

Frederik Kunstner, Raunak Kumar, and Mark W. Schmidt. Homeomorphic-invariance of EM: Non-asymptotic convergence in KL divergence for exponential families via mirror descent. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Simon Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

Hoai An Le Thi and Tao Pham Dinh. DC programming and DCA: Thirty years of developments. *Mathematical Programming*, 169(1):5–68, 2018.

Hoai An Le Thi, Van Ngai Huynh, and Tao Pham Dinh. Convergence analysis of difference-of-convex algorithm with subanalytic data. *Journal of Optimization Theory and Applications*, 179(1):103–126, 2018.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints. *SIAM Journal on Optimization*, 26 (1):57–95, 2016.

Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Jerry Ma and Denis Yarats. Quasi-Hyperbolic Momentum and Adam for Deep Learning. In *International Conference on Learning Representations*, 2019.

Hoomaan Maskan, Konstantinos Zygalakis, and Alp Yurtsever. A Variational Perspective on High-Resolution ODEs. *Advances in Neural Information Processing Systems*, 36, 2024.

Hoomaan Maskan, Paniz Halvachi, Suvrit Sra, and Alp Yurtsever. Randomized block coordinate dc algorithm. *EURO Journal on Computational Optimization*, page 100123, 2025a.

Hoomaan Maskan, Yikun Hou, Suvrit Sra, and Alp Yurtsever. Revisiting frank-wolfe for structured nonconvex optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Hoomaan Maskan, Konstantinos C Zygalakis, Armin Eftekhari, and Alp Yurtsever. A unified model for high-resolution odes: New insights on accelerated methods. *arXiv preprint arXiv:2503.15136*, 2025c.

Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

Michael Muehlebach and Michael Jordan. A Dynamical Systems Perspective on Nesterov Acceleration. In *International Conference on Machine Learning*, pages 4656–4662. PMLR, 2019.

Y. Nesterov. A Method for Solving the Convex Programming Problem with Convergence Rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

Yu Nesterov. Smooth Minimization of Non-smooth Functions. *Mathematical Programming*, 103(1):127–152, 2005.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2003.

Yurii Nesterov and Sebastian U. Stich. Efficiency of accelerated coordinate descent method on structured optimization problems. 2016.

Boris T Polyak. Some Methods of Speeding Up the Convergence of Iteration Methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (1):1–38, 2014.

Bin Shi, Simon S Du, Weijie J Su, and Michael I Jordan. Acceleration via Symplectic Discretization of High-Resolution Differential Equations. *arXiv preprint arXiv:1902.03694*, 2019.

Bin Shi, Simon Shaolei Du, Michael I. Jordan, and Weijie J. Su. Understanding the Acceleration Phenomenon via High-Resolution Differential Equations. *ArXiv*, abs/1810.08907, 2021.

Jonathan W Siegel. Accelerated First-Order Methods: Differential Equations and Lyapunov Functions. *arXiv preprint arXiv:1903.05671*, 2019.

Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

Boya Sun, Jemin George, and Solmaz S. Kia. High-Resolution Modeling of the Fastest First-Order Optimization Method for Strongly Convex Functions. In *59th IEEE Conference on Decision and Control, CDC 2020, Jeju Island, South Korea, December 14-18, 2020*, pages 4237–4242. IEEE, 2020.

Pham Dinh Tao and Le Thi Hoai An. A DC optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505, 1998.

Hoai An Le Thi, Tao Pham Dinh, Hoang Phuc Hau Luu, and Hoai Minh Le. Deterministic and stochastic dca for dc programming. In *Springer Handbook of Engineering Statistics*, pages 675–702. Springer, 2023.

Bryan Van Scoy, Randy A. Freeman, and Kevin M. Lynch. The Fastest Known Globally Convergent First-Order Method for Minimizing Strongly Convex Functions. *IEEE Control Systems Letters*, 2(1):49–54, 2018.

Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A Variational Perspective on Accelerated Methods in Optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016. ISSN 0027-8424.

Ashia C Wilson, Ben Recht, and Michael I Jordan. A Lyapunov Analysis of Accelerated Methods in Optimization. *J. Mach. Learn. Res.*, 22:113–1, 2021.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.

Alp Yurtsever and Suvrit Sra. CCCP is Frank-Wolfe in disguise. *Advances in Neural Information Processing Systems*, 35:35352–35364, 2022.

Peiyuan Zhang, Antonio Orvieto, Hadi Daneshmand, Thomas Hofmann, and Roy S Smith. Revisiting the Role of Euler Numerical Integration on Acceleration and Stability in Convex Optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3979–3987. PMLR, 2021.