Université catholique de Louvain

UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

CORE
CENTER FOR
OPERATIONS RESEARCH
AND ECONOMETRICS

New second-order and tensor methods in Convex Optimization

Nikita Doikov

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en Sciences de l'Ingénieur

Dissertation committee:

Prof. Yurii Nesterov (Université catholique de Louvain, Supervisor)
Prof. Alexandre d'Aspremont (École Normale Supérieure)
Prof. Coralia Cartis (University of Oxford)
Prof. Daniele Catanzaro (Université catholique de Louvain)
Prof. François Glineur (Université catholique de Louvain)
Prof. Roland Keunings (Université catholique de Louvain, Chair)

September, 2021

Abstract

In the recent years, we can see that the interest for new optimization methods keeps growing. The modern problems are usually ill-conditioned and high-dimensional. As a consequence, it is hard to solve them by using only the classical techniques. At the same time, the *first-order* or the *gradient methods* very often suffer from slow convergence, reaching their theoretical limitations.

One of the natural ideas for improving the performance of the numerical algorithms is to use higher derivatives of the objective. The classical second-order optimization scheme is called *Newton's method*. It has very fast local quadratic convergence, provided that the starting point is sufficiently close to the optimum. However, contrary to first-order algorithms, the classical Newton's method with unit step size does not possess any global convergence guarantees in the general case.

The main goal of this thesis is to develop and analyse *second-order* and *high-order* optimization methods for solving composite convex optimization problems, together with the different problem classes, for which we can establish the *global* iteration complexity bounds. We are interested in studying implementable algorithms with explicitly stated convergence rates, aiming to have both theoretical and practical justification of the methods.

Acknowledgements

It has been extreme luck and great honor to be supervised by Yurii Nesterov, who gave me a unique opportunity to observe an excellent example of how a high level research must be performed. He formulated a very interesting research question that I was able to work on with a visible hope to succeed and excited by the challenge. His generous and thorough help at all stages of the work has been both very educational and kind to me. I want to share my huge gratitude to him for these very nice years of my PhD studies!

I am very grateful to the members of my jury, Alexandre d'Aspremont, Coralia Cartis, Daniele Catanzaro, François Glineur, and Roland Keunings, for the important suggestions, advices, and comments on my work, which have resulted in much better presentation of the thesis.

I wish to thank a lot my colleagues from UCLouvain, especially, Anton Rodomanov, Geovani Grapiglia, Mihai Florea, Evgeniya Vorontsova, Masoud Ahookhosh, and Valentin Leplat, who shared with me the moments of my stay in this magnificent place. Many people were responsible for making perfect working conditions at UCLouvain, among others, I would like to thank Catherine Germain, Margaux Hubin, Marie-Christine Joveneau, Nathalie Ponet, Pascale Premereur, and Etienne Huens for their help.

I met so many great people in my life who taught me and who helped me in my efforts to do research. I am very thankful to Konstantin Vorontsov, Dmitry Kropotov, Dmitry Vetrov, Yury Maximov, Alexander Gasnikov, Pavel Dvurechensky, Peter Richtárik, Filip Hanzely, Dmitry Grishchenko, Jérôme Malick, and Sebastian Stich, who played a role in forming my research directions.

This thesis would not be possible to finish without constant and warm support from all of my friends and members of my family, to whom I am delighted to dedicate this work.

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 788368).

Contents

Chapter 1

Introduction

The first study of optimization principles and algorithms was undertaken long before the invention of the computer. Today, we can highlight the work of Cauchy in 1847 on the gradient method [91]. It appears that steepest descent and Armijo-type line search were already known at that time, while the rigorous proofs of convergence were waiting for one hundred years to be discovered.

From the middle of the twentieth century, Optimization was quickly recognized as one of the most important parts of Computational Mathematics and Computer Science. Numerical optimization methods formed the foundations for the information revolution, that keeps changing our lives nowadays. The first-order or the gradient methods were established as a fundamental tool for solving nonlinear optimization problems. Extensive research on their convergence started from the 1950s. Let us refer to the paper of Polyak in 1963 on the gradient methods [129], where the proofs of convergence were presented in a modern form, and the works of Shor [143], who discovered the subgradient method in 1962.

Besides, it became clear very soon that general optimization problems are mainly *unsolvable*. Indeed, the class of *all* optimization problems is so large that an intention to develop a universal method seems too ambitious. Convex Analysis, which had taken its modern shape in 1970 due to the book of Rockafellar [132], gave rise to the field of Convex Optimization. Presumably, convex optimization problems are among the only efficiently solvable continuous problems in Optimization Theory. These ideas were reflected in the classic monograph of Nemirovski and Yudin [106], written in 1979. After this work, it became possible to speak of the *complexity* of solving optimization problems from a particular *problem class*. There appeared to exist unavoidable lower complexity bounds for different problem classes, and the *optimal* methods that achieve the corresponding bounds. The Fast Gradient Method, which is optimal for Smooth Convex Optimization, was discovered by Nesterov in 1983 [107].

Over the past decades, progress in first-order optimization theory has been immense. But despite many great achievements, it remains a major bottleneck for the gradient methods that their rate of convergence is slow due to the fundamental theoretical limitations, that are represented by the lower complexity bounds.

Newton's method is a classical numerical algorithm, which has a reputation for being powerful. The initial versions of the method were considered by Newton in 1669 for solving polynomial equations, and in general form by Raphson in 1690 [131]. Its convergence was studied in the works of Fine and Bennett in 1916 [51, 11], and in the paper of Kantorovich in 1948 [79]. From the optimization perspective, Newton's method is a *second-order* algorithm that is based on the quadratic approximation of the target function. Thus the rate is *locally quadratic*, which is much faster than the rate of the gradient methods. However, when the starting point is far away from the optimum, the convergence of Newton's method can be arbitrarily slow, or even absent.

In this thesis, our goal is to develop efficient second-order and high-order optimization methods that have *global* iteration complexity guarantees. We study several modifications of Newton's method and analyse their convergence rates. Some of the approaches were known from previous works but received extended analysis and important new features. The other algorithms are completely new. For all our methods, we prove the global rates and compare them with the rates of the first-order methods. Numerical experiments are included.

First-Order Optimization. Let us mention some of the most representative directions and works in the area.

In the beginning of this century, it was revealed that we can significantly accelerate first-order methods by moving out of the black-box optimization concept. Very often, we know additional information about the objective, and it might help an algorithm to be more effective by using that knowledge properly. In this vein, the framework of *composite optimization*, which is able to treat simple nondifferentiable components, was developed by Nesterov [114], and by Beck and Teboulle [9] with applications to image processing.

Another major source of information is the primal-dual structure of the problem. The *smoothing technique* (Nesterov, 2005 [108], 2007 [110]; d'Aspremont, El Ghaoui, Jordan, Lanckriet, 2007 [32]; d'Aspremont, 2008 [31]; Devolder, Glineur, Nesterov, 2012 [35]; Kelner, Lee, Orecchia, Sidford, 2014 [81]) is a striking example of using such structure of a given nondifferentiable function in the methods that were initially developed for differentiable objectives. This technique has been widely adopted in many applications, including the principal component analysis (PCA), semidefinite programming (SDP), optimal control, and discrete optimization problems. One more example of utilizing the min-max structure of the problem is the Mirror Prox algorithm, that was proposed by Nemirovski in 2004 [104]. A particular instance of this algorithm known as the *extragradient* method was developed by Korpelevich in 1976 [84].

There was growing interest in the interplay between Optimization and Machine Learning in the 2010s (see the volume [146]). *Stochastic methods* with their complexity bounds were studied in the works of Nemirovski, Juditsky, Lan, and Shapiro [105, 87]. The first-order methods with *inexact oracle information*, and the *universal methods* that can automatically adapt to the smoothness properties of the objective were developed by Devolder, Glineur, and Nesterov in [36, 115], and for the stochastic setting by Dvurechensky and Gasnikov in [47]. Adaptive subgradient methods for online learning and stochastic optimization were introduced by Duchi, Hazan, and Singer in 2011 [46].

A large group of problems in Machine Learning and Statistics can be modelled as a *finite-sum minimization* problem, where the target objective is represented as a (huge) sum of losses evaluated at different objects from a given dataset. A notable achievement was the development of the *variance reduction* technique by Schmidt, Le Roux, and Bach in 2012 [141] for the gradient methods solving such problems.

The modern huge-scale problems needed new ideas and new methods, while some of the other developments were revisitings of the old techniques. Thus the *coordinate descent* methods became very popular for solving the problems with thousands and millions of variables, after the first complexity guarantees were established by Nesterov in 2012 [113]. The accelerated coordinate methods with nonuniform random samplings were proposed by Lee and Sidford, 2013 [90], and with improved sampling distributions in 2016 by Allen-Zhu, Qu, Richtárik, and Yuan [3], and by Nesterov and Stich [125]. The coordinate descent method with volume sampling, that has provably better performance when increasing the batch size, was developed by Rodomanov and Kropotov in 2020 [135].

The *conditional gradient methods* also received substantial attention during the past decade, though the first algorithm of this type was proposed back in 1956 by Frank and Wolfe [52]. It appeared that these methods are very efficient for solving high-dimensional problems over the convex sets with difficult structure (see the works of Jaggi, 2013 [73], Lacoste-Julien, Jaggi, Schmidt, and Pletscher [86], Lan, 2013 [88], Harchaoui, Juditsky, and Nemirovski, 2015 [69]).

One of the recent promising research directions is *computer aided* analysis for the performance of first-order methods (Drori and Teboulle, 2014 [45]; Kim and Fessler, 2016 [82]; Taylor, Hendrickx, and Glineur, 2017 [149, 148]). With the help of computers, there were developed the accelerated gradient methods that match the lower complexity bounds with the best numerical factors.

Second-Order Optimization. From the beginning of using Newton's method in computational practice, there have been many techniques developed to improve its convergence properties.

A popular approach, that is often called the *damped* Newton method, is to perform a line search for the Newton direction. This idea was proposed in 1948 by Kantorovich. A more modern reference is the book of Ortega and Rheinboldt [128], originally published in 1970. For some classes of problems, it is possible to establish the global convergence for the damped Newton iterations. However, there are two serious issues with this approach. First, the method might not work when the Hessian is a degenerate matrix (which happens even to the convex problems). Second, the complexity guarantees of the damped Newton method are usually much *worse* than that of the basic gradient methods. Therefore, from the theoretical perspective, there is no point in using the second-order information in this case, until entering the region of quadratic convergence.

To deal with the degeneracy of the Hessian, one can use the Levenberg-Marquardt algorithm, first published in 1944 [92] and then rediscovered in 1963 [97]. They suggested to regularize the Hessian with the identity matrix multiplied by some positive coefficient. It can be viewed as a strategy for combining the Newton algorithm with the gradient method. So the regularization parameter should mix the best of the performances of these two methods. At the same time, the Levenberg-Marquardt algorithm may suffer from the slow worst-case convergence of the first-order schemes, while the choice of the regularization parameter is not easy.

The *trust-region* approach is a different and very popular technique of globalizing the Newton iterations. The idea is to restrict the quadratic model of the function onto a neighbourhood of the current point. The size of this neighbourhood is a parameter that we need to choose. It should balance the error of the model and the length of the method step. Trust-region methods originated in the work of Goldfeld, Quandt, and Trotter in 1966 [56]; afterwards, they were extensively developed by Conn, Gould, and Toint (see their book [30], published in 2000). An example of slow behaviour of Newton's method and its trust-region modifications was constructed by Cartis, Gould, and Toint in 2013 [25]. It was demonstrated that for unconstrained minimization of a smooth function with globally Lipschitz continuous Hessian, the number of iterations of the Newton algorithm might be as many as of the steepest descent.

A big step in a second-order optimization theory was made after the paper [124] by Nesterov and Polyak in 2006, where *cubic regularization* of Newton's method with its global complexity guarantees was justified. The main idea of [124] is to use a global *upper* approximation model of the objective, which is the second-order Taylor's polynomial augmented by a cubic term. For different problem classes, it was shown that the Cubic Newton algorithm has global rates which are better than those of the gradient methods.

Moreover, one can find elements of all three approaches (a line search, the Levenberg-Marquardt, and the trust-region techniques) in the cubic regularization scheme, but all these features are just consequences of the core idea, which is to employ a global upper approximation. Probably the first appearance of the cubic regularization of Newton's method in the scientific literature was the paper [64] by Griewank in 1981.

The following results provide a good perspective for the development of the cubic regularization approach. *Accelerated* second-order schemes for convex minimization were discovered in (Nesterov, 2008 [111]). *Adaptive* cubic regularization methods were developed in (Cartis, Gould, and Toint, 2011 [21, 22]). The latter algorithms showed encouraging performance, employing both an adaptive estimation of the regularization parameter and efficient approximations of the exact cubic step. Extending the idea of adaptive search, *universal* schemes that can automatically adjust to a second-order smoothness of a particular objective function were proposed in (Grapiglia and Nesterov, 2017 [60, 61]). The methods based on *probabilistic* models with cubic regularization and line search were developed in (Cartis and

Scheinberg, 2018 [28]) for solving large-scale unconstrained minimization problems.

In the same vein, the Gauss-Newton algorithms with global complexity guarantees for solving a system of nonlinear equations were proposed in (Nesterov, 2007 [109]).

The *lower* complexity bounds for the second- and high-order methods were obtained by (Cartis, Gould, and Toint, 2010 [20]; Arjevani, Shamir, and Shiff, 2019 [4]; Agarwal and Hazan, 2018 [1]). The accelerated proximal method that was proposed in (Monteiro and Svaiter, 2013 [101]) turned out to be nearly *optimal* for second-order convex optimization, matching the corresponding lower bound up to logarithmic terms. Such additional payment is required for some heavy auxiliary line search at each iteration.

There are two open theoretical questions related to the cubic regularization technique, which we address in our thesis.

First, it is still not fully understood, what the global complexity bounds of the regularized Newton schemes for the problems with *strongly* convex and *uniformly* convex objectives are. For the first-order algorithms, strongly convex functions with Lipschitz continuous gradient serve as an example of *nondegenerate* problem class, that is the most favourable to the methods. Therefore, a comparison between the first-order and the second-order schemes on these problems is of a high importance.

Second, a nice property of the classical Newton's method is *affine-invariance*. It makes the method independent of the coordinate system, which can be chosen in the wrong way in applications. On the contrary, in the Cubic Newton method we are obliged to fix the norm for the regularizer. As a consequence, the method is no longer affine-invariant and quite sensitive to the choice of the coordinate system.

Affine-invariant characterization of Newton's method is mainly related to the framework of *self-concordant* functions, introduced for the study of the *interior-point* methods by Nesterov and Nemirovski in 1994 [123]. From the global perspective, this class provides us with an upper second-order approximation of the objective, which naturally leads to the damped Newton iterations. Several new results are related to the analysis of the damped Newton method for *generalized self-concordant* functions (Bach, 2010 [5]; Sun and Tran-Dinh, 2019 [147]), and the notion of *Hessian stability* (Karimireddy, Stich, and Jaggi, 2018 [80]). However, for more refined problem classes, we can often obtain much better complexity estimates by using the cubic regularization technique (see Dvurechensky and Nesterov, 2018 [49]).

In this thesis, we propose a new family of second-order algorithms called

Contracting Newton methods that have both the affine-invariance property and the fast global rate of the Cubic Newton method.

Tensor Methods. It seems to be a natural idea to increase the efficiency of the methods by employing high-order oracles. The study of high-order numerical methods for solving nonlinear equations is dated back to the work of Chebyshev in 1838, where the scalar methods of order three and four were proposed [29]. The methods of arbitrary order for solving nonlinear equations were studied by Evtushenko and Tretyakov in 2014 [50].

The main obstacle to using this approach in Optimization consists in a prohibiting complexity of the corresponding Taylor's approximations formed by the high-order multidimensional polynomials, which are difficult to store, handle, and minimize. If we go just one step above the commonly used quadratic approximation, we get a multidimensional polynomial of degree three which is never convex. Consequently, its usefulness for optimization methods was questionable.

However, recently in the work of Nesterov, 2019 [118], it was shown that Taylor's polynomials of *convex functions* have a very interesting structure. It appeared that their augmentation by a power of Euclidean norm with a reasonably big coefficients gives us a global upper *convex* model of the objective function, which keeps all advantages of the local high-order approximation.

Hence, it became possible to speak about efficient implementation of the tensor methods, while their rate of convergence in terms of the iterations is dramatically fast. The global complexity bounds of the basic and accelerated tensor methods were studied by (Baes, 2009 [6]; Nesterov, 2019 [118]; Gasnikov et al., 2019 [54]). Universal tensor methods, which can automatically adapt to the Hölder parameters of the objective, were developed by Grapiglia and Nesterov in 2019 [63]. Optimal combinations of the tensor methods for minimization problems with a sum of functions were studied by Kamzolov, Gasnikov, and Dvurechensky in 2020 [77]. Adaptive high-order methods for nonconvex optimization, together with sharp worst-case complexity bounds were investigated by Cartis, Gould, and Toint in 2020 [27].

Application of high-order methods for optimization of a smooth approximation of nonsmooth functions was considered by Bullins, 2020 [16].

In this thesis, our focus on the tensor methods is twofold. Firstly, it is of theoretical interest and curiosity to study the methods in its general form. We believe that understanding the core principles behind the methods of different order may lead us to new developments in the second-order and even in the first-order optimization algorithms. Secondly and more importantly, recently (Nesterov, 2020 [122]) we received a confirmation that the third-order schemes can be efficiently implemented by employing *only* the *second-order* information. Therefore, it is not possible to avoid the tensor methods, when speaking on the second-order optimization.

Structure of the Thesis. The rest of this chapter is organized as follows. Section 1.1 contains an *overview of our contributions.* Then we introduce the latest research direction on high-order methods in Smooth Convex Optimization. We list some known algorithms of different order and the corresponding convergence theory. During description of the methods, we highlight issues with them which motivate our developments presented in the follow-up parts. Preliminaries and our notation are in Section 1.3.1. In Section 1.3.2, we review the Gradient Method. Sections 1.4 and 1.5 are devoted to second- and high-order methods, respectively. In Section 1.6 we discuss arithmetical complexity for the oracles of different order.

The main results of the thesis are presented within the following chapters.

Chapter 2 is devoted to *uniformly convex* functions. In Section 2.1 we study the global performance of a regularized Newton method for the uniformly convex problems, and in Section 2.2, the local convergence of high-order Tensor Methods.

Chapter 3 presents our results related to a *contraction* of the smooth part of the objective. In Section 3.1, we develop new affine-invariant high-order algorithms for solving the composite convex minimization problems with bounded domain. In Section 3.2, we study the performance of the contracting second-order schemes. We propose new *accelerated* methods based on the contraction technique in Section 3.3.

Chapter 4 is devoted to *inexact* and *stochastic* versions of the methods. In Section 4.1, we study inexact high-order Tensor Methods. In Section 4.2, we investigate inexact contracting second-order method, whose steps are computed using a first-order gradient-based algorithm. We develop stochastic variants of our contracting second-order schemes in Section 4.3.

Chapter 5 contains final discussion of our results and highlights some possible directions for the future research.

1.1 Overview of the Contributions

Our thesis is based on new results published in six papers in the leading peer-reviewed journals of Mathematical Optimization and Machine Learning. These contributions can be summarized as follows.

Global performance of Cubic Newton for uniformly convex problems. We introduce the notion of second-order *condition number* for uniformly convex functions with Hölder continuous Hessian of degree $\nu \in [0, 1]$, and the corresponding degree of uniform convexity is $q = 2 + \nu$. We show that a regularized Newton scheme achieves the global linear rate of convergence for these problem classes, and the condition number plays the role of the main complexity factor. Then we establish this rate for the *adaptive* Cubic Newton Method which does not depend on any parameters of the problem class (*automatically* achieving the best complexity estimate). As a by-product of our developments, we justify an intuitively plausible result that the global iteration complexity of the Cubic Newton is always better than that of the Gradient Method on the class of strongly convex functions with uniformly bounded second derivative.

We present these results in Section 2.1 based on the paper:

- Nikita Doikov and Yurii Nesterov. *Minimizing uniformly convex functions by cubic regularization of Newton method,* Journal of Optimization Theory and Applications, 2021 [42].

Local convergence of Tensor Methods. We study local convergence of high-order Tensor Methods. We justify local superlinear convergence for the methods of order $p \geq 2$, in the case when the composite objective is uniformly convex of arbitrary degree q from the interval $2 \leq q < p + 1$. For strongly convex functions ($q = 2$), this gives the local rate of order p. This convergence is established both in the function value and in the norm of minimal subgradient. Then we discuss the global complexity bounds for the Tensor Method in convex and uniformly convex cases. Lastly, we show how local convergence of the methods can be globalized by using inexact Proximal-Point iterations.

These results are presented in Section 2.2 based on the paper:

- Nikita Doikov and Yurii Nesterov. *Local convergence of tensor methods,* Mathematical Programming, 2021 [41].

New affine-invariant second- and high-order methods. We develop new affine-invariant algorithms for solving the composite convex minimization problem with *bounded domain*. We present a general framework of *Contracting-Point Methods*, which solve at each iteration an auxiliary subproblem restricting the smooth part of the objective function onto contraction of the initial domain. This framework provides us with a systematic way for developing optimization methods of different order, endowed with the global complexity bounds. We show that using an appropriate affine-invariant smoothness condition, it is possible to implement one iteration of the Contracting-Point Method by one step of the pure tensor method of degree $p \geq 1$. The resulting global rate of convergence in functional residual is then $\mathcal{O}(1/k^p)$, where k is the iteration counter. It is important that all constants in our bounds are affine-invariant. For $p = 1$, our scheme recovers the well-known Frank-Wolfe algorithm, providing it with a new interpretation by a general perspective of tensor methods. For $p = 2$, we obtain new second-order scheme called *Contracting Newton Method*, which has global convergence of the order $\mathcal{O}(1/k^2)$. It can be seen as an implementation of the *trust-region idea*.

Further, we study a performance of the contracting second-order schemes under the assumption of Hölder continuous Hessian of degree $\nu \in [0, 1]$ (w.r.t. arbitrary norm). First, we introduce a new global second-order *lower* model of a smooth function. Then, we show that the Contracting Newton Method at every iteration minimizes this lower approximation of the smooth component of the objective augmented by the composite term. We prove the global rate of the order $\mathcal{O}(1/k^{1+\nu})$ in the general convex case. For strongly convex functions, we establish $\mathcal{O}(1/k^{2+2\nu})$ for the universal scheme. And if the parameters of the problem class are known, we can prove a global linear rate. Finally, we present aggregated models which accumulate second-order information into *quadratic Estimating Functions*. This leads to another optimization process, called *Aggregating Newton Method*, with the global convergence of the same order $\mathcal{O}(1/k^{1+\nu})$ as for general convex case. The latter method can be seen as a second-order counterpart of the dual averaging gradient schemes [112, 116].

These results are presented in Sections 3.1, 3.2 and based on the papers:

- Nikita Doikov and Yurii Nesterov. *Convex optimization based on global lower second-order models,* Advances in Neural Information Processing Systems (NeurIPS), 2020 [39].

- Nikita Doikov and Yurii Nesterov. *Affine-invariant contracting-point* *methods for convex optimization,* CORE Discussion Papers 2020/29 [37].

New contracting proximal algorithms. Utilizing contraction technique, we propose new *accelerated* methods for Smooth Convex Optimization called *Contracting Proximal Methods.* At every step, we need to minimize a contracted version of the objective function augmented by a regularization term in the form of Bregman divergence. This approach can be interpreted as a combination of *contracting-point* and *proximal-point* ideas. For our general scheme, we provide global convergence analysis admitting inexactness in solving the auxiliary subproblem. In the case of using for this purpose the basic Tensor Method of order $p \geq 1$, we demonstrate an acceleration effect for both convex and uniformly convex composite objective function. The global convergence of the resulting scheme is $\mathcal{O}(1/k^{p+1})$ in the general convex case. Thus, our construction explains acceleration for methods of any order starting from one. The augmentation of the number of calls of oracle due to computing the contracted proximal steps, is limited by the logarithmic factor in the worst-case complexity bound.

We present these results in Section 3.3 based on the paper:

- Nikita Doikov and Yurii Nesterov. *Contracting proximal methods for smooth convex optimization,* SIAM Journal on Optimization, 2020 [38].

Efficient inexact and stochastic second- and high-order methods with global complexity guarantees. First, we study inexact high-order Tensor Methods. At every step of such methods, we use the approximate solution to the auxiliary problem, defined by the bound for the residual in function value. We propose two *dynamic* strategies for choosing the inner accuracy: the first one is decreasing as $1/k^{p+1}$, where $p \geq 1$ is the order of the method and k is the iteration counter, and the second approach is using for the inner accuracy the last progress in the target objective. We show that inexact Tensor Methods with these strategies achieve the same global convergence rate as in the error-free case. For the second approach, when objective is strongly convex, we establish global linear rates as well, and local superlinear rates when $p \geq 2$. We also consider acceleration of inexact Tensor Methods, using our Contracting Proximal iteration with dynamic condition of inexactness defined in terms of the residual in function value. Lastly, we present computational results on a variety of machine learning problems for several methods and different accuracy policies.

Then we propose a two-level optimization scheme, which is the implementation of the inexact Contracting Newton Method, via computing its steps by the first-order Conditional Gradient Method. For the resulting algorithm, we establish the global complexity $\mathcal{O}(\varepsilon^{-1/2})$ calls of the *second-order local oracle* (computing the gradient and the Hessian of the smooth part of the objective), and $\mathcal{O}(\varepsilon^{-1})$ calls of the *linear minimization oracle* of the composite part, where $\varepsilon > 0$ is the required accuracy in the functional residual. Additionally, we address efficient implementation of our method for optimization over the standard simplex. Numerical experiments with our scheme confirm its good practical performance both in the number of iterations, and in computational time.

Finally, we consider the problem of finite-sum minimization. We develop stochastic extensions of our Contracting Newton Method. During the iterations of the basic variant, we need to increase the batch size for randomized estimates of gradients and Hessians up to the order $\mathcal{O}(k^4)$ and $\mathcal{O}(k^2)$ respectively. Using the *variance reduction* technique [141] for the gradients, we reduce the batch size up to the level $\mathcal{O}(k^2)$ for both estimates. At the same time, the global convergence rate of the resulting methods is of the order $\mathcal{O}(1/k^2)$, as for general convex functions with Lipschitz continuous Hessian. We present computational results for solving empirical risk minimization problem, comparing new second-order algorithms with stochastic first-order methods.

These results are presented in Chapter 4. Section 4.1 is based on the paper:

- Nikita Doikov and Yurii Nesterov. *Inexact tensor methods with dynamic accuracies,* International Conference on Machine Learning (ICML), 2020 [40].

Sections 4.2 and 4.3 are based on the aforementioned papers [37, 39], respectively.

1.2 Table of Algorithms

In the following table, we list optimization algorithms that we analyse in this thesis. The methods from Chapters 3 and 4 are all *new*.

Chapter 1	
Gradient Method	(1.3.10)
Newton's Method	(1.4.1)
Damped Newton	(1.4.4)
Cubic Newton	(1.4.9)
Tensor Method	(1.5.1)
Computing Inexact Tensor Step for $p = 3$	(1.5.3)
Chapter 2	
Regularized Newton	(2.1.19)
Adaptive Cubic Newton	(2.1.22)
Proximal-Point Method	(2.2.26)
Chapter 3	
Conceptual Contracting-Point Methods I, II	(3.1.4), (3.1.10)
Contracting-Point Tensor Methods I, II	(3.1.22), (3.1.24)
Contracting Newton I, II	(3.2.10), (3.2.23)
Aggregating Newton	(3.2.30)
Contracting Proximal Method	(3.3.22)
Contracting Proximal Tensor Method	(3.3.58)
Chapter 4	
Monotone Inexact Tensor Methods I, II	(4.1.4), (4.1.14)
Inexact Tensor Method with Averaging	(4.1.31)
Inexact Accelerated Scheme	(4.1.35)
Inexact Contracting Newton	(4.2.2)
Stochastic Contracting Newton	(4.3.2)
Stochastic Variance-Reduced Contracting Newton	(4.3.14)

Now, let us present *global rates* of convergence in terms of the functional residual for different first-order, second-order, and tensor methods on *general convex functions*, which are several times differentiable. We use $\tilde{\mathcal{O}}(\cdot)$ to hide logarithmic terms that depend on the target accuracy. We denote by k the iteration counter. Our results are marked as **new**.

First-order methods

Method	**Rate**	**Assumption**	**Affine-invariant**
Gradient Method [117]	$\mathcal{O}(k^{-1})$	Lipschitz grad.	$-$
Frank-Wolfe Algorithm [52]	$\mathcal{O}(k^{-1})$	Bounded dom.	$+$
Fast Gradient Method [107]	$\mathcal{O}(k^{-2})$	Lipschitz grad.	$-$

Second-order methods

Newton's Method [117]	—		$+$
Prox. Point + Newton's **(new)**	$\tilde{\mathcal{O}}(k^{-1.5})$	Lipschitz Hess.	$-$
Cubic Newton [124]	$\mathcal{O}(k^{-2})$	Lipschitz Hess.	$-$
Contracting Newton **(new)**	$\mathcal{O}(k^{-2})$	Bounded dom.	$+$
Aggregating Newton **(new)**	$\mathcal{O}(k^{-2})$	Bounded dom.	$+$
Contracting Proximal Method + Cubic Newton **(new)**	$\tilde{\mathcal{O}}(k^{-3})$	Lipschitz Hess.	$-$
Accel. Cubic Newton [111]	$\mathcal{O}(k^{-3})$	Lipschitz Hess.	$-$
Accel. Cubic Newton + line search [101, 117]	$\tilde{\mathcal{O}}(k^{-3.5})$	Lipschitz Hess.	$-$
Third-order Prox. Point + second-order impl. [120]	$\tilde{\mathcal{O}}(k^{-4})$	Lipschitz grad. and third deriv.	$-$
Third-order Prox. Point + second-order impl. + line search [121]	$\tilde{\mathcal{O}}(k^{-5})$	Lipschitz grad. and third deriv.	$-$

Tensor methods of order $p \geq 1$

Prox. Point + Tensor **(new)**	$\tilde{\mathcal{O}}(k^{-\frac{p+1}{2}})$	Lipschitz p-th deriv.	$-$
Basic Tensor Method [118]	$\mathcal{O}(k^{-p})$	Lipschitz p-th deriv.	$-$
Contracting-Point Tensor Method **(new)**	$\mathcal{O}(k^{-p})$	Bounded dom.	$+$
Contracting Proximal Method + Tensor Method **(new)**	$\tilde{\mathcal{O}}(k^{-(p+1)})$	Lipschitz p-th deriv.	$-$
Accel. Tensor [118]	$\mathcal{O}(k^{-(p+1)})$	Lipschitz p-th deriv.	$-$
Accel. Tensor + line search [54]	$\tilde{\mathcal{O}}(k^{-\frac{3p+1}{2}})$	Lipschitz p-th deriv.	$-$

We discuss elements from the tables in the further sections of the thesis.

1.3 Smooth Convex Optimization

We start with a formal statement of our target optimization problem, and specify the basic notation which is necessary for all statements in the thesis. The definition of Lipschitz continuity is coupled with a number of examples and some useful properties. Then, we briefly review the classical Gradient Method as an introduction into the subject of Smooth Convex Optimization. For an exhaustive study of the topic we refer to the classic books and lecture notes [117, 10, 14, 127, 130]. The Gradient Method serves as an important baseline to our further developments in second-order and high-order methods.

1.3.1 Preliminaries and Notation

We denote by $\mathbb{E}$ a finite-dimensional real vector space. Then, our main problem of interest can be formulated in the *composite form*, as follows:

$$\min_x \Big\{ F(x) \overset{\mathrm{def}}{=} f(x) + \psi(x) \Big\}, \tag{1.3.1}$$

where $\psi : \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ is a *simple* proper closed convex function, and function f is convex and several times continuously differentiable at every point $x \in \operatorname{dom} \psi = \{x \in \mathbb{E} \; : \; \psi(x) < +\infty\}$.

Example 1.3.1. When $\psi(x) \equiv 0$, (1.3.1) becomes the unconstrained minimization problem with a smooth convex objective:

$$\min_{x \in \mathbb{E}} f(x).$$

Example 1.3.2. Let $Q \subseteq \mathbb{E}$ be a *simple* closed convex set, and ψ be its $\{0, +\infty\}$-indicator:

$$\psi(x) \;\; = \;\; \begin{cases} 0, & x \in Q, \\ +\infty, & \text{otherwise}. \end{cases}$$

Then, problem (1.3.1) is to minimize f over Q:

$$\min_{x \in Q} f(x).$$

Example 1.3.3. Let $\mathbb{E} = \mathbb{R}^n$ and

$$\psi(x) \quad = \quad \|x\|_1 \quad \overset{\text{def}}{=} \quad \sum_{i=1}^{n} |x^{(i)}|.$$

Then, (1.3.1) is a problem with ℓ_1-*regularization.*

Thus, the framework of composite optimization [114] provides a unified way to treat the constrained problems and the problems with explicit non-differentiable components. The main requirement is that ψ should have a *simple* structure, which means that corresponding auxiliary subproblems are efficiently solvable. We will see examples of subproblems when discussing the methods. Typically, we substitute some model for f in (1.3.1), while the composite component ψ remains unchanged.

Having fixed the primal vector space $\mathbb{E}$, we denote by $\mathbb{E}^*$ its dual space, which is a space of linear functions on $\mathbb{E}$. The value of linear function $s \in \mathbb{E}^*$ on vector $x \in \mathbb{E}$ is denoted by $\langle s, x \rangle \overset{\text{def}}{=} s(x)$. Of course, one can always identify $\mathbb{E}$ and $\mathbb{E}^*$ with $\mathbb{R}^n$, when some basis is fixed, but often it is useful to separate these spaces, in order to avoid ambiguities.

For a smooth function $f : \operatorname{dom} f \to \mathbb{R}$, where $\operatorname{dom} f \subseteq \mathbb{E}$ is open, we denote by $\nabla f(x)$ its gradient and by $\nabla^2 f(x)$ its Hessian, evaluated at point $x \in \operatorname{dom} f \subseteq \mathbb{E}$. Note that

$$\nabla f(x) \quad \in \quad \mathbb{E}^*, \qquad \nabla^2 f(x) h \;\in\; \mathbb{E}^*,$$

for all $h \in \mathbb{E}$. For $p \geq 1$, we denote by $D^p f(x)[h_1, \ldots, h_p]$ the p-th directional derivative of f along directions $h_1, \ldots, h_p \in \mathbb{E}$. Note that $D^p f(x)$ is a p-linear symmetric form on $\mathbb{E}$. If $h_i = h$ for all $1 \leq i \leq p$, a shorter notation $D^p f(x)[h]^p$ is used. For its gradient in h, we use the following notation:

$$D^p f(x)[h]^{p-1} \quad \overset{\text{def}}{=} \quad \tfrac{1}{p} \nabla_h \, D^p f(x)[h]^p \quad \in \quad \mathbb{E}^*, \qquad h \in \mathbb{E}.$$

In particular, $D^1 f(x)[h]^0 \equiv \nabla f(x)$, and $D^2 f(x)[h]^1 \equiv \nabla^2 f(x) h$.

For a convex but not necessary differentiable function ψ, we denote by $\partial \psi(x) \subseteq \mathbb{E}^*$ its subdifferential at point $x \in \operatorname{dom} \psi \subseteq \mathbb{E}$:

$$\partial \psi(x) \quad \overset{\text{def}}{=} \quad \bigl\{ g \in \mathbb{E}^* \;:\; \forall y \in \operatorname{dom} \psi \; (\psi(y) \geq \psi(x) + \langle g, y - x \rangle) \bigr\}.$$

We denote by x^* a solution to problem (1.3.1), assuming that it exists:

$$x^* \quad \in \quad \operatorname*{Argmin}_x F(x), \qquad F^* \quad \overset{\mathrm{def}}{=} \quad F(x^*).$$

Then, this point satisfies the following optimality condition (see, e.g. Theorem 3.1.23 in [117]):

$$\langle \nabla f(x^*), x - x^* \rangle + \psi(x) \quad \geq \quad \psi(x^*), \qquad x \in \operatorname{dom} \psi. \tag{1.3.2}$$

In other words, the following inclusion holds:

$$-\nabla f(x^*) \quad \in \quad \partial \psi(x^*).$$

From now on, let us fix some self-adjoint positive-definite linear operator $B : \mathbb{E} \to \mathbb{E}^*$ (notation $B = B^* \succ 0$). We use it to endow the primal space with the Euclidean norm:

$$\|x\| \quad \overset{\mathrm{def}}{=} \quad \langle Bx, x \rangle^{1/2}, \qquad x \in \mathbb{E}.$$

Then, the norm for the dual space is induced in the standard way,

$$\|s\|_* \quad \overset{\mathrm{def}}{=} \quad \max_{h \in \mathbb{E}} \bigl\{ \langle s, h \rangle \ : \ \|h\| \leq 1 \bigr\} \quad = \quad \langle s, B^{-1} s \rangle^{1/2}, \qquad s \in \mathbb{E}^*.$$

In what follows, we work with the Euclidean norms, unless the contrary is explicitly stated (we will consider general norms in Chapter 3).

For any linear operator $A : \mathbb{E} \to \mathbb{E}^*$ its norm is defined as

$$\|A\| \quad \overset{\mathrm{def}}{=} \quad \max_{h \in \mathbb{E}} \bigl\{ \|Ah\|_* \ : \ \|h\| \leq 1 \bigr\}.$$

Similarly, the norm of $D^p f(x)$ for any $p \geq 1$ is induced by the Euclidean norm for the primal space, as follows:

$$\begin{aligned} \|D^p f(x)\| \quad &\overset{\mathrm{def}}{=} \quad \max_{h_1, \ldots, h_p \in \mathbb{E}} \Bigl\{ D^p f(x)[h_1, \ldots, h_p] \ : \ \forall i \ (\|h_i\| \leq 1) \Bigr\} \\ &= \quad \max_{h \in \mathbb{E}} \Bigl\{ |D^p f(x)[h]^p| \ : \ \|h\| \leq 1 \Bigr\}. \end{aligned}$$

See Appendix 1 in [123] for the proof of the last equation, which is valid for any multilinear symmetric form.

The norm can be used to characterize the smoothness of our objective.

We say that for some $p \geq 1$, the p-th derivative of f is *Lipschitz continuous* on a convex set $Q \subseteq \operatorname{dom} f$, if for all $x, y \in Q$, it holds

$$\begin{aligned} & \|D^p f(x) - D^p f(y)\| \\ \overset{\text{def}}{=} \quad & \max_{h \in \mathbb{E}} \Bigl\{ |D^p f(x)[h]^p - D^p f(y)[h]^p| \; : \; \|h\| \leq 1 \Bigr\} \\ \leq \quad & L_p \|x - y\|, \end{aligned} \tag{1.3.3}$$

with some positive constant L_p. For $p = 1$, we get the functions with Lipschitz continuous gradient, and for $p = 2$, with Lipschitz continuous Hessian.

Let Q be a convex set. For k times continuously differentiable on Q functions, whose p-th derivative ($p \leq k$) is Lipschitz continuous, the standard notation is

$$f \quad \in \quad C^{k,p}(Q).$$

When $p < k$ and Q is an open convex set, Lipschitz continuity is equivalent to the boundness of the higher $(p+1)$th derivative. This fact can be useful for computing the corresponding Lipschitz constants.

Example 1.3.4. For the power of the Euclidean norm

$$f(x) = \tfrac{1}{p+1} \|x - x_0\|^{p+1}, \qquad p \geq 1, \qquad x, x_0 \in \mathbb{E},$$

(1.3.3) holds for all $x, y \in \mathbb{E}$ with $L_p = p!$ (see Theorem 7.1 in [136]).

Example 1.3.5. For given linear functions $a_i \in \mathbb{E}^*$, $1 \leq i \leq m$, consider the following convex function (SoftMax):

$$f(x) \quad = \quad \log\Bigl(\sum_{i=1}^m e^{\langle a_i, x \rangle} \Bigr), \qquad x \in \mathbb{E}.$$

Let us use operator $B = \sum_{i=1}^m a_i a_i^* : \mathbb{E} \to \mathbb{E}^*$, that is defined by the equation

$$Bh \quad = \quad \sum_{i=1}^m a_i a_i^* h \quad = \quad \sum_{i=1}^m \langle a_i, h \rangle a_i, \qquad \forall h \in \mathbb{E}.$$

We assume that $B \succ 0$ (i.e. $\langle Bh, h \rangle > 0$ for any $h \in \mathbb{E}$), otherwise we can reduce dimensionality of the problem. Then, (1.3.3) holds for all $x, y \in \mathbb{E}$ with

$$L_1 \quad = \quad 1, \qquad L_2 \quad = \quad 2, \qquad L_3 \quad = \quad 4.$$

Proof. Denote $\kappa(x) = \sum_{i=1}^m e^{\langle a_i, x\rangle}$. Let us fix arbitrary $x, y \in \mathbb{E}$ and direction $h \in \mathbb{E}$. Then, straightforward computation gives:

$$\begin{aligned}
\langle \nabla f(x), h\rangle &= \tfrac{1}{\kappa(x)} \textstyle\sum_{i=1}^m e^{\langle a_i,x\rangle}\langle a_i, h\rangle, \\
\langle \nabla^2 f(x)h, h\rangle &= \tfrac{1}{\kappa(x)} \textstyle\sum_{i=1}^m e^{\langle a_i,x\rangle}\langle a_i, h\rangle^2 - \left(\tfrac{1}{\kappa(x)} \textstyle\sum_{i=1}^m e^{\langle a_i,x\rangle}\langle a_i, h\rangle\right)^2 \\
&= \tfrac{1}{\kappa(x)} \textstyle\sum_{i=1}^m e^{\langle a_i,x\rangle}\left(\langle a_i, h\rangle - \langle \nabla f(x), h\rangle\right)^2 \;\geq\; 0.
\end{aligned}$$

Hence, we get,

$$\begin{aligned}
\|\nabla^2 f(x)\| &= \max_{\|h\|\leq 1} \langle \nabla^2 f(x)h, h\rangle \;\leq\; \max_{\|h\|\leq 1} \textstyle\sum_{i=1}^m \langle a_i, h\rangle^2 \\
&= \max_{\|h\|\leq 1} \|h\|^2 \;=\; 1.
\end{aligned}$$

Thus we obtain $L_1 = 1$. For higher derivatives, we have the following representations:

$$\begin{aligned}
D^3 f(x)[h]^3 &= \tfrac{1}{\kappa(x)} \sum_{i=1}^m e^{\langle a_i,x\rangle}\left(\langle a_i, h\rangle - \langle \nabla f(x), h\rangle\right)^3 \\
&\leq \langle \nabla^2 f(x)h, h\rangle \max_{1\leq i,j\leq m} \langle a_i - a_j, h\rangle \;\leq\; 2\|h\|^3,
\end{aligned}$$

and

$$\begin{aligned}
D^4 f(x)[h]^4 &= \tfrac{1}{\kappa(x)} \sum_{i=1}^m e^{\langle a_i,x\rangle}\left(\langle a_i, h\rangle - \langle \nabla f(x), h\rangle\right)^4 - 3\langle \nabla^2 f(x)h, h\rangle^2 \\
&\leq D^3 f(x)[h]^3 \max_{1\leq i,j\leq m} \langle a_i - a_j, h\rangle \;\leq\; 4\|h\|^4,
\end{aligned}$$

which give $L_2 = 2$ and $L_3 = 4$. □

Example 1.3.6. Using $\mathbb{E} = \mathbb{R}$, and $a_1 = 0, a_2 = 1$ in the previous example, we obtain the logistic regression loss function:

$$f(x) \;=\; \log(1 + e^x), \qquad x \in \mathbb{R}.$$

However, a more specific analysis provides us with the following estimates for its Lipschitz constants, which are tight:

$$L_1 \;=\; \tfrac{1}{4}, \qquad L_2 \;=\; \tfrac{1}{6\sqrt{3}}, \qquad L_3 \;=\; \tfrac{1}{8}.$$

Proof. Denote $\sigma(x) = f'(x) = \frac{1}{1+e^{-x}}$.

Direct calculations give:

$$\begin{aligned}
\sigma'(x) &= \sigma(x)\cdot(1-\sigma(x)),\\
\sigma''(x) &= \sigma'(x)\cdot(1-2\sigma(x)),\\
\sigma'''(x) &= \sigma'(x)\cdot(6\sigma(x)^2 - 6\sigma(x)+1)\\
&= 6\sigma'(x)\cdot(\sigma(x)-\tfrac{1}{2}-\tfrac{1}{2\sqrt{3}})\cdot(\sigma(x)-\tfrac{1}{2}+\tfrac{1}{2\sqrt{3}}),\\
\sigma^{(4)}(x) &= \sigma''(x)\cdot(12\sigma(x)^2-12\sigma(x)+1)\\
&= \sigma''(x)\cdot(\sigma(x)-\tfrac{1}{2}-\tfrac{1}{\sqrt{6}})\cdot(\sigma(x)-\tfrac{1}{2}+\tfrac{1}{\sqrt{6}}).
\end{aligned}$$

Hence, considering the stationary points, we get

$$\begin{aligned}
\max_{x\in\mathbb{R}} f''(x) &= \max_{x\in\mathbb{R}} \sigma'(x) \;=\; \alpha\cdot(1-\alpha)\big|_{\alpha=\frac{1}{2}} \;=\; \tfrac{1}{4},\\
\max_{x\in\mathbb{R}} |f'''(x)| &= \max_{x\in\mathbb{R}} |\sigma''(x)|\\
&= \max\big\{|\alpha\cdot(1-\alpha)\cdot(1-2\alpha)| \;:\; \alpha = \tfrac{1}{2}\pm\tfrac{1}{2\sqrt{3}}\big\}\\
&= \Big|\big(\tfrac{1}{2}+\tfrac{1}{2\sqrt{3}}\big)\cdot\big(\tfrac{1}{2}-\tfrac{1}{2\sqrt{3}}\big)\cdot\tfrac{1}{\sqrt{3}}\Big| \;=\; \tfrac{1}{6\sqrt{3}},
\end{aligned}$$

and finally

$$\begin{aligned}
\max_{x\in\mathbb{R}} |f^{(4)}(x)| &= \max_{x\in\mathbb{R}} |\sigma'''(x)|\\
&= \max\big\{|\alpha\cdot(1-\alpha)\cdot(6\alpha^2-6\alpha+1)| \;:\; \alpha\in\{\tfrac{1}{2}, \tfrac{1}{2}\pm\tfrac{1}{\sqrt{6}}\}\big\}\\
&= |\alpha\cdot(1-\alpha)\cdot(6\alpha^2-6\alpha+1)|\,\Big|_{\alpha=\frac{1}{2}} \;=\; \tfrac{1}{8}.
\end{aligned}$$

□

Taylor's polynomial is a standard tool of Numerical Analysis. For a smooth function f, integer number $p \geq 1$, and given $x \in \operatorname{dom} f$, denote

$$\Omega_p(f,x;y) \overset{\text{def}}{=} f(x) + \sum_{i=1}^{p} \tfrac{1}{i!} D^i f(x)[y-x]^i. \tag{1.3.4}$$

For $f \in C^{p,p}(Q)$, we can globally bound the residual between the function and its polynomial approximation $\Omega_p(f,x;y) \approx f(y)$, as follows.

Lemma 1.3.7. *For all $x, y \in Q$, it holds*

$$|f(y) - \Omega_p(f,x;y)| \leq \tfrac{L_p \|y-x\|^{p+1}}{(p+1)!}, \tag{1.3.5}$$

$$\|\nabla f(y) - \nabla \Omega_p(f,x;y)\|_* \leq \tfrac{L_p \|y-x\|^{p}}{p!}, \tag{1.3.6}$$

$$\|\nabla^2 f(y) - \nabla^2 \Omega_p(f,x;y)\| \leq \tfrac{L_p \|y-x\|^{p-1}}{(p-1)!}. \tag{1.3.7}$$

Proof. Indeed, by Taylor's theorem, we have

$$\begin{array}{rl}
& |f(y) - \Omega_p(f,x;y)| \\
= & |\int\limits_0^1 \tfrac{(1-\tau)^{p-1}}{(p-1)!} D^p f(x+\tau(y-x))[y-x]^p d\tau - \tfrac{1}{p!} D^p f(x)[y-x]^p| \\
= & |\int\limits_0^1 \tfrac{(1-\tau)^{p-1}}{(p-1)!} \big(D^p f(x+\tau(y-x))[y-x]^p - D^p f(x)[y-x]^p \big) d\tau| \\
\overset{(1.3.3)}{\leq} & \tfrac{L_p \|y-x\|^{p+1}}{(p-1)!} \int\limits_0^1 (1-\tau)^{p-1} \tau d\tau = \tfrac{L_p \|y-x\|^{p+1}}{(p+1)!}.
\end{array}$$

Applying the same reasoning to functions $\langle \nabla f(\cdot), h \rangle$ and $\langle \nabla^2 f(\cdot) h, h \rangle$ with direction $h \in \mathbb{E}$ being fixed, we get (1.3.6) and (1.3.7). □

We say that a differentiable function f is *strongly convex* on a convex set $Q \subseteq \operatorname{dom} f$ if it satisfies inequality

$$f(y) \geq f(x) + \langle \nabla f(x), y-x \rangle + \tfrac{\mu \|y-x\|^2}{2}, \qquad \forall x, y \in Q, \tag{1.3.8}$$

for some constant $\mu > 0$. For a strongly convex objective, the solution to minimization problem (1.3.1) always exists and unique [117].

Sometimes, we need to use the following technical bound. It can be immediately seen from the geometrical meaning of integration.

Lemma 1.3.8. *For every $s > 1$, the following inequality holds:*

$$\sum_{i=1}^{k} \frac{1}{i^s} \quad \leq \quad \frac{s}{s-1}, \qquad \forall k \geq 1. \tag{1.3.9}$$

Proof. The statement easily follows by observing that

$$\sum_{i=1}^{k} \frac{1}{i^s} \quad = \quad 1 + \sum_{i=2}^{k} \frac{1}{i^s} \quad \leq \quad 1 + \int_{1}^{+\infty} \frac{dx}{x^s} \quad = \quad \frac{s}{s-1}. \qquad \square$$

1.3.2 The Gradient Method

A classical first-order optimization scheme is called the *Gradient Method.* At each iteration, we substitute the linear approximation with a quadratic regularizer for the smooth part of the composite objective. Then, we use the minimum of the current model as the next point of the process:

$$\boxed{\begin{aligned} x_0 \quad &\in \quad \mathrm{dom}\,\psi, \qquad k \geq 0 : \\ x_{k+1} \quad &= \quad \operatorname*{argmin}_{y} \Big\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \tfrac{H_k \|y - x_k\|^2}{2} \\ &\qquad + \psi(y) \Big\}. \end{aligned}} \tag{1.3.10}$$

When $\psi(x) \equiv 0$, one iteration of this method can be rewritten in the following canonical form:

$$x_{k+1} \quad = \quad x_k - \tfrac{1}{H_k} B^{-1} \nabla f(x_k).$$

Therefore, operator B plays the role of a fixed *preconditioner.* For a particular problem instance, we can try to pick it in a way to improve the smoothness characteristics of the objective (see Example 1.3.5).

In the general case, performing the composite gradient step for arbitrary ψ is related to computing the corresponding prox-operator [8].

Parameter H_k should be chosen so as to have a significant decrease in the function value at every iteration. This can be achieved by ensuring the following upper bound:

$$f(x_{k+1}) \quad \leq \quad f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \tfrac{H_k \|x_{k+1} - x_k\|^2}{2}, \qquad (1.3.11)$$

which holds for the constant step size rule $H_k \equiv L_1$ due to inequality (1.3.5). If the Lipschitz constant is not known, a simple adaptive search for H_k can be performed to ensure (1.3.11) (see [114]). Alternative strategies for choosing H_k include the Polyak's stepsize rule [129, 130] (see also [70]), and the adaptive rule from [99].

Let us assume that the gradient of the smooth part is Lipschitz continuous ($L_1 < +\infty$). Then, the Gradient Method (1.3.10) needs

$$K \quad = \quad \mathcal{O}\Bigl(\tfrac{L_1 \|x_0 - x^*\|^2}{\varepsilon} \Bigr) \qquad (1.3.12)$$

iterations to find an ε-solution in the functional residual: $F(x_K) - F^* \leq \varepsilon$ [114]. For strongly convex functions, the rate of convergence is *linear*, and the corresponding iteration complexity up to logarithmic factors depends only on the *condition number* of the problem:

$$K \quad = \quad \mathcal{O}\Bigl(\tfrac{L_1}{\mu} \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr). \qquad (1.3.13)$$

We recover these complexity estimates in Theorems 2.2.7 and 2.2.9 from Chapter 2 as a particular case $p = 1$.

For the same problem classes, we can get an accelerated rate of convergence by using the Fast Gradient Method [107, 114], achieving

$$\mathcal{O}\Bigl(\sqrt{\tfrac{L_1 \|x_0 - x^*\|^2}{\varepsilon}} \Bigr) \qquad \text{and} \qquad \mathcal{O}\Bigl(\sqrt{\tfrac{L_1}{\mu}} \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr)$$

complexity estimates, correspondingly. These rates are known to be optimal for the first-order black-box optimization [106, 102].

An additional advantage of using the adaptive search over parameter H_k within the gradient methods is that the algorithm becomes *universal*, meaning that it can automatically adapt to different problem classes [115]. We discover this phenomenon in the context of regularized Newton methods in Section 2.1 of Chapter 2.

See also [36, 34] for studies of the gradient methods with inexact or stochastic first-order oracle information.

1.4 Second-Order Methods in Optimization

Now we are going to review second-order numerical optimization schemes. In such algorithms at each iteration we can exploit *second-order local oracle* for the smooth part of the objective. The oracle returns the function value, the gradient and the Hessian, computed at the given point. Having more information about the objective, the second-order algorithms are expected to be more powerful than the gradient methods.

1.4.1 Newton's Method

In the classical Newton's Method, we approximate the smooth part f of the objective by its second-order Taylor's polynomial evaluated at the current point. The composite part ψ remains unchanged. The next point is defined as a minimum of this model (we assume that a finite minimum exists). Hence, iterations of Newton's Method for solving problem (1.3.1) can be represented as follows:

$$\boxed{\begin{array}{rcl} x_0 & \in & \operatorname{dom}\psi, \qquad k \geq 0: \\[1em] x_{k+1} & \in & \operatorname*{Argmin}_y \Big\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle \\[0.5em] & & \qquad + \tfrac{1}{2}\langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \psi(y) \Big\}. \end{array}} \tag{1.4.1}$$

When $\psi(x) \equiv 0$ and the Hessian is invertible, the step of the method can be rewritten in a shorter form:

$$x_{k+1} \quad = \quad x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

Comparing with the Gradient Method, each step of method (1.4.1) is obviously more expensive. However, we can hope that the second-order information may significantly accelerate the rate of convergence.

The standard and well-known result about Newton's Method is its local quadratic convergence [78, 117]. Later on, it was generalized to the case of composite optimization problems [89]. Local superlinear convergence of the Incremental Newton method for finite-sum minimization problems was established in [134]. See [137, 138] for a modern study of the local superlinear convergence of the quasi-Newton methods.

Let us present a simple proof of the local quadratic rate for method (1.4.1).

Theorem 1.4.1. *Let $f \in C^{2,2}(\mathrm{dom}\,\psi)$, so the Hessian $\nabla^2 f(\cdot)$ is Lipschitz continuous on $\mathrm{dom}\,\psi$ with constant $0 < L_2 < +\infty$.*

Let x^* be a solution to problem (1.3.1) with $\nabla^2 f(x^*) \succeq \mu B$ for some positive μ.

Let x_k belong to a neighbourhood of x^*:

$$x_k \quad \in \quad U \quad \overset{\mathrm{def}}{=} \quad \Big\{ x \in \mathrm{dom}\,\psi \ : \ \|x - x^*\| \leq \tfrac{2\mu}{3L_2} \Big\}.$$

Then, for one step of method (1.4.1), we have $x_{k+1} \in U$ and the rate of convergence is quadratic:

$$\|x_{k+1} - x^*\| \quad \leq \quad \tfrac{L_2}{2(\mu - L_2\|x_k - x^*\|)} \|x_k - x^*\|^2. \tag{1.4.2}$$

Proof. Let us plug $x = x_{k+1}$ into the stationary condition (1.3.2) for x^*. Thus we get

$$\langle \nabla f(x^*), x_{k+1} - x^* \rangle + \psi(x_{k+1}) \quad \geq \quad \psi(x^*). \tag{1.4.3}$$

At the same time, the stationary condition for one Newton's step is

$$\langle \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k), x - x_{k+1} \rangle + \psi(x) \quad \geq \quad \psi(x_{k+1}),$$

for all $x \in \mathrm{dom}\,\psi$. Summing up this inequality for $x = x^*$ with (1.4.3), we obtain

$$\begin{array}{rcl} 0 & \leq & \langle \nabla f(x^*) - \nabla f(x_k) - \nabla^2 f(x_k)(x_{k+1} - x_k), x_{k+1} - x^* \rangle \\[2ex] & = & \langle \nabla f(x^*) - \nabla f(x_k) - \nabla^2 f(x_k)(x^* - x_k), x_{k+1} - x^* \rangle \\[2ex] & & - \langle \nabla^2 f(x_k)(x_{k+1} - x^*), x_{k+1} - x^* \rangle \\[2ex] & \overset{(1.3.6)}{\leq} & \tfrac{L_2 \|x_k - x^*\|^2 \cdot \|x_{k+1} - x^*\|}{2} - \langle \nabla^2 f(x_k)(x_{k+1} - x^*), x_{k+1} - x^* \rangle. \end{array}$$

Hence, assuming the nontrivial case $x_{k+1} \neq x^*$, we get

$$\begin{array}{rcl} \tfrac{L_2\|x_k - x^*\|^2}{2} & \geq & \tfrac{\langle \nabla^2 f(x_k)(x_{k+1} - x^*), x_{k+1} - x^* \rangle}{\|x_{k+1} - x^*\|} \\[2ex] & \geq & \tfrac{\langle \nabla^2 f(x^*)(x_{k+1} - x^*), x_{k+1} - x^* \rangle - L_2 \|x_k - x^*\| \cdot \|x_{k+1} - x^*\|^2}{\|x_{k+1} - x^*\|} \\[2ex] & \geq & (\mu - L_2\|x_k - x^*\|) \cdot \|x_{k+1} - x^*\|, \end{array}$$

which proves the required bound. □

Remark 1.4.2. Let us denote $\beta \overset{\mathrm{def}}{=} \frac{2\mu}{3L_2}$ and $\delta_k \overset{\mathrm{def}}{=} \beta\|x_k - x^*\|$.

Assume $x_0 \in \operatorname{int} U$. Then, $\delta_0 < 1$. According to (1.4.2), it holds

$$\delta_k \;\; \leq \;\; \delta_{k-1}^2 \;\; \leq \;\; \delta_{k-2}^{2^2} \;\; \leq \;\; \delta_{k-3}^{2^3} \;\; \leq \;\; \ldots \;\; \leq \;\; \delta_0^{2^k}.$$

Therefore, we have $\|x_k - x^*\| \leq \varepsilon$ after

$$k \;\; \geq \;\; \log_2 \log_2 \tfrac{1}{\beta\varepsilon} - \log_2 \log_2 \tfrac{1}{\delta_0}.$$

iterations of Newton's Method. □

We see that the method starts to double the number of the right digits of the answer every step, when it enters the neighbourhood of x^*. This is a very fast convergence, and for all practical purposes a small number of steps is enough to solve the problem. However, there is no evidence how long it can take to enter U.

Example 1.4.3. Consider the following minimization problem:

$$\min_{x \in \mathbb{R}} \Big\{ f(x) \;\; = \;\; \log(1 + \exp(x)) - \tfrac{x}{2} + \tfrac{\mu x^2}{2} \Big\}.$$

Clearly, the objective is strongly convex with parameter $\mu > 0$, and its second derivative is Lipschitz continuous with constant $L_2 = \frac{1}{6\sqrt{3}}$ (Example 1.3.6). The optimal point is $x^* = 0$.

Hence, according to Theorem 1.4.1, the region of quadratic convergence of Newton's Method is $U = \{x \in \mathbb{R} \; : \; |x| \leq 4\sqrt{3}\mu\}$.

Let us fix $\mu = 10^{-2}$, and choose $x_0 = 50$, which is outside of the region. Then, one Newton's step produces the point

$$x_1 \;\; = \;\; x_0 - \tfrac{f'(x_0)}{f''(x_0)} \;\; = \;\; 50 - \tfrac{\sigma(50)}{\sigma(50)\cdot(1 - \sigma(50)) + 10^{-2}},$$

with $\sigma(50) = \frac{1}{1+e^{-50}} \overset{\boldsymbol{m}}{=} 1$ (in the machine precision). So, $x_1 \overset{\boldsymbol{m}}{=} -50$. Consequently,

$$x_2 \;\; = \;\; x_1 - \tfrac{f'(x_1)}{f''(x_1)} \;\; \overset{\boldsymbol{m}}{=} \;\; -50 - \tfrac{\sigma(-50) - 1}{\sigma(-50)\cdot(1 - \sigma(-50)) + 10^{-2}},$$

with $\sigma(-50) = \frac{1}{1+e^{50}} \overset{\boldsymbol{m}}{=} 0$, and thus $x_2 \overset{\boldsymbol{m}}{=} 50$.

Therefore, Newton's Method starts to oscillate between the points 50 and -50 (see Figure 1.1). □

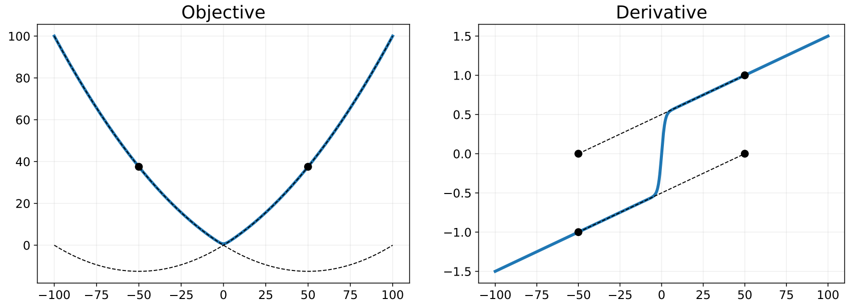


Figure 1.1: Oscillation of the classical Newton's Method.

1.4.2 Damped Newton Method

In order to globalize Newton's iterations, we may incorporate into the method an auxiliary sequence of positive coefficients $\{\gamma_k\}_{k\geq 0}$.

When $\gamma_k \leq 1$, the following algorithm is usually called the Damped Newton Method:

$$\boxed{\begin{array}{rcl} x_0 & \in & \operatorname{dom}\psi, \qquad k \geq 0: \\[2ex] x_{k+1} & \in & \underset{y}{\operatorname{Argmin}}\Big\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle \\[1ex] & & \qquad + \tfrac{1}{2\gamma_k}\langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \psi(y) \Big\}. \end{array}} \qquad (1.4.4)$$

Without the composite term, the iterations are as follows (assuming the Hessian is invertible):

$$x_{k+1} \quad = \quad x_k - \gamma_k(\nabla^2 f(x_k))^{-1}\nabla f(x_k).$$

Therefore, for $\gamma_k = 1$ we obtain the standard Newton's step. To ensure the global rate, we need to pick up the coefficients in a smarter way. A natural choice for γ_k is to certify that the new function value $f(x_{k+1})$ is upper bounded by the minimum of the Damped Newton model:

$$\begin{array}{rcl} f(x_{k+1}) & \leq & f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle \\ & & \quad + \tfrac{1}{2\gamma_k}\langle \nabla^2 f(x_k)(x_{k+1} - x_k), x_{k+1} - x_k \rangle. \end{array} \qquad (1.4.5)$$

Then we would have the progress in the objective value at each step of the method, and the following global convergence guarantee holds.

Theorem 1.4.4. *Let for some positive parameters μ and L_1,*

$$\mu B \;\; \preceq \;\; \nabla^2 f(x) \;\; \preceq \;\; L_1 B, \qquad \forall x \in \mathrm{dom}\,\psi. \tag{1.4.6}$$

Thus f is strongly convex, and we assume it belongs to $C^{2,1}(\mathrm{dom}\,\psi)$. Let inequality (1.4.5) be satisfied for some $\gamma_k \geq \frac{\mu}{2L_1}$.

Then, we have the global linear rate:

$$F(x_{k+1}) - F^* \;\; \leq \;\; \Bigl(1 - \tfrac{1}{8} \cdot \bigl(\tfrac{\mu}{L_1}\bigr)^2\Bigr) \cdot (F(x_k) - F^*). \tag{1.4.7}$$

Proof. Indeed, for arbitrary $y \in \mathrm{dom}\,\psi$, we have

$$\begin{array}{rcl} F(x_{k+1}) & = & f(x_{k+1}) + \psi(x_{k+1}) \\ & \overset{(1.4.6)}{\leq} & f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle \\ & & + \, \frac{1}{2\gamma_k} \langle \nabla^2 f(x_k)(x_{k+1} - x_k), x_{k+1} - x_k \rangle + \psi(x_{k+1}) \\ & \leq & f(x_k) + \langle \nabla f(x_k), y - x_k \rangle \\ & & + \, \frac{1}{2\gamma_k} \langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \psi(y) \\ & \leq & F(y) + \frac{L_1^2 \|y - x_k\|^2}{\mu}, \end{array}$$

where in the last inequality we used (1.4.6), the lower bound for γ_k, and the convexity of f.

Now, let us take $y = \alpha x^* + (1-\alpha)x_k$, for $\alpha = \frac{\mu^2}{4L_1^2} \in (0,1)$. Therefore, using the strong convexity of F, we obtain

$$\begin{array}{rcl} F(x_{k+1}) & \leq & \alpha F^* + (1-\alpha)F(x_k) + \frac{L_1^2 \alpha^2 \|x_k - x^*\|^2}{\mu} \\ & \leq & \alpha F^* + (1-\alpha)F(x_k) + \frac{2L_1^2}{\mu^2}\alpha^2 (F(x_k) - F^*). \end{array}$$

Hence,

$$F(x_{k+1}) - F^* \;\; \leq \;\; \bigl(1 - \alpha + \tfrac{2L_1^2}{\mu^2}\alpha^2\bigr) \cdot (F(x_k) - F^*).$$

Substituting the above value of α completes the proof. □

It is clear that the constant choice $\gamma_k \equiv \frac{\mu}{L_1}$ satisfies the conditions of Theorem 1.4.4. In practice though, we may run a simple adaptive search to fit the value of the coefficients (analogously to that one used in the gradient methods). In any case, it is important to switch regularly between the pure Newton's step $\gamma_k = 1$ to have the local superlinear guarantee as well.

According to the bound (1.4.7), we get $F(x_K) - F^* \leq \varepsilon$ after

$$K \quad = \quad \mathcal{O}\Bigl(\bigl(\tfrac{L_1}{\mu}\bigr)^2 \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr) \tag{1.4.8}$$

iterations of method (1.4.4). The good news is that this is a global guarantee and hence the Damped Newton Method converges to the optimum starting from an arbitrary initial point.

However, estimate (1.4.8) is much worse than the corresponding one (1.3.13) of the basic Gradient Method. This fact seems to be disappointing. Indeed, we use additional second-order information about the objective in algorithm (1.4.4), and the computations are more expensive. At the same time, from the theoretical perspective, we are not gaining any clear advantages until entering the region of quadratic convergence.

Another valuable observation is that the Newton's Method is entirely defined using only *affine-invariant* objects. Thus it does not depend on the choice of coordinate system or particular norms. Theorem 1.4.1 and Theorem 1.4.4 both use an artificial operator B (which we fix to define the Euclidean norm).

The framework of *self-concordant* functions [123, 117] was developed for the affine-invariant characterization of the Newton's Method. However, when the problem class is more specialized, the *cubic regularization* technique provides us with the better complexity guarantees [49].

1.4.3 Cubic Regularization

The Cubic Newton Method can be represented as follows:

$$\boxed{\begin{array}{rcl} x_0 & \in & \operatorname{dom} \psi, \qquad k \geq 0 : \\ \\ x_{k+1} & = & \operatorname*{argmin}\limits_{y} \Bigl\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle \\ & & \quad + \frac{1}{2} \langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \frac{H_k \|y - x_k\|^3}{6} + \psi(y) \Bigr\}. \end{array}} \tag{1.4.9}$$

When $\psi(x) \equiv 0$, the cubic step satisfies the following nonlinear system of equations:

$$\begin{aligned} x_{k+1} &= x_k - \left(\nabla^2 f(x_k) + \tfrac{H_k r_k}{2} B\right)^{-1} \nabla f(x_k), \\ r_k &= \|x_{k+1} - x_k\|. \end{aligned} \tag{1.4.10}$$

In this form, it is similar to the Levenberg-Marquardt regularization [92, 97] approach, which performs the Newton step $x_{k+1} = x_k - G_k^{-1} \nabla f(x_k)$ with a modified Hessian matrix $G_k = \nabla^2 f(x_k) + \gamma_k B \succ 0$.

Another interpretation of equations (1.4.10) is related to the *trust-region idea* [30]: at each step of the method, we restrict the quadratic model onto a neighbourhood of the current point $\Delta(x_k) = \{x \; : \; \|x - x_k\| \leq \epsilon_k\}$ for a certain $\epsilon_k > 0$.

To produce the solution of (1.4.10), let us consider the following auxiliary function,

$$h(r) \;\; \overset{\text{def}}{=} \;\; \|s(r)\| - r, \qquad r > 0,$$

where

$$s(r) \;\; \overset{\text{def}}{=} \;\; \left(\nabla^2 f(x_k) + \tfrac{H_k r}{2} B\right)^{-1} \nabla f(x_k),$$

for some fixed $k \geq 0$. Note that the value of r_k from the method step (1.4.10) solves the equation

$$h(r) \;\; = \;\; 0. \tag{1.4.11}$$

Hence, it can be computed by using a univariate numerical method that finds the root of (1.4.11). Recall that due to convexity, we always have $\nabla^2 f(x_k) \succeq 0$. The derivative of $h(\cdot)$ is equal to

$$h'(r) \;\; = \;\; -\tfrac{H_k}{2\|s(r)\|} \langle Bs(r), (\nabla^2 f(x_k) + \tfrac{H_k r}{2} B)^{-1} Bs(r) \rangle - 1 \;\; < \;\; 0,$$

and thus the function is monotonically decreasing. Moreover, by directly computing the second derivative, we conclude that the function $h(\cdot)$ is convex. Its graph is shown in Figure 1.2.

We can use the standard bisection method to solve (1.4.11). A more efficient strategy is to apply the univariate Newton's Method, one iteration of which is

$$r^+ \;\; = \;\; r - \tfrac{h(r)}{h'(r)}. \tag{1.4.12}$$

If we start the process from the region of positive values of $h(r)$, then it is possible to guarantee the linear rate of convergence with factor $\frac{1}{2}$ for

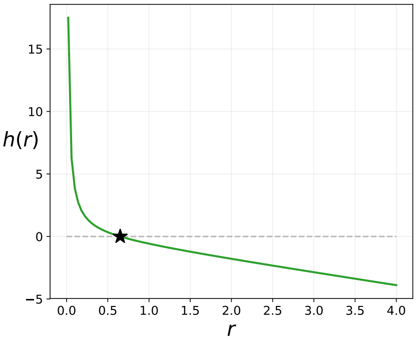


Figure 1.2: The root of the auxiliary univariate function that corresponds to the Cubic Newton step.

method (1.4.12) as applied to (1.4.11) (see Appendix A.1 in [117]). In practice, the convergence of the univariate Newton's Method can be even faster.

It is reasonable to precompute the *eigenvalue* or the *tridiagonal* decomposition of the Hessian in advance, then each step of the univariate method takes only linear time in the problem dimension (which is for solving the corresponding linear systems). Hence, the arithmetical cost of computing the Cubic Newton step is comparable to that one of the classical Newton's. See also [124, 58] for more detailed analysis of the cubic subproblem, and [30] for techniques developed for trust-region methods.

Both factorized-based and matrix-free Lanczos approaches for the cubic subproblem were considered in detail in [21] for the first time. The use of the gradient methods for computing an inexact cubic step was studied in [18, 120]. Randomized versions of the Cubic Newton suitable for solving high-dimensional problems were proposed in [28, 44, 68].

If the Lipschitz constant L_2 is known, we can set $H_k \equiv L_2$. Alternatively, one can use adaptive estimation of the regularization parameter (we study the Adaptive Cubic Newton, algorithm (2.1.22) in Section 2.1 from Chapter 2).

For the class of convex functions with Lipschitz continuous Hessian ($L_2 < +\infty$), the Cubic Newton Method (1.4.9) needs

$$K \quad = \quad \mathcal{O}\Bigl(\sqrt{\tfrac{L_2 D_0^3}{\varepsilon}}\,\Bigr) \tag{1.4.13}$$

iterations to solve the problem up to ε-precision in terms of the functional residual $F(x_K) - F^* \leq \varepsilon$ [124], where

$$D_0 \quad \overset{\text{def}}{=} \quad \sup_x \Big\{ \|x - x^*\| \; : \; F(x) \leq F(x_0) \Big\}.$$

The dependence on ε is much better than that in the global complexity (1.3.12) of the Gradient Method. Estimate (1.4.13) follows from Theorem 2.2.7 in Chapter 2 as a particular case $p = 2$.

1.5 Tensor Methods

Tensor methods is a natural generalization of the gradient and the second-order methods to arbitrary order.

1.5.1 Basic Method

Let us present a basic variant of the regularized composite Tensor Method of a fixed order $p \geq 1$, for convex objective.

$$\boxed{\begin{array}{rcl} x_0 & \in & \mathrm{dom}\,\psi, \qquad k \geq 0 : \\ \\ x_{k+1} & \in & \mathop{\mathrm{Argmin}}\limits_y \Big\{ \Omega_p(f, x_k; y) + \tfrac{H_k \|y - x_k\|^{p+1}}{(p+1)!} + \psi(y) \Big\}, \end{array}} \tag{1.5.1}$$

where Ω_p is the standard Taylor's polynomial of order p, defined by (1.3.4).

For $p = 1$, this is the Gradient Method (1.3.10). When $p = 2$, this is the Cubic Newton (1.4.9).

Unless f is a quadratic function (i.e. its third derivative $D^3 f$ is zero), the third-order Taylor's polynomial $\Omega_3(f, x; y)$ is always *nonconvex* in y. This is easy to see by considering just a one-dimensional polynomial of degree 3; it has a stationary point which is not the global minimum and thus the polynomial is nonconvex (see Figure 1.3).

When the regularization parameter is big enough ($H_k \geq L_p$), it follows from (1.3.5) that the model used in the Tensor Method is a global *upper* approximation of the objective $F(y)$. There is still no evidence that this model is convex, though.

Making H_k a little bigger, it is possible to prove the following important result [118], which is crucial for implementability of method (1.5.1). We include its proof for completeness of our presentation.

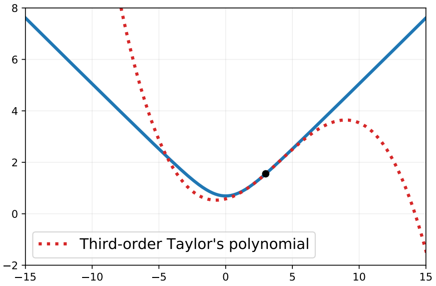


Figure 1.3: Third-order Taylor's approximation of a convex function.

Theorem 1.5.1. *Let*

$$H \quad \geq \quad pL_p. \tag{1.5.2}$$

Then for arbitrary $x \in \operatorname{dom} f$ *the function*

$$g(y) \quad \equiv \quad \Omega_p(f,x;y) + \tfrac{H\|y-x\|^{p+1}}{(p+1)!}$$

is convex.

Proof. We may assume nontrivial case $p \geq 3$. Then, the gradient of g is

$$\nabla g(y) \quad = \quad \nabla \Omega_p(f,x;y) + \tfrac{H\|y-x\|^{p-1}}{p!} B(y-x),$$

and the Hessian is

$$\nabla^2 g(y) \quad = \quad \nabla^2 \Omega_p(f,x;y) + \tfrac{H\|y-x\|^{p-1}}{p!} B + \tfrac{(p-1)H\|y-x\|^{p-3}}{p!} C,$$

where $C : \mathbb{E} \to \mathbb{E}^*$ is a symmetric linear operator, defined by the equation

$$\langle Ch_1, h_2 \rangle \quad = \quad \langle B(y-x), h_1 \rangle \cdot \langle B(y-x), h_2 \rangle, \qquad h_1, h_2 \in \mathbb{E}.$$

In particular, we have $\langle Ch, h \rangle = \langle B(y-x), h \rangle^2 \geq 0$, for all $h \in \mathbb{E}$. Hence,

$$\begin{aligned} \nabla^2 g(y) \quad &\succeq \quad \nabla^2 \Omega_p(f,x;y) + \tfrac{H\|y-x\|^{p-1}}{p!} B \\ &\overset{(1.3.7)}{\succeq} \quad \nabla^2 f(y) + \left(\tfrac{H\|y-x\|^{p-1}}{p!} - \tfrac{L_p\|y-x\|^{p-1}}{(p-1)!} \right) B \\ &\overset{(1.5.2)}{\succeq} \quad \nabla^2 f(y) \quad \succeq \quad 0, \end{aligned}$$

where the last inequality is due to convexity of f. Therefore, g is also convex. □

Due to this theorem, the regularized model is convex for *any* $p \geq 1$ (see Figure 1.4 for $p = 3$). Therefore, we can try to solve the subproblem by using the tools of Linear Algebra and Convex Optimization. For $p = 3$, efficient implementation of the Tensor Step was presented in [118]. We discuss this result in the next section. It remains to be an open problem — how to implement the Tensor Method when $p \geq 4$.

Another interesting open question is how tight bound (1.5.2) is for the regularization parameter, which ensures convexity of the model.

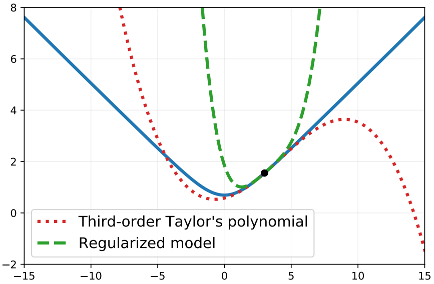


Figure 1.4: Regularization of third-order Taylor's polynomial.

Assuming the p-th derivative of the smooth part is Lipschitz continuous ($L_p < +\infty$), algorithm (1.5.1) needs

$$K \quad = \quad \mathcal{O}\Bigl(\Bigl[\tfrac{L_p D_0^{p+1}}{\varepsilon} \Bigr]^{\frac{1}{p}} \Bigr)$$

iterations to solve the problem up to ε-accuracy: $F(x_K) - F^* \leq \varepsilon$ [6, 118]. We prove this complexity bound in Theorem 2.2.7 from Chapter 2. It is clear that the dramatic improvement in the rate of convergence comes from increasing difficulty in solving the subproblem.

Utilizing the notion of *Estimating Sequences* the rate of high-order Tensor Methods can be accelerated, achieving the complexity [6, 118]:

$$\mathcal{O}\Bigl(\Bigl[\tfrac{L_p D_0^{p+1}}{\varepsilon} \Bigr]^{\frac{1}{p+1}} \Bigr).$$

It can be improved up to the level

$$\mathcal{O}\Big(\Big[\tfrac{L_p D_0^{p+1}}{\varepsilon}\Big]^{\frac{2}{3p+1}}\Big),$$

by using a special line-search in each iteration [101, 54]. The latter rate was shown to be the optimal [4, 118], when $f \in C^{p,p}(\mathbb{E})$.

1.5.2 Implementation by Second-Order Oracle

For $p = 3$, an auxiliary problem in algorithm (1.5.1) at iteration $k \geq 0$ is as follows:

$$\min_{h \in \mathbb{E}} \Big\{ \langle \nabla f(x_k), h \rangle + \tfrac{1}{2}\langle \nabla^2 f(x_k) h, h \rangle + \tfrac{1}{6} D^3 f(x_k)[h]^3 + \tfrac{L_3 \|h\|^4}{4} \Big\},$$

where we assume for simplicity that the composite part is absent: $\psi(x) \equiv 0$, and the regularization parameter is being fixed: $H_k \equiv 6L_3$. Note that this is a bigger value than in the bound (1.5.2); it is needed for analysing the inexact steps.

We present a procedure proposed in [122] for computing the next iterate x_{k+1} of the third-order Tensor Method by solving this auxiliary problem — algorithm (1.5.3).

Every iteration of the procedure is basically the Gradient Step for the auxiliary problem, but with a specific choice of *prox-function* (see [152, 7, 95] for the notion of *relative smoothness*), formed by the second derivative of the initial objective and augmented by the fourth power of the Euclidean norm:

$$d(h) \quad = \quad \tfrac{1}{2}\langle \nabla^2 f(x_k) h, h \rangle + \tfrac{L_3}{4}\|h\|^4,$$

while $d(h) = \frac{1}{2}\|h\|^2$ is used in the standard variant (1.3.10) of the Gradient Method. The arithmetical complexity of such iteration is comparable to that one of the Cubic Newton, and the similar techniques can be used to perform it (see the corresponding discussion in Section 1.4.3).

Additionally, in each step we approximate action of the third derivative $D^3 f(x)[h]^2$ by a finite difference of the gradients:

$$D^3 f(x)[h]^2 \quad \approx \quad \tfrac{1}{\tau^2}\Big[\nabla f(x + \tau h) + \nabla f(x - \tau h) - 2\nabla f(x)\Big],$$

for sufficiently small $\tau > 0$.

Thus we need an access only to the *second-order local oracle* for f. The

Hessian is computed only once and the number of the gradient computations is proportional to the total number of iterations of the process.

Computing Inexact Tensor Step of Order $p = 3$.

Initialization (input: x_k). Choose $\delta > 0$. Set $h_0 = 0 \in \mathbb{E}$,

$\gamma = 2 + \sqrt{2}$, $\tau = \frac{3\delta}{8\gamma \|\nabla f(x_k)\|_*}$, $R = 2\big[\frac{\gamma}{L_3}\|\nabla f(x_k)\|_*\big]^{\frac{1}{3}}$.

Iteration $i \geq 0$.

1: Compute an approximate gradient:

$$\begin{array}{rcl} g_i & = & \nabla f(x_k) + \nabla^2 f(x_k) h_i \\ & & + \frac{1}{2\tau^2}\big[\nabla f(x_k + \tau h_i) + \nabla f(x_k - \tau h_i) - 2\nabla f(x_k)\big]. \end{array} \qquad (1.5.3)$$

2: If $\|g_i\|_* \leq \frac{1}{6}\|\nabla f(x_k + h_i)\|_* - \delta$, then

return $x_{k+1} = x_k + h_i$.

3: Set $\hat{g}_i = g_i - \gamma \nabla^2 f(x_k) h_i - \gamma L_3 \|h_i\|^3 B h_i$.

4: Compute the next iterate:

$$h_{i+1} \;=\; \operatorname*{argmin}_{h : \|h\| \leq R} \Big\{ \langle \hat{g}_i, h\rangle + \frac{\gamma}{2}\langle \nabla^2 f(x_k) h, h\rangle + \frac{\gamma L_3 \|h\|^4}{4} \Big\}.$$

Let us assume

$$f \;\in\; C^{4,1}(\mathbb{E}) \cap C^{4,3}(\mathbb{E}), \qquad (1.5.4)$$

so f is 4-times continuously differentiable and both its first and third derivatives are Lipschitz ($L_1 < +\infty$ and $L_3 < +\infty$). One can show that $L_2 \leq \sqrt{2L_1 L_3}$ (see Lemma 4 in [122]), so it holds:

$$C^{4,1}(\mathbb{E}) \cap C^{4,3}(\mathbb{E}) \;\subseteq\; C^{4,2}(\mathbb{E}) \;\subseteq\; C^{2,2}(\mathbb{E}).$$

Then, for a particular value of the tolerance parameter,

$$\delta \;\approx\; \frac{\varepsilon_g^{3/2}}{\|\nabla f(x_k)\|_*^{1/2} + \|\nabla^2 f(x_k)\|^{3/2} / L_3^{1/2}},$$

where ε_g is a lower bound for the norm of the gradients, algorithm (1.5.3) needs $\mathcal{O}\bigl(\log \frac{1}{\varepsilon_g}\bigr)$ iterations to return x_{k+1}, which is a good approximation of the third-order Tensor Step [122], that provides us with the global convergence guarantees. Hence, we obtain formally the *second-order* schemes with complexities

$$K = \tilde{\mathcal{O}}\bigl(\varepsilon^{-\frac{1}{3}}\bigr) \qquad \text{and} \qquad K = \tilde{\mathcal{O}}\bigl(\varepsilon^{-\frac{1}{4}}\bigr)$$

oracle calls to find a point x_K s.t. $f(x_K) - f^* \leq \varepsilon$, for the basic and for the accelerated Tensor Methods respectively. Later on, the complexity of optimization methods on the functional class (1.5.4) was improved up to

$$K = \tilde{\mathcal{O}}\bigl(\varepsilon^{-\frac{1}{5}}\bigr)$$

second-order oracle calls [121, 76]. This is better than the known lower bound $\Omega\bigl(\varepsilon^{-\frac{2}{7}}\bigr)$ obtained for the second-order methods on $f \in C^{2,2}(\mathbb{E})$ [4]. Of course, such an acceleration is possible due to a finer specification (1.5.4) of the problem class.

1.6 Arithmetical Complexity of Oracles

Let us discuss a relationship between analytical and arithmetical complexities of optimization methods.

In this thesis, we are mainly interested in *analytical* complexity, originated in [102]. This is a total number of *oracle calls* for an optimization method, that is required to solve arbitrary problem from a given problem class. The first-order oracle returns the function value and the gradient computed at the given point,

$$O_1 : \quad x \quad \mapsto \quad (f(x), \nabla f(x)),$$

while the second-order oracle reveals additional information, which is the Hessian of the objective,

$$O_2 : \quad x \quad \mapsto \quad (f(x), \nabla f(x), \nabla^2 f(x)).$$

Analogously, one can define the local oracle of degree $p \geq 1$ that returns all the derivatives up to the fixed order,

$$O_p : \quad x \quad \mapsto \quad (f(x), \ldots, D^p f(x)).$$

For the most iterative methods that we consider, the number of iterations is always proportional to the number of oracle calls. Hence, we can usually estimate *arithmetical* complexity as a product of the analytical complexity and the cost of each iteration. The cost of one iteration consists of the number of arithmetical operations required to implement the oracle call, and some possible additional operations to solve an auxiliary problem. Thus, we come to the following intuitive formula:

$$\begin{aligned}\texttt{Arithmetical Complexity} \quad &= \quad \texttt{Analytical Complexity} \\ &\quad\ * \big(\texttt{Oracle Call} + \texttt{Auxiliary Computations}\big).\end{aligned}$$

It is clear that for the oracles of different order, the cost of their call may be quite different. At the same time, it strictly depends on the target objective and the way we represent it. Let us consider some typical examples.

1. *Separable Optimization.* In applications related to Machine Learning and Statistics [53, 146], very often we have the following structural representation of the objective:

   $$f(x) \quad = \quad \tfrac{1}{M}\sum_{i=1}^{M} \phi(\langle a_i, x\rangle), \qquad x \in \mathbb{R}^n,$$

   where $a_i \in \mathbb{R}^n, 1 \le i \le M$ are given data vectors, and $\phi : \mathbb{R} \to \mathbb{R}$ is a fixed convex loss function. In this case, we have

   $$\nabla f(x) \quad = \quad A^T s(x),$$

   where $A \in \mathbb{R}^{M\times n}$ is the matrix whose rows are formed by vectors $a_1, \dots, a_M$, and $s(x) \in \mathbb{R}^M$ is a vector,

   $$\big[s(x)\big]^{(i)} \quad \overset{\text{def}}{=} \quad \tfrac{1}{M}\phi'(\langle a_i, x\rangle), \qquad 1 \le i \le M.$$

   Assume that ϕ and its derivatives are computable in $\mathcal{O}(1)$ operations. Then the most difficult part is the computation of matrix-vector products Ax and $A^T s(x)$, for the given x, which requires $\mathcal{O}(Mn)$ arithmetical operations in general.

   For the Hessian matrix, we have

   $$\nabla^2 f(x) \quad = \quad A^T d(x) A \quad \in \quad \mathbb{R}^{n\times n},$$

where $d(x) \in \mathbb{R}^{M\times M}$ is a diagonal matrix,

$$\big[d(x)\big]^{(i,i)} \quad \overset{\text{def}}{=} \quad \tfrac{1}{M}\phi''(\langle a_i, x\rangle), \qquad 1 \le i \le M.$$

Hence, the second-order oracle call already costs $\mathcal{O}(Mn^2)$ arithmetical operations, if using a trivial matrix multiplication algorithm.

For any $p \ge 1$ and for $h_1, \dots, h_p \in \mathbb{R}^n$, we have

$$D^p f(x)[h_1, \dots, h_p] \quad = \quad \tfrac{1}{M}\sum_{i=1}^{M} \phi^{(p)}(\langle a_i, x\rangle) \prod_{j=1}^{p} \langle a_i, h_j\rangle.$$

The computation of the directional derivative of order $p \ge 1$ along some fixed directions requires only $\mathcal{O}(pMn)$ arithmetical operations, while it costs $\mathcal{O}(pMn^p)$ operations and $\mathcal{O}(n^p)$ amount of memory to compute and keep the whole tensor, which is enormous for big n and p. Thus it is important that we can implement third-order tensor methods by using *only* second-order oracle calls (see Section 1.5.2).

2. *Log-Sum-Exp.* Let us consider the function from Example 1.3.5:

$$f(x) \quad = \quad \log\Bigg(\sum_{i=1}^{m} e^{\langle a_i, x\rangle}\Bigg), \qquad x \in \mathbb{R}^n,$$

where $a_i \in \mathbb{R}^n$, $1 \le i \le m$ are given vectors. Denoting by $A \in \mathbb{R}^{m\times n}$ the matrix whose rows are formed by vectors $a_1, \dots, a_m$, and using an auxiliary vector $\pi(x) \in \mathbb{R}^m$,

$$\big[\pi(x)\big]^{(i)} \quad \overset{\text{def}}{=} \quad e^{\langle a_i, x\rangle} \cdot \Bigg(\sum_{j=1}^{m} e^{\langle a_j, x\rangle}\Bigg)^{-1}, \qquad 1 \le i \le m,$$

we have the following expression for the gradient:

$$\nabla f(x) \quad = \quad A^T \pi(x),$$

and for the Hessian:

$$\nabla^2 f(x) \quad = \quad A^T\big(\operatorname{diag}(\pi(x)) - \pi(x)\pi(x)^T\big)A.$$

It costs $\mathcal{O}(mn)$ and $\mathcal{O}(mn^2)$ arithmetical operations to call the first-order and second-order oracles, respectively.

Note that in the case of data sparsity, these complexity estimates can be improved by working only with nonzero elements.

3. *Approximations by finite differences.* Finally, let us discuss a general approach for computation of the Hessian, that is based on the Taylor formula. We have already seen this technique in Section 1.5.2, where the action of the third derivative was approximated by the gradients.

   For a fixed point $x \in \mathbb{E}$ and arbitrary direction $h \in \mathbb{E}$, we know the following approximation of the Hessian-vector product:

   $$\nabla^2 f(x) h \quad \approx \quad \tfrac{1}{\tau}\big[\nabla f(x + \tau h) - \nabla f(x)\big],$$

   where $\tau > 0$ is a small parameter. Hence, we can approximate the action of the Hessian to arbitrary vector by using only *two* gradient computations.

   Now, let us assume that $\mathbb{E} = \mathbb{R}^n$ and $e_1, \ldots, e_n \in \mathbb{R}^n$ is the standard basis. Then we can build the matrix $A_{x,\tau} \in \mathbb{R}^{n \times n}$ whose rows are the approximations of $\nabla^2 f(x) e_i$, $1 \leq i \leq n$, i.e.

   $$A_{x,\tau}^{(i,j)} \quad \overset{\text{def}}{=} \quad \tfrac{1}{\tau}\big[\nabla f(x + \tau e_i) - \nabla f(x)\big]^{(j)}, \qquad 1 \leq i, j \leq n.$$

   After symmetrization, we get a symmetric approximation to the Hessian matrix:

   $$\nabla^2 f(x) \quad \approx \quad \tfrac{1}{2}\big(A_{x,\tau} + A_{x,\tau}^T\big).$$

   Forming this matrix requires $n+1$ gradient computations. Therefore, the second-order oracle can be implemented by $n+1$ calls of the first-order one. However, this approach works only if the target accuracy for our problem is not very high, due to the limits of machine precision.

   It is clear that we may also obtain approximations of the gradients and the Hessians by using only the function values, which results in *zeroth-order* or *derivative-free* methods.

   Finite differencing is discussed in more details in [127]. Its applications to the cubically regularized Newton method were considered in [24].

Chapter 2

Minimizing Uniformly Convex Functions

Let us start with the definition of uniform convexity. Then we list some basic properties related to this notion.

We say that function F is *uniformly convex* of degree $q \geq 2$ on a convex set $Q \subseteq \operatorname{dom} F$ if for some constant $\sigma > 0$ it satisfies inequality

$$F(y) \quad \geq \quad F(x) + \langle F'(x), y - x \rangle + \tfrac{\sigma \| y-x \|^q}{q}, \tag{2.0.1}$$

for all $x, y \in Q$ and for all subgradients $F'(x) \in \partial F(x)$. Uniformly convex functions of degree $q = 2$ are known as strongly convex.

The following convenient condition is sufficient for function F to be uniformly convex on a convex set Q.

Lemma 2.0.1. *If for all $x, y \in Q$ and for all $F'(x) \in \partial F(x)$, $F'(y) \in \partial F(y)$ it holds*

$$\langle F'(x) - F'(y), x - y \rangle \quad \geq \quad \sigma \| y - x \|^q, \tag{2.0.2}$$

then function F is uniformly convex of degree q on set Q with parameter σ.

Proof. Indeed, for a particular selection of subgradients, we have by the

Newton-Leibniz formula:

$$\begin{aligned} & F(y) - F(x) - \langle F'(x), y - x \rangle \\ = \quad & \int\limits_0^1 \langle F'(x + \tau(y-x)) - F'(x), y - x \rangle d\tau \\ \overset{(2.0.2)}{\geq} \quad & \int\limits_0^1 \sigma \tau^{q-1} \|y - x\|^q d\tau \quad = \quad \tfrac{\sigma \|y-x\|^q}{q}. \end{aligned}$$

□

The main source of uniformly convex functions for us consists in taking a power of the Euclidean norm (see Example 2.1.4 in Section 2.1.1).

From now on, considering the composite optimization problem:

$$F^* \quad \overset{\mathrm{def}}{=} \quad \min_x \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\},$$

we set $Q := \mathrm{dom}\, \psi \subseteq \mathrm{dom}\, f$. The optimum x^* always exists for F being uniformly convex and closed. From (2.0.1) it follows that the optimum is unique, since $0 \in \partial F(x^*)$.

A useful consequence of the uniform convexity is the following upper bound for the functional residual.

Lemma 2.0.2. *For every $x \in \mathrm{dom}\, \psi$ and for all $F'(x) \in \partial F(x)$ it holds*

$$F(x) - F^* \quad \leq \quad \tfrac{q-1}{q} \left(\tfrac{1}{\sigma} \right)^{\frac{1}{q-1}} \|F'(x)\|_*^{\frac{q}{q-1}}. \tag{2.0.3}$$

Proof. Let us minimize the left- and the right-hand sides of (2.0.1) with respect to y independently:

$$\begin{aligned} F^* \quad & = \quad \min_{y \in \mathrm{dom}\, \psi} F(y) \quad \overset{(2.0.1)}{\geq} \quad \min_{y \in \mathbb{E}} \Big\{ F(x) + \langle F'(x), y - x \rangle + \tfrac{\sigma \|y-x\|^q}{q} \Big\} \\ & = \quad F(x) - \tfrac{q-1}{q} \left(\tfrac{1}{\sigma} \right)^{\frac{1}{q-1}} \|F'(x)\|_*^{\frac{q}{q-1}}, \end{aligned}$$

and this is (2.0.3). □

2.1 Second-Order Global Nondegeneracy

We have seen that for general convex functions with Lipschitz continuous Hessian, the Cubic Newton has a better rate of convergence than that of the Gradient Method for convex functions with Lipschitz continuous gradient (see the dependence on ε in the complexity estimates (1.4.13) and (1.3.12), respectively). At the same time, it is known that the first-order methods achieve the fast *linear rate*, when the objective is *strongly* convex. The corresponding complexity of the Gradient Method is (1.3.13).

Despite a number of nice properties, global complexity bounds of the cubically regularized Newton Method for the cases of strongly convex and uniformly convex objective are not still fully investigated, as well as the notion of second-order non-degeneracy (see also the discussion in Section 5 in [111]). We are going to address this question in the current part of the thesis.

In Section 2.1.1 we consider the class of twice-differentiable uniformly convex functions with Hölder continuous Hessian. We introduce the notion of the *condition number* ω_ν of a certain degree $\nu \in [0, 1]$ and present some basic examples.

In Section 2.1.2 we describe a general regularized Newton scheme and show the linear rate of convergence for this method on the class of uniformly convex functions with a known degree $\nu \in [0, 1]$ of nondegeneracy. Then we introduce the adaptive cubically regularized Newton method and collect useful inequalities and properties, which are related to this algorithm.

In Section 2.1.3 we study global iteration complexity of the cubically regularized Newton method on the classes of uniformly convex functions with Hölder continuous Hessian. We show that for nondegeneracy of *any* degree $\nu \in [0, 1]$, which is formalized by the condition $\omega_\nu < +\infty$, the algorithm automatically achieves the linear rate of convergence with the value ω_ν being the main complexity factor.

Finally, in Section 2.1.4 we compare our complexity bounds with the known bounds for other methods and discuss the results. In particular, we justify an intuitively plausible (but quite a delayed) result that the global complexity of the cubically regularized Newton method is always better than that of the Gradient Method on the class of strongly convex functions with uniformly bounded second derivative.

2.1.1 Uniformly Convex Functions with Hölder Continuous Hessian

Let us assume that the smooth part $f(\cdot)$ of the composite problem,

$$\min_x \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\},$$

is uniformly convex. It is reasonable to define the best possible constant of uniform convexity σ in inequality (2.0.2) for a certain degree q. This leads us to a system of constants:

$$\sigma_q \quad \overset{\text{def}}{=} \quad \inf_{\substack{x,y \,\in\, \mathrm{dom}\,\psi \\ x \neq y}} \tfrac{\langle \nabla f(x) - \nabla f(y), x - y \rangle}{\|x-y\|^q}, \qquad q \geq 2. \tag{2.1.1}$$

We prefer to use inequality (2.0.2) for the definition of σ_q, instead of (2.0.1), because of its symmetry in x and y. Note that the value σ_q depends on the domain of ψ. However, we omit this dependence in our notation since it is always clear from the context.

It is easy to see that σ_q as a univariate function in q is log-concave. Thus, for all $q_2 > q_1 \geq 2$ we have:

$$\sigma_q \quad \geq \quad \big(\sigma_{q_1}\big)^{\frac{q_2 - q}{q_2 - q_1}} \cdot \big(\sigma_{q_2}\big)^{\frac{q - q_1}{q_2 - q_1}}, \qquad q \in [q_1, q_2]. \tag{2.1.2}$$

For a twice-differentiable function f, we say that it has *Hölder continuous Hessian* of degree $\nu \in [0,1]$ on a convex set $Q \subseteq \mathrm{dom}\, f$, if for some constant $\mathcal{H} \geq 0$, it holds:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \quad \leq \quad \mathcal{H}\|x-y\|^{\nu}, \qquad \forall x, y \in Q. \tag{2.1.3}$$

Two simple consequences of (2.1.3) are as follows:

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\|_* \quad \leq \quad \tfrac{\mathcal{H}\|x-y\|^{1+\nu}}{1+\nu}, \tag{2.1.4}$$

$$|f(y) - \Omega_2(f, x; y)| \quad \leq \quad \tfrac{\mathcal{H}\|x-y\|^{2+\nu}}{(1+\nu)(2+\nu)}, \tag{2.1.5}$$

where $\Omega_2(f, x; y)$ is the quadratic model of f at the point x:

$$\Omega_2(f, x; y) \quad \overset{\text{def}}{=} \quad f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(y-x), y-x \rangle.$$

In order to characterize the level of smoothness of function f on the set

$Q := \mathrm{dom}\,\psi$, let us define the system of Hölder constants (see [60]):

$$\mathcal{H}_\nu \overset{\mathrm{def}}{=} \sup_{\substack{x,y \in \mathrm{dom}\,\psi \\ x \neq y}} \frac{\|\nabla^2 f(x) - \nabla^2 f(y)\|}{\|x-y\|^\nu}, \qquad \nu \in [0,1]. \tag{2.1.6}$$

We allow $\mathcal{H}_\nu$ to be equal to $+\infty$ for some ν. Note that $\mathcal{H}_\nu$ as a function in ν is log-convex. Thus, any $0 \leq \nu_1 < \nu_2 \leq 1$ such that $\mathcal{H}_{\nu_i} < +\infty, i = 1, 2$, provide us with the following upper bounds for the whole interval:

$$\mathcal{H}_\nu \leq \big(\mathcal{H}_{\nu_1}\big)^{\frac{\nu_2-\nu}{\nu_2-\nu_1}} \cdot \big(\mathcal{H}_{\nu_2}\big)^{\frac{\nu-\nu_1}{\nu_2-\nu_1}}, \qquad \nu \in [\nu_1, \nu_2]. \tag{2.1.7}$$

If for some specific $\nu \in [0,1]$ we have $\mathcal{H}_\nu = 0$, this implies that $\nabla^2 f(x) = \nabla^2 f(y)$ for all $x, y \in \mathrm{dom}\,\psi$. In this case restriction $f|_{\mathrm{dom}\,\psi}$ is a quadratic function and we conclude that $\mathcal{H}_\nu = 0$ for *all* $\nu \in [0,1]$. At the same time, having two points $x, y \in \mathrm{dom}\,\psi$ with $0 < \|x-y\| \leq 1$, we get a simple uniform lower bound for all constants $\mathcal{H}_\nu$:

$$\mathcal{H}_\nu \geq \|\nabla^2 f(x) - \nabla^2 f(y)\|, \qquad \nu \in [0,1].$$

Let us give an example of a function, that has a Hölder continuous Hessian for all $\nu \in [0,1]$.

Example 2.1.1. For a given $a_i \in \mathbb{E}^*$, $1 \leq i \leq m$, consider the function:

$$f(x) = \log\left(\sum_{i=1}^m e^{\langle a_i, x\rangle}\right), \qquad x \in \mathbb{E},$$

and fix the Euclidean norm $\|x\| = \langle Bx, x\rangle^{1/2}$, with operator $B := \sum_{i=1}^m a_i a_i^*$. Without loss of generality, we assume that $B \succ 0$. From Example 1.3.5, we know that the corresponding Lipschitz constants are: $L_1 = 1$, and $L_2 = 2$.

Hence,

$$\mathcal{H}_0 \leq 1, \qquad \mathcal{H}_1 \leq 2.$$

Therefore, by (2.1.7) we get, for any $\nu \in [0,1]$

$$\mathcal{H}_\nu \leq 2^\nu.$$

□

Let us imagine now that we want to describe the iteration complexity of some method, which solves the composite optimization problem up to an absolute accuracy $\varepsilon > 0$ in the function value. We assume that the smooth part f of the objective is uniformly convex and has a Hölder continuous Hessian. Which degrees q and ν should be used in our analysis? Suppose that, for the number of *calls of the oracle*, we are interested in obtaining a polynomial-time bound of the form:

$$\mathcal{O}\left((\mathcal{H}_\nu)^\alpha \cdot (\sigma_q)^\beta \cdot \log \tfrac{F(x_0)-F^*}{\varepsilon} \right), \quad \alpha, \beta \neq 0.$$

Denote by $[x]$ an intuitive *physical dimension* of variable $x \in \mathbb{E}$, and by $[f]$ the *physical dimension* of the value $f(x)$. Then, we have $[\nabla f(x)] = [f]/[x]$ and $[\nabla^2 f(x)] = [f]/[x]^2$. This gives us

$$[\mathcal{H}_\nu] \;=\; \tfrac{[f]}{[x]^{2+\nu}}, \quad [\sigma_q] = \tfrac{[f]}{[x]^q}, \quad [\,(\mathcal{H}_\nu)^\alpha \cdot (\sigma_q)^\beta\,] = \tfrac{[f]^{\alpha+\beta}}{[x]^{\alpha(2+\nu)+\beta q}}.$$

While x and $f(x)$ can be measured in arbitrary physical quantities, the value "number of iterations" *cannot have* physical dimension. This leads to the following relations:

$$\alpha + \beta = 0 \qquad \text{and} \qquad \alpha(2+\nu) + \beta q = 0.$$

Therefore, despite to the fact that our function can belong to several problem classes simultaneously, from the physical point of view only one option is available:

$$\boxed{q = 2 + \nu}$$

Hence, for a twice-differentiable convex function f with

$$\inf_{\nu \in [0,1]} \mathcal{H}_\nu \;\; > \;\; 0,$$

we can define only one meaningful *condition number* of degree $\nu \in [0,1]$:

$$\boxed{\omega_\nu \;\; \overset{\text{def}}{=} \;\; \frac{\mathcal{H}_\nu}{\sigma_{2+\nu}}} \tag{2.1.8}$$

If for some particular ν we have $\sigma_{2+\nu} \; = \; 0$ or $\mathcal{H}_\nu \; = \; +\infty$ then by our definition: $\omega_\nu \overset{\text{def}}{=} +\infty$.

It will be shown that the condition number ω_ν serves as a main factor in the global iteration complexity bounds for the regularized Newton method as applied to the composite problem. Let us prove that this number is bounded from below.

Lemma 2.1.2. *Let* $\inf_{\nu \in [0,1]} \mathcal{H}_\nu > 0$ *and therefore the condition number be* *well defined. Then,*

$$\omega_\nu \quad \geq \quad \left(\frac{1}{1+\nu} \;+\; \inf_{\substack{x,y \in \mathrm{dom}\,\psi \\ x \neq y}} \frac{\|\nabla^2 f(x)\|}{\|\nabla^2 f(y) - \nabla^2 f(x)\|} \right)^{-1}, \quad \forall \nu \in [0,1]. \tag{2.1.9}$$

In the case when $\mathrm{dom}\,\psi$ *is unbounded:* $\sup_{x \in \mathrm{dom}\,\psi} \|x\| = +\infty$, *we have*

$$\omega_\nu \quad \geq \quad 1 + \nu, \qquad \forall \nu \in (0,1]. \tag{2.1.10}$$

Proof. Indeed, for any $x, y \in \mathrm{dom}\,\psi$, $x \neq y$, we have:

$$\begin{array}{rcl} \sigma_{2+\nu} & \overset{(2.1.1)}{\leq} & \frac{\langle \nabla f(y) - \nabla f(x), y - x \rangle}{\|y-x\|^{2+\nu}} \\ & = & \frac{\langle \nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x), y - x \rangle}{\|y-x\|^{2+\nu}} \;+\; \frac{\langle \nabla^2 f(x)(y-x), y-x \rangle}{\|y-x\|^{2+\nu}} \\ & \overset{(2.1.4)}{\leq} & \frac{\mathcal{H}_\nu}{1+\nu} \;+\; \frac{\|\nabla^2 f(x)\|}{\|y-x\|^\nu}. \end{array}$$

Now, dividing both sides of this inequality by $\sigma_{2+\nu}$ (we assume it is positive, the other case is trivial), we get inequality (2.1.9) from the definition of $\mathcal{H}_\nu$ (2.1.6). Inequality (2.1.10) can be obtained by taking the limit $\|y\| \to +\infty$. □

From inequalities (2.1.2) and (2.1.7) we can get the following upper bound:

$$\omega_\nu \quad \leq \quad \left(\omega_{\nu_1}\right)^{\frac{\nu_2 - \nu}{\nu_2 - \nu_1}} \cdot \left(\omega_{\nu_2}\right)^{\frac{\nu - \nu_1}{\nu_2 - \nu_1}}, \qquad \forall \nu \in [\nu_1, \nu_2],$$

where $0 \leq \nu_1 < \nu_2 \leq 1$. However, it turns out that in *unbounded case* we can have a meaningful condition number ω_ν only for a *single degree.*

Lemma 2.1.3. *Let* $\mathrm{dom}\,\psi$ *be unbounded:* $\sup_{x \in \mathrm{dom}\,\psi} \|x\| = +\infty$. *Assume that for a fixed* $\nu \in [0,1]$ *we have* $\omega_\nu < +\infty$. *Then,*

$$\omega_\alpha = +\infty \quad \textit{for all} \quad \alpha \in [0,1] \setminus \{\nu\}.$$

Proof. Consider firstly the case: $\alpha > \nu$. From the condition $\omega_\nu < +\infty$ we conclude that $\mathcal{H}_\nu < +\infty$. Then, for any $x, y \in \mathrm{dom}\,\psi$ we have:

$$\begin{array}{rcl} \frac{\sigma_{2+\alpha}\|y-x\|^{2+\alpha}}{2+\alpha} & \leq & f(y) - f(x) - \langle \nabla f(x), y - x \rangle \\ & \overset{(2.1.5)}{\leq} & \frac{1}{2}\langle \nabla^2 f(x)(y-x), (y-x) \rangle + \frac{\mathcal{H}_\nu \|y-x\|^{2+\nu}}{(1+\nu)(2+\nu)}. \end{array}$$

Dividing both sides of this inequality by $\|y-x\|^{2+\alpha}$ and letting $\|x\| \to +\infty$, we get $\sigma_{2+\nu} = 0$. Therefore, $\omega_\alpha = +\infty$. For the second case, $\alpha < \nu$, we cannot have $\omega_\alpha < +\infty$, since the previous reasoning results in $\omega_\nu = +\infty$. □

Let us look now at an important example of a uniformly convex function with a Hölder continuous Hessian. It is convenient to start with some properties of powers of Euclidean norm.

Lemma 2.1.4. *For fixed real $p \geq 1$, consider the following function:*

$$f_p(x) \quad = \quad \tfrac{1}{p}\|x\|^p, \quad x \in \mathbb{E}.$$

1. For $p \geq 2$, function $f_p(\cdot)$ is uniformly convex of degree p:[1]

$$\langle \nabla f_p(x) - \nabla f_p(y), x-y \rangle \quad \geq \quad 2^{2-p}\|x-y\|^p, \qquad \forall x,y \in \mathbb{E}. \quad (2.1.11)$$

2. If $1 \leq p \leq 2$, then function $f_p(\cdot)$ has a ν-Hölder continuous gradient with $\nu = p-1$:

$$\|\nabla f_p(x) - \nabla f_p(y)\|_* \leq 2^{1-\nu}\|x-y\|^\nu, \qquad \forall x,y \in \mathbb{E}. \quad (2.1.12)$$

Proof. Firstly, recall two useful inequalities, which are valid for all $a, b \geq 0$:

$$|a^\alpha - b^\alpha| \;\leq\; |a-b|^\alpha, \quad \text{when} \quad 0 \leq \alpha \leq 1, \quad (2.1.13)$$

$$|a^\alpha - b^\alpha| \;\geq\; |a-b|^\alpha, \quad \text{when} \quad \alpha \geq 1. \quad (2.1.14)$$

Let us fix arbitrary $x, y \in \mathbb{E}$. The left hand side of inequality (2.1.11) equals

$$\langle \|x\|^{p-2}Bx - \|y\|^{p-2}By, x-y \rangle \;=\; \|x\|^p + \|y\|^p - \langle Bx, y\rangle(\|x\|^{p-2} + \|y\|^{p-2}),$$

and we need to verify that it is bigger than $2^{2-p}\big[\|x\|^2 + \|y\|^2 - 2\langle Bx,y\rangle\big]^{\frac{p}{2}}$. The case $x = 0$ or $y = 0$ is trivial. Therefore, assume $x \neq 0$ and $y \neq 0$. Denoting $\tau := \frac{\|y\|}{\|x\|}$, $r := \frac{\langle Bx,y\rangle}{\|x\|\cdot\|y\|}$, we have the following statement to prove:

$$1 + \tau^p \quad \geq \quad r\tau(1+\tau^{p-2}) + 2^{2-p}\big[1+\tau^2 - 2r\tau\big]^{\frac{p}{2}}, \quad \tau > 0, \quad |r| \leq 1.$$

Since the function in the right-hand side is convex in r, we need to check only two marginal cases:

[1] For the integer values of p, this inequality was proved in [111].

1. $r = 1$: $\quad 1+\tau^p \;\geq\; \tau(1+\tau^{p-2}) + 2^{2-p}|1-\tau|^p$, which is equivalent to $(1-\tau)(1-\tau^{p-1}) \geq 2^{2-p}|1-\tau|^p$. This is true by (2.1.14).
2. $r = -1$: $\quad 1+\tau^p \;\geq\; -\tau(1+\tau^{p-2}) + 2^{2-p}(1+\tau)^p$, which is equivalent to $(1+\tau^{p-1}) \geq 2^{2-p}(1+\tau)^{p-1}$. This is true in view of the convexity of function τ^{p-1} for $\tau \geq 0$.

Thus we have proved (2.1.11). Let us prove the second statement. Consider the function $\hat{f}_q(s) = \frac{1}{q}\|s\|_*^q$, $s \in \mathbb{E}^*$, with $q = \frac{p}{p-1} \geq 2$. In view of our first statement, we have, for all $s_1, s_2 \in \mathbb{E}^*$

$$\langle s_1 - s_2, \nabla \hat{f}_q(s_1) - \nabla \hat{f}_q(s_2)\rangle \quad \geq \quad \left(\tfrac{1}{2}\right)^{q-2} \|s_1 - s_2\|_*^q. \tag{2.1.15}$$

For arbitrary $x_1, x_2 \in \mathbb{E}$, define $s_i = \nabla f_p(x_i) = \frac{Bx_i}{\|x_i\|^{2-p}}$, $i = 1, 2$. Then $\|s_i\|_* = \|x_i\|^{p-1}$, and consequently,

$$x_i \quad = \quad \|x_i\|^{2-p} B^{-1} s_i \;=\; \|s_i\|_*^{\frac{2-p}{p-1}} B^{-1} s_i \;=\; \nabla \hat{f}_q(s_i).$$

Therefore, by substituting these vectors in (2.1.15), we get,

$$\left(\tfrac{1}{2}\right)^{q-2} \|\nabla f_p(x_1) - \nabla f_p(x_2)\|_*^q \leq \langle \nabla f_p(x_1) - \nabla f_p(x_2), x_1 - x_2\rangle.$$

Thus, $\|\nabla f_p(x_1) - \nabla f_p(x_2)\|_* \leq 2^{\frac{q-2}{q-1}} \|x_1 - x_2\|^{\frac{1}{q-1}}$. It remains to note that $\frac{1}{q-1} = p - 1 = \nu$. □

Example 2.1.5. For real $q \geq 2$ and an arbitrary $x_0 \in \mathbb{E}$, consider the following function:

$$f(x) \quad = \quad \tfrac{1}{q}\|x - x_0\|^q \;=\; f_q(x - x_0), \quad x \in \mathbb{E}.$$

Then $\sigma_q \;=\; \left(\tfrac{1}{2}\right)^{q-2}$. Moreover, if $q = 2 + \nu$ for some $\nu \in (0,1]$, then it holds,

$$\mathcal{H}_\nu \quad \leq \quad (1+\nu)2^{1-\nu},$$

and $\mathcal{H}_\alpha \;=\; +\infty$, for all $\alpha \in [0,1] \setminus \{\nu\}$. Therefore, in this case we have $\omega_\nu \;\leq\; 2(1+\nu)$, and $\omega_\alpha = +\infty$ for all $\alpha \in [0,1] \setminus \{\nu\}$.

Proof. Let us take an arbitrary $x \neq 0$ and set $y := -x$. Then,

$$\langle \nabla f(x) - \nabla f(y), y - x\rangle \;=\; \langle \|x\|^{q-2}Bx + \|x\|^{q-2}Bx, 2x\rangle \;=\; 4\|x\|^q.$$

On the other hand, $\|y-x\|^q = 2^q\|x\|^q$. Therefore, $\sigma_q \overset{(2.1.1)}{\leq} 2^{2-q}$, and (2.1.11) tells us that this inequality is satisfied as equality.

Let us prove now that $\mathcal{H}_\nu \leq (1+\nu)2^{1-\nu}$ for $q = 2+\nu$ with some $\nu \in (0,1]$. This is

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \;\leq\; (1+\nu)2^{1-\nu}\|x-y\|^\nu, \qquad \forall x,y \in \mathbb{E}. \qquad (2.1.16)$$

The corresponding Hessians can be represented as follows:

$$\nabla^2 f(x) \;\;=\;\; \|x\|^\nu B + \tfrac{\nu B x x^* B}{\|x\|^{2-\nu}}, \quad x \in \mathbb{E}\setminus\{0\}, \quad \nabla^2 f(0) = 0.$$

For the case $x=y=0$, inequality (2.1.16) is trivial. Assume now that $x \neq 0$. If $0 \in [x,y]$, then $y = -\beta x$ for some $\beta \geq 0$ and we have:

$$\begin{aligned} \|\nabla^2 f(x) - \nabla^2 f(-\beta x)\| &\leq |1-\beta^\nu|(1+\nu)\|x\|^\nu \\ &\leq (1+\beta)^\nu(1+\nu)2^{1-\nu}\|x\|^\nu \\ &= (1+\nu)2^{1-\nu}\|x-y\|^\nu, \end{aligned}$$

which is (2.1.16). Let $0 \notin [x,y]$. For an arbitrary fixed direction $h \in \mathbb{E}$, we get:

$$\begin{aligned} &\left|\left\langle(\nabla^2 f(x) - \nabla^2 f(y))h, h\right\rangle\right| \\ &= \left|(\|x\|^\nu - \|y\|^\nu)\cdot\|h\|^2 + \nu\cdot\left(\tfrac{\langle Bx,h\rangle^2}{\|x\|^{2-\nu}} - \tfrac{\langle By,h\rangle^2}{\|y\|^{2-\nu}}\right)\right|. \end{aligned}$$

Consider the points $u = \frac{Bx}{\|x\|^{1-\nu}} = \nabla f_s(x)$ and $v = \frac{By}{\|y\|^{1-\nu}} = \nabla f_s(y)$ with $s := 1+\nu$. Then,

$$\|x\|^\nu = \|u\|_*, \quad \tfrac{\langle Bx,h\rangle^2}{\|x\|^{2-\nu}} = \tfrac{\langle u,h\rangle^2}{\|u\|_*} \quad \text{and} \quad \|y\|^\nu = \|v\|_*, \quad \tfrac{\langle By,h\rangle^2}{\|y\|^{2-\nu}} = \tfrac{\langle v,h\rangle^2}{\|v\|_*}.$$

Therefore,

$$\begin{aligned} &\left|\left\langle(\nabla^2 f(x) - \nabla^2 f(y))h, h\right\rangle\right| \\ &= \left|(\|u\|_* - \|v\|_*)\cdot\|h\|^2 \;+\; \nu\cdot\left(\tfrac{\langle u,h\rangle^2}{\|u\|_*} - \tfrac{\langle v,h\rangle^2}{\|v\|_*}\right)\right|. \end{aligned} \qquad (2.1.17)$$

Let us estimate the right-hand side of (2.1.17) from above. Consider a continuously differentiable univariate function:

$$\phi(\tau) \;:=\; \|u(\tau)\|_*\cdot\|h\|^2 + \nu\cdot\tfrac{\langle u(\tau),h\rangle^2}{\|u(\tau)\|_*}, \quad u(\tau) \;:=\; u + \tau(v-u), \quad \tau \in [0,1].$$

Note that

$$\begin{array}{rcl}\phi'(\tau) & = & \frac{\langle u(\tau), B^{-1}(v-u)\rangle}{\|u(\tau)\|_*} \cdot \|h\|^2 + \frac{2\nu \langle u(\tau),h\rangle\langle v-u,h\rangle}{\|u(\tau)\|_*} \\ & & - \frac{\nu\langle u(\tau),h\rangle^2 \langle u(\tau),B^{-1}(v-u)\rangle}{\|u(\tau)\|_*^3} \\ & = & \frac{\langle u(\tau), B^{-1}(v-u)\rangle}{\|u(\tau)\|_*} \cdot \underbrace{\left(\|h\|^2 - \frac{\nu\langle u(\tau),h\rangle^2}{\|u(\tau)\|_*^2}\right)}_{\geq 0} + \frac{2\nu \langle u(\tau),h\rangle\langle v-u,h\rangle}{\|u(\tau)\|_*}.\end{array}$$

Denote $\gamma := \frac{\langle u(\tau),h\rangle}{\|u(\tau)\|_* \cdot \|h\|} \in [-1,1]$. Then,

$$\left|\phi'(\tau)\right| \;\leq\; \|v-u\|_* \cdot \|h\|^2 \cdot \left(1 - \nu\gamma^2 + 2\nu|\gamma|\right) \;\leq\; (1+\nu)\cdot\|v-u\|_* \cdot \|h\|^2.$$

Thus, we have

$$\begin{array}{rcl}\left|\left\langle (\nabla^2 f(x) - \nabla^2 f(y))h, h\right\rangle\right| & = & |\phi(1) - \phi(0)| \\ & \leq & (1+\nu)\cdot\|v-u\|_*\cdot\|h\|^2.\end{array} \tag{2.1.18}$$

It remains to use the definition of u and v and apply inequality (2.1.12) with $p = s$. Thus, we have proved, that for $q = 2+\nu$ the Hessian of f is Hölder continuous of degree ν. At the same time, taking $y = 0$, we get $\|\nabla^2 f(x) - \nabla^2 f(y)\| = \|\nabla^2 f(x)\| = (1+\nu)\|x\|^\nu$. These values cannot be uniformly bounded in $x \in \mathbb{E}$ by any multiple of $\|x\|^\alpha$ with $\alpha \neq \nu$. So, the Hessian of f is *not* Hölder continuous for any degree different from $2+\nu$. □

Remark 2.1.6. Inequalities (2.1.11) and (2.1.12) have the following symmetric consequences:

$$p \geq 2 \quad\Rightarrow\quad \|\nabla f_p(x) - \nabla f_p(y)\|_* \;\geq\; 2^{2-p}\|x-y\|^{p-1},$$

$$p \leq 2 \quad\Rightarrow\quad \|\nabla f_p(x) - \nabla f_p(y)\|_* \;\leq\; 2^{2-p}\|x-y\|^{p-1},$$

which are valid for all $x, y \in \mathbb{E}$.

2.1.2 Regularized Newton Method

First, let us consider the case when we know that for a specific $\nu \in [0,1]$ function f has a Hölder continuous Hessian: $\mathcal{H}_\nu < +\infty$. Then, from (2.1.5), we have the global upper bound for the objective function, for all $x, y \in$

$\mathrm{dom}\,\psi$:

$$F(y) \quad \leq \quad M_{\nu,H}(x;y) \quad \stackrel{\mathrm{def}}{=} \quad \Omega_2(f,x;y) + \tfrac{H\|y-x\|^{2+\nu}}{(1+\nu)(2+\nu)} + \psi(y),$$

when the regularization parameter is large enough: $H \geq \mathcal{H}_\nu$. Thus, it is natural to employ the minimum of a regularized quadratic model:

$$T_{\nu,H}(x) \;\stackrel{\mathrm{def}}{=}\; \operatorname*{argmin}_y M_{\nu,H}(x;y), \qquad M^*_{\nu,H}(x) \;\stackrel{\mathrm{def}}{=}\; \min_y M_{\nu,H}(x;y),$$

and define the following general iteration process [60]:

$$\boxed{x_0 \;\in\; \mathrm{dom}\,\psi, \quad x_{k+1} \;=\; T_{\nu,H_k}(x_k), \quad k \geq 0} \tag{2.1.19}$$

where the value H_k is chosen either to be a constant from the interval $(0, 2\mathcal{H}_\nu]$ or by some adaptive procedure.

For the class of uniformly convex functions of degree $q = 2+\nu$, we can justify the following global convergence result for this process.

Theorem 2.1.7. *Assume that for some $\nu \in [0,1]$ we have $0 < \mathcal{H}_\nu < +\infty$ and $\sigma_{2+\nu} > 0$. Let the coefficients $\{H_k\}_{k\geq 0}$ in process (2.1.19) satisfy the following conditions:*

$$0 \;\leq\; H_k \;\leq\; \beta\mathcal{H}_\nu, \qquad F(x_{k+1}) \;\leq\; M^*_{\nu,H_k}(x_k), \qquad k \geq 0, \tag{2.1.20}$$

with some constant $\beta \geq 0$. Then for the sequence $\{x_k\}_{k\geq 0}$ generated by the process we have

$$\begin{aligned} & F(x_{k+1}) - F^* \\ & \leq \; \left(1 \;-\; \tfrac{1+\nu}{2+\nu} \cdot \min\Bigl\{\tfrac{(1+\nu)}{\omega_\nu(1+\beta)(2+\nu)}, 1\Bigr\}^{\frac{1}{1+\nu}}\right)(F(x_k) - F^*)\,. \end{aligned} \tag{2.1.21}$$

Thus, the rate of convergence is linear and for reaching the gap

$$F(x_K) - F^* \quad \leq \quad \varepsilon$$

it is enough to perform

$$K \quad = \quad \Bigl\lceil \tfrac{2+\nu}{1+\nu} \cdot \max\Bigl\{\tfrac{\omega_\nu(1+\beta)(2+\nu)}{(1+\nu)}, 1\Bigr\}^{\frac{1}{1+\nu}} \log \tfrac{F(x_0)-F^*}{\varepsilon} \Bigr\rceil$$

iterations.

Proof. Let us fix an arbitrary $k \geq 0$ and consider the progress achieved at one step of the method. For any $y \in \operatorname{dom} \psi$, we have

$$\begin{aligned} F(x_{k+1}) & \overset{(2.1.20)}{\leq} M^*_{\nu,H_k}(x_k) \;\; \leq \;\; \Omega_2(f, x_k; y) + \tfrac{H_k \|y - x_k\|^{2+\nu}}{(1+\nu)(2+\nu)} + \psi(y) \\ & \overset{(2.1.5)}{\leq} F(y) + \tfrac{(H_k + \mathcal{H}_\nu)\|y - x_k\|^{2+\nu}}{(1+\nu)(2+\nu)} \\ & \overset{(2.1.20)}{\leq} F(y) + \tfrac{(1+\beta)\mathcal{H}_\nu \|y - x_k\|^{2+\nu}}{(1+\nu)(2+\nu)}. \end{aligned}$$

Now, define $y = \alpha x^* + (1-\alpha)x_k$, with $\alpha \in [0,1]$. Hence, we obtain

$$F(x_{k+1}) \;\; \leq \;\; F(x_k) - \alpha\,(F(x_k) - F^*) + \alpha^{2+\nu} \tfrac{(1+\beta)\mathcal{H}_\nu \|x_k - x^*\|^{2+\nu}}{(1+\nu)(2+\nu)},$$

Then, taking into account uniform convexity (2.0.1), we get

$$F(x_{k+1}) \;\; \leq \;\; F(x_k) - \Bigl(\alpha - \alpha^{2+\nu} \tfrac{(1+\beta)\mathcal{H}_\nu}{(1+\nu)\sigma_{2+\nu}} \Bigr) (F(x_k) - F^*) .$$

The minimum of the right-hand side is attained at

$$\alpha^* \;\; = \;\; \min\bigl\{ \tfrac{(1+\nu)}{\omega_\nu (2+\nu)(1+\beta)}, 1 \bigr\}^{\frac{1}{1+\nu}} .$$

Plugging this value into the bound above, we get inequality (2.1.21). □

Unfortunately, in practice it is difficult to decide on an appropriate value of $\nu \in [0,1]$ with $\mathcal{H}_\nu < +\infty$. Hence, it is interesting to develop *universal methods* that are not based on some particular parameters. Recently, it was shown [60], that one good choice for such universal scheme is the Cubic Newton. This is actually the process (2.1.19) with the fixed parameter $\nu = 1$.

From now on, we omit the unnecessary index:

$$M_H(x; y) \;\; \overset{\text{def}}{=} \;\; M_{1,H}(x; y) \;\; \equiv \;\; \Omega_2(f, x; y) + \tfrac{H\|y - x\|^3}{6} + \psi(y),$$

$$T_H(x) \;\; \overset{\text{def}}{=} \;\; T_{1,H}(x) \;\; \equiv \;\; \operatorname*{argmin}_y M_{1,H}(x; y),$$

and

$$M^*_H(x) \;\; \overset{\text{def}}{=} \;\; M^*_{1,H}(x) \;\; \equiv \;\; M_H(x; T_H(x)).$$

The adaptive scheme of our method with dynamic estimation of the constant H is as follows.

Adaptive Cubic Regularization of Newton Method

```
Initialization. Choose $x_0 \in \operatorname{dom}\psi$, $H_0 > 0$.

Iteration $k \geq 0$.

1: Find the minimal integer $i_k \geq 0$ such that
       $F(T_{H_k 2^{i_k}}(x_k)) \leq M^*_{H_k 2^{i_k}}(x_k)$.

2: Perform the Cubic Step: $x_{k+1} = T_{H_k 2^{i_k}}(x_k)$.

3: Set $H_{k+1} := \frac{1}{2} H_k 2^{i_k}$.
```

(2.1.22)

We present now the main properties of the composite Cubic Newton step $x \mapsto T_H(x)$. We start by denoting

$$r_H(x) \quad \overset{\text{def}}{=} \quad \|T_H(x) - x\|.$$

Since point $T_H(x)$ is a minimum of the strictly convex function $M_H(x;\cdot)$, it satisfies the following first-order optimality condition, for all $y \in \operatorname{dom}\psi$:

$$\begin{aligned} &\big\langle \nabla f(x) + \nabla^2 f(x)(T_H(x) - x) \\ &\quad + \tfrac{H r_H(x)}{2} B(T_H(x) - x), y - T_H(x) \big\rangle \; + \psi(y) \;\geq\; \psi(T_H(x)). \end{aligned} \tag{2.1.23}$$

In other words, the vector

$$\psi'(T_H(x)) \quad \overset{\text{def}}{=} \quad -\nabla f(x) - \nabla^2 f(x)(T_H(x) - x) - \tfrac{H r_H(x)}{2} B(T_H(x) - x)$$

belongs to the subdifferential of ψ:

$$\psi'(T_H(x)) \quad \in \quad \partial\psi(T_H(x)). \tag{2.1.24}$$

We have discussed computation of the point $T = T_H(x)$, satisfying condition (2.1.24) in Section 1.4.9. Arithmetical complexity of such a procedure is usually similar to that for the standard Newton step.

Plugging into (2.1.23) $y := x \in \operatorname{dom} \psi$, we get:

$$\begin{aligned} \langle \nabla f(x), x - T_H(x) \rangle \quad \geq \quad & \langle \nabla^2 f(x)(T_H(x) - x), T_H(x) - x \rangle \\ & + \tfrac{H r_H^3(x)}{2} + \psi(T_H(x)) - \psi(x). \end{aligned} \tag{2.1.25}$$

Thus, we obtain the following bound for the minimal value $M_H^*(x)$ of the cubic model:

$$\begin{aligned} M_H^*(x) \quad & \overset{(2.1.25)}{\leq} \quad f(x) - \tfrac{1}{2}\langle \nabla^2 f(x)(T_H(x) - x), T_H(x) - x \rangle \\ & \qquad - \tfrac{H r_H^3(x)}{3} + \psi(x) \\ & = \quad F(x) - \tfrac{1}{2}\langle \nabla^2 f(x)(T_H(x) - x), T_H(x) - x \rangle \\ & \qquad - \tfrac{H r_H^3(x)}{3}. \end{aligned} \tag{2.1.26}$$

If for some value $\nu \in [0,1]$ the Hessian is Hölder continuous: $\mathcal{H}_\nu < +\infty$, then by (2.1.4) and (2.1.24) we get for the subgradient at new point,

$$F'(T_H(x)) \quad \overset{\text{def}}{=} \quad \nabla f(T_H(x)) + \psi'(T_H(x)),$$

the following bound:

$$\begin{aligned} & \| F'(T_H(x)) \|_* \\ & \quad \leq \quad \| \nabla f(T_H(x)) - \nabla f(x) - \nabla^2 f(x)(T_H(x) - x) \|_* \\ & \qquad + \tfrac{H r_H^2(x)}{2} \\ & \quad \overset{(2.1.4)}{\leq} \quad \tfrac{\mathcal{H}_\nu r_H^{1+\nu}(x)}{1+\nu} + \tfrac{H r_H^2(x)}{2} = r_H^{1+\nu}(x) \left(\tfrac{\mathcal{H}_\nu}{1+\nu} + \tfrac{H r_H^{1-\nu}(x)}{2} \right). \end{aligned} \tag{2.1.27}$$

One of the main strong points of the classical Newton's method is its local *quadratic convergence* for the class of strongly convex functions with Lipschitz continuous Hessian: $\sigma_2 > 0$ and $0 < \mathcal{H}_1 < +\infty$ (see Section 1.4.1). This property holds for the cubically regularized Newton as well [124, 111].

Indeed, ensuring $F(T_H(x)) \leq M_H^*(x)$ as in algorithm (2.1.22), and having $H \leq \beta \mathcal{H}_1$ with some $\beta \geq 0$, we get:

$$\begin{array}{rcl}
F(T_H(x)) - F^* & \overset{(2.0.3)}{\leq} & \frac{1}{2\sigma_2}\|F'(T_H(x))\|_*^2 \overset{(2.1.27)}{\leq} \frac{(1+\beta)^2\mathcal{H}_1^2}{8\sigma_2} r_H^4(x) \\
& \leq & \frac{(1+\beta)^2\mathcal{H}_1^2}{8\sigma_2^3}\langle \nabla^2 f(x)(T_H(x)-x), T_H(x)-x\rangle^2 \\
& \overset{(2.1.26)}{\leq} & \frac{(1+\beta)^2\mathcal{H}_1^2}{2\sigma_2^3}\left(F(x)-F^*\right)^2 .
\end{array}$$

And the region of quadratic convergence is as follows:

$$\mathcal{Q} \;=\; \left\{ x \,:\, F(x) - F^* \;\leq\; \frac{2\sigma_2^3}{(1+\beta)^2\mathcal{H}_1^2} \right\}.$$

After reaching it, the method starts to double the number of the right digits of the answer at every step, and this cannot last for a long time. Therefore, from now on we are mainly interested in the *global complexity bounds* of algorithm (2.1.22), which work for an arbitrary starting point x_0.

For noncomposite case, as it was shown in [60], if for some $\nu \in [0,1]$ we have $0 < \mathcal{H}_\nu < +\infty$ and the objective is just *convex*, then algorithm (2.1.22) with small initial parameter H_0 generates a solution $\hat{x}$ with $f(\hat{x}) - f^* \leq \varepsilon$ in

$$\mathcal{O}\Bigl(\bigl(\tfrac{\mathcal{H}_\nu D_0^{2+\nu}}{\varepsilon}\bigr)^{\frac{1}{1+\nu}} \Bigr)$$

iterations, where $D_0 \;=\; \sup\limits_x \{\|x - x^*\| \;:\; f(x) \leq f(x_0)\}$. Thus, the method has a sublinear rate of convergence on the class of convex functions with Hölder continuous Hessian. It can *automatically adapt* to the actual level of smoothness. In what follows we show that the same algorithm achieves linear rate of convergence for the class of *uniformly convex* functions of degree $q = 2 + \nu$, namely for functions with bounded condition number: $\inf_{\nu \in [0,1]} \omega_\nu < +\infty$.

In the remaining part, we usually assume that the smooth part of our objective is not *purely quadratic*. This is equivalent to the condition

$$\inf_{\nu \in [0,1]} \mathcal{H}_\nu \;>\; 0.$$

However, to conclude this section, let us briefly discuss the case

$$\min_{\nu \in [0,1]} \mathcal{H}_\nu = 0.$$

If we knew in advance that f is a convex quadratic function, then no regularization would be needed since a single step $x \mapsto T_H(x)$ with $H = 0$ would solve the problem. However, if our function is given by a black-box oracle and we do not know a priori that its smooth part is quadratic, then we can still use algorithm (2.1.22). For this case we prove the following simple result.

Proposition 2.1.8. *Let $A : \mathbb{E} \to \mathbb{E}^*$ be a self-adjoint positive semidefinite linear operator and $b \in \mathbb{E}^*$. Assume that $f(x) \; = \; \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$, and the minimum $x^* \in \mathrm{Argmin}_x \bigl\{ F(x) \overset{\mathrm{def}}{=} f(x) + \psi(x) \bigr\}$ does exist. Then, in order to get $F(x_K) - F^* \leq \varepsilon$ with arbitrary $\varepsilon > 0$, it is enough to perform*

$$K \quad = \quad \left\lceil \log_2 \tfrac{H_0 \|x_0 - x^*\|^3}{6\varepsilon} \; + \; 1 \right\rceil \tag{2.1.28}$$

iterations of algorithm (2.1.22). Therefore, the convergence of the method is very fast in the quadratic case.

Proof. In our case, the quadratic model coincides with the smooth part of the objective: $\Omega_2(f, x; y) \equiv f(y)$, for all $x, y \in \mathbb{E}$. Therefore, at every iteration k of algorithm (2.1.22) we have $i_k = 0$ and $H_k = 2^{-k} H_0$. Note that $x_{k+1} = T_{2^{-k}H_0}(x_k) = \mathrm{argmin}_y \bigl\{ F(y) + \frac{2^{-k}H_0}{6}\|y - x_k\|^3 \bigr\}$, and

$$F(x_{k+1}) \quad \leq \quad F(y) + \tfrac{2^{-k}H_0}{6}\|y - x_k\|^3, \qquad \forall y \in \mathrm{dom}\, \psi. \tag{2.1.29}$$

If it holds that $\|x_{k+1} - x^*\| \leq \|x_k - x^*\|$ for all $k \geq 0$, then by plugging $y \equiv x^*$ into (2.1.29), we get $F(x_{k+1}) - F^* \leq 2^{-k}\frac{H_0}{6}\|x_0 - x^*\|^3$, which results in the estimate (2.1.28). In order to verify $\|x_{k+1} - x^*\| \leq \|x_k - x^*\|$, observe that

$$\begin{array}{rcl} \|x_k - x^*\|^2 & = & \|(x_k - x_{k+1}) + (x_{k+1} - x^*)\|^2 \\[2ex] & = & \|x_{k+1} - x^*\|^2 + \|x_k - x_{k+1}\|^2 \\[2ex] & & \qquad + \, 2\langle B(x_k - x_{k+1}), x_{k+1} - x^*\rangle. \end{array}$$

Then it is enough to show that $\langle B(x_k - x_{k+1}), x^* - x_{k+1}\rangle \; \leq \; 0$. Since x_{k+1} satisfies the first-order optimality condition:

$$-2^{-(k+1)}H_0\|x_{k+1} - x_k\|B(x_{k+1} - x_k) \quad \overset{\mathrm{def}}{=} \quad F'(x_{k+1}) \; \in \; \partial F(x_{k+1}),$$

we have

$$\langle B(x_k - x_{k+1}), x^* - x_{k+1} \rangle \quad = \quad \tfrac{2^{k+1}}{H_0 \| x_k - x_{k+1} \|} \langle F'(x_{k+1}), x^* - x_{k+1} \rangle \; \leq \; 0,$$

where the last inequality follows from the convexity of the objective. □

2.1.3 Complexity of the Universal Scheme

In this section, we are going to justify the global linear rate of convergence of algorithm (2.1.22) for the class of twice differentiable uniformly convex functions with Hölder continuous Hessian. Universality of this method is ensured by the adaptive estimation of the parameter H over the whole sequence of iterations. It is important to distinguish two cases: $H_{k+1} < H_k$ and $H_{k+1} \geq H_k$.

First, we need to estimate the progress in the objective function after minimizing the cubic model. There are two different situations here:

$$\text{either} \quad H r_H^{1-\nu}(x) \; \leq \; \tfrac{2\mathcal{H}_\nu}{1+\nu}, \quad \text{or} \quad H r_H^{1-\nu}(x) \; > \; \tfrac{2\mathcal{H}_\nu}{1+\nu}.$$

Lemma 2.1.9. *Let $0 < \mathcal{H}_\nu < +\infty$ and $\sigma_{2+\nu} > 0$ for some $\nu \in [0,1]$. Then for an arbitrary $x \in \operatorname{dom} \psi$ and $H > 0$, we have:*

$$\begin{array}{l} F(x) - M_H^*(x) \\ \geq \; \min\Big[(F(x) - F^*) \cdot \tfrac{(1+\nu)}{(2+\nu)} \cdot \min\big\{ \big(\tfrac{(1+\nu)}{2(2+\nu)\omega_\nu} \big)^{\frac{1}{1+\nu}}, \; 1 \big\}, \\ \quad (F(T_H(x)) - F^*)^{\frac{3(1+\nu)}{2(2+\nu)}} \cdot \big(\tfrac{2+\nu}{1+\nu} \big)^{\frac{3(1+\nu)}{2(2+\nu)}} \cdot \tfrac{(\sigma_{2+\nu})^{\frac{3}{2(2+\nu)}}}{3\sqrt{H}} \Big]. \end{array} \tag{2.1.30}$$

Proof. Let us consider two cases. 1) $H r_H^{1-\nu}(x) \leq \frac{2\mathcal{H}_\nu}{1+\nu}$. Then, for an arbitrary $y \in \operatorname{dom} \psi$, we have:

$$\begin{array}{rcl} M_H^*(x) & = & \Omega_2(f, x; T_H(x)) + \frac{H \| T_H(x) - x \|^3}{6} + \psi(T_H(x)) \\ & \leq & \Omega_2(f, x; y) + \frac{H r_H^{1-\nu}(x) \| y - x \|^{2+\nu}}{2(2+\nu)} + \psi(y) \\ & \overset{(2.1.5)}{\leq} & F(y) + \frac{\mathcal{H}_\nu \| y - x \|^{2+\nu}}{(1+\nu)(2+\nu)} + \frac{H r_H^{1-\nu}(x) \| y - x \|^{2+\nu}}{2(2+\nu)} \\ & \leq & F(y) + \frac{2\mathcal{H}_\nu \| y - x \|^{2+\nu}}{(1+\nu)(2+\nu)}, \end{array}$$

where the first inequality follows from the fact that

$$T_H(x) \quad = \quad \operatorname*{argmin}_y \Big\{ \Omega_2(f,x;y) + \tfrac{Hr_H^{1-\nu}(x)\|y-x\|^{2+\nu}}{2(2+\nu)} + \psi(y) \Big\}.$$

Let us restrict y to the segment: $y = \alpha x^* + (1-\alpha)x$, with $\alpha \in [0,1]$. Taking into account the uniform convexity of the objective, we get:

$$\begin{aligned} M_H^*(x) \quad & \leq \quad F(x) - \alpha\,(F(x) - F^*) + \alpha^{2+\nu}\tfrac{2\mathcal{H}_\nu\|x-x^*\|^{2+\nu}}{(1+\nu)(2+\nu)} \\ & \overset{(2.0.1)}{\leq} \quad F(x) - \Big(\alpha - \alpha^{2+\nu}\tfrac{2\mathcal{H}_\nu}{(1+\nu)\sigma_{2+\nu}}\Big)\,(F(x) - F^*)\,. \end{aligned}$$

The minimum of the right-hand side is attained at

$$\alpha^* \quad = \quad \min\Big\{ \tfrac{(1+\nu)}{2(2+\nu)\omega_\nu}, 1 \Big\}^{\frac{1}{1+\nu}}.$$

By plugging this value into the bound, we have:

$$M_H^*(x) \quad \leq \quad F(x) - \min\Big\{ \Big(\tfrac{(1+\nu)}{2(2+\nu)\omega_\nu}\Big)^{1/(1+\nu)},\ 1 \Big\} \cdot \tfrac{(1+\nu)}{(2+\nu)} \cdot (F(x) - F^*)\,,$$

and this is the first argument of the minimum in (2.1.30).

2) $Hr_H^{1-\nu}(x) > \frac{2\mathcal{H}_\nu}{1+\nu}$. By (2.1.27) we have the bound:

$$\|F'(T_H(x))\|_* \quad < \quad Hr_H^2(x). \tag{2.1.31}$$

Because $\nabla^2 f(x) \succeq 0$, we get the second argument of the minimum:

$$\begin{aligned} F(x) - M_H^*(x) \quad & \overset{(2.1.26)}{\geq} \quad \tfrac{Hr_H^3(x)}{3} \quad \overset{(2.1.31)}{\geq} \quad \tfrac{\|F'(T_H(x))\|_*^{\frac{3}{2}}}{3\sqrt{H}} \\ & \overset{(2.0.3)}{\geq} \quad \Big(\tfrac{2+\nu}{1+\nu}\Big)^{\frac{3(1+\nu)}{2(2+\nu)}} \cdot \tfrac{(\sigma_{2+\nu})^{\frac{3}{2(2+\nu)}}}{3\sqrt{H}} \\ & \qquad \cdot\ (F(T_H(x)) - F^*)^{\frac{3(1+\nu)}{2(2+\nu)}}\,. \end{aligned}$$

□

Denote by ϑ_ν the following auxiliary value, for $\nu \in [0,1]$:

$$\vartheta_\nu \quad \overset{\text{def}}{=} \quad \frac{\mathcal{H}_\nu^{\frac{2}{1+\nu}}}{(\sigma_{2+\nu})^{\frac{1-\nu}{(1+\nu)(2+\nu)}}} \cdot \frac{6\cdot(8+\nu)^{\frac{1-\nu}{1+\nu}}}{((1+\nu)(2+\nu))^{\frac{2}{1+\nu}}} \cdot \left(\tfrac{1+\nu}{2+\nu}\right)^{\frac{1-\nu}{2+\nu}}. \tag{2.1.32}$$

Then, the next lemma shows what happens when the parameter H is increasing during the iterations.

Lemma 2.1.10. *Assume that for a fixed $x \in \operatorname{dom}\psi$ the parameter $H > 0$ is such that:*

$$F(T_H(x)) \quad > \quad M_H^*(x). \tag{2.1.33}$$

If for some $\nu \in [0,1]$, we have $\sigma_{2+\nu} > 0$, then it holds:

$$H\left(F(T_{2H}(x)) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad < \quad \vartheta_\nu. \tag{2.1.34}$$

Proof. Firstly, let us prove, that from (2.1.33) we have

$$Hr_H^{1-\nu}(x) \quad < \quad \tfrac{6\mathcal{H}_\nu}{(1+\nu)(2+\nu)}. \tag{2.1.35}$$

Assuming by contradiction, $Hr_H^{1-\nu}(x) \;\geq\; \frac{6\mathcal{H}_\nu}{(1+\nu)(2+\nu)}$, we get

$$\begin{aligned} M_H^*(x) \quad &= \quad \tfrac{H\|T_H(x)-x\|^3}{6} + \Omega_2(f,x;T_H(x)) + \psi(T_H(x)) \\ &\geq \quad \tfrac{\mathcal{H}_\nu\|T_H(x)-x\|^{2+\nu}}{(1+\nu)(2+\nu)} + \Omega_2(f,x;T_H(x)) + \psi(T_H(x)) \\ &\overset{(2.1.5)}{\geq} \quad F(T_H(x)), \end{aligned}$$

which contradicts (2.1.33). Secondly, by its definition, $M_H^*(x)$ is a concave function of H. Therefore, its derivative $\frac{d}{dH}M_H^*(x) = \frac{1}{6}r_H^3(x)$ is non-increasing. Hence, it holds:

$$r_{2H}(x) \quad \leq \quad r_H(x) \quad \overset{(2.1.35)}{<} \quad \left(\tfrac{6\mathcal{H}_\nu}{(1+\nu)(2+\nu)H}\right)^{\frac{1}{1-\nu}}. \tag{2.1.36}$$

Finally, by the smoothness and the uniform convexity, we obtain:

$$\begin{aligned}
& H\left(F(T_{2H}(x)) - F^*\right)^{\frac{1-\nu}{2+\nu}} \\
& \overset{(2.0.3)}{\leq} \quad H\left(\tfrac{1+\nu}{2+\nu}\left(\tfrac{1}{\sigma_{2+\nu}}\right)^{\frac{1}{1+\nu}}\right)^{\frac{1-\nu}{2+\nu}} \|F'(T_{2H}(x))\|_*^{\frac{1-\nu}{1+\nu}} \\
& \overset{(2.1.27)}{\leq} \quad H\left(\tfrac{1+\nu}{2+\nu}\left(\tfrac{1}{\sigma_{2+\nu}}\right)^{\frac{1}{1+\nu}}\right)^{\frac{1-\nu}{2+\nu}} \left(r_{2H}^{1+\nu}(x) \cdot \left(\tfrac{\mathcal{H}_\nu}{1+\nu} + H r_{2H}^{1-\nu}(x)\right)\right)^{\frac{1-\nu}{1+\nu}} \\
& \overset{(2.1.36)}{<} \quad H\left(\tfrac{1+\nu}{2+\nu}\left(\tfrac{1}{\sigma_{2+\nu}}\right)^{\frac{1}{1+\nu}}\right)^{\frac{1-\nu}{2+\nu}} \left(r_{2H}^{1+\nu}(x) \cdot \tfrac{(8+\nu)\mathcal{H}_\nu}{(1+\nu)(2+\nu)}\right)^{\frac{1-\nu}{1+\nu}} \\
& \overset{(2.1.36)}{<} \quad \left(\tfrac{1+\nu}{2+\nu}\left(\tfrac{1}{\sigma_{2+\nu}}\right)^{\frac{1}{1+\nu}}\right)^{\frac{1-\nu}{2+\nu}} \left(\tfrac{\mathcal{H}_\nu}{(1+\nu)(2+\nu)}\right)^{\frac{2}{1+\nu}} 6(8+\nu)^{\frac{1-\nu}{1+\nu}} \\
& \overset{\text{def}}{=} \quad \vartheta_\nu. \qquad \Box
\end{aligned}$$

We are now ready to prove the main result about the universal scheme.

Theorem 2.1.11. *Assume that for a fixed* $\nu \in [0,1]$ *we have* $0 < \mathcal{H}_\nu < +\infty$ *and* $\sigma_{2+\nu} > 0$. *Let parameter* H_0 *in algorithm* (2.1.22) *be small enough:*

$$H_0 \quad \leq \quad \tfrac{\vartheta_\nu}{(F(x_0)-F^*)^{(1-\nu)/(2+\nu)}}, \tag{2.1.37}$$

where ϑ_ν *is defined by* (2.1.32). *Let the sequence* $\{x_k\}_{k=0}^K$ *generated by the method satisfy condition:*

$$F(T_{H_k 2^j}(x_k)) - F^* \;\geq\; \varepsilon \;>\; 0, \qquad 0 \leq j \leq i_k, \quad 0 \leq k \leq K-1. \tag{2.1.38}$$

Then, for every $0 \leq k \leq K-1$, *we have*

$$\begin{aligned}
& F(x_{k+1}) - F^* \\
& \leq \; \left(1 - \min\left\{\tfrac{(2+\nu)((1+\nu)(2+\nu))^{1/(1+\nu)}}{(1+\nu)6^{3/2}\cdot 2^{1/2}\cdot(8+\nu)^{(1-\nu)/(2+2\nu)}} \cdot \tfrac{1}{(\omega_\nu)^{\frac{1}{1+\nu}}}, \tfrac{1}{2}\right\}\right) \qquad (2.1.39) \\
& \qquad \cdot \left(F(x_k) - F^*\right).
\end{aligned}$$

Therefore, the rate of convergence is linear, and

$$K \quad \leq \quad \max\left\{(\omega_\nu)^{\frac{1}{1+\nu}} \cdot \tfrac{1+\nu}{2+\nu} \cdot \tfrac{6^{3/2}\cdot 2^{1/2}\cdot(8+\nu)^{(1-\nu)/(2+2\nu)}}{((1+\nu)(2+\nu))^{1/(1+\nu)}}, 1\right\} \cdot \log \tfrac{F(x_0)-F^*}{\varepsilon}.$$

Moreover, we have the following bound for the total number of oracle calls N_K during the first K iterations:

$$N_K \quad \leq \quad 2K + \log_2 \tfrac{\vartheta_\nu}{\varepsilon^{(1-\nu)/(2+\nu)}} - \log_2 H_0. \tag{2.1.40}$$

Proof. The proof is based on Lemmas 2.1.9 and 2.1.10, and monotonicity of the sequence $\bigl\{F(x_k)\bigr\}_{k\geq 0}$. Firstly, we need to show that every iteration of the method is well-defined. Namely, we are going to verify that for a fixed $0 \leq k \leq K-1$, there exists a finite integer $\ell \geq 0$ such that either $F(T_{H_k 2^\ell}(x_k)) \leq M^*_{H_k 2^\ell}(x_k)$ or $F(T_{H_k 2^{\ell+1}}(x_k)) - F^* < \varepsilon$.

Indeed, let us set

$$\ell \quad := \quad \max\Bigl\{0, \log_2 \Bigl\lceil \tfrac{\vartheta_\nu}{H_k \varepsilon^{(1-\nu)/(2+\nu)}} \Bigr\rceil \Bigr\},$$

and

$$H \quad := \quad H_k 2^\ell \quad \geq \quad \tfrac{\vartheta_\nu}{\varepsilon^{(1-\nu)/(2+\nu)}}. \tag{2.1.41}$$

Then, if we have both $F(T_H(x_k)) > M^*_H(x_k)$, and $F(T_{2H}(x_k)) - F^* \geq \varepsilon$, we get by Lemma 2.1.10:

$$H \quad \overset{(2.1.34)}{<} \quad \tfrac{\vartheta_\nu}{(F(T_{2H}(x_k)) - F^*)^{(1-\nu)/(2+\nu)}} \quad \leq \quad \tfrac{\vartheta_\nu}{\varepsilon^{(1-\nu)/(2+\nu)}},$$

which contradicts (2.1.41). Therefore, if we are unable to find the value $0 \leq i_k \leq \ell$ (see step 1 of the algorithm) in a finite number of steps, that only means we have already solved the problem up to accuracy ε.

Now, let us show that for every $0 \leq k \leq K$ it holds:

$$H_k \left(F(x_k) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad \leq \quad \max\{\vartheta_\nu,\ H_0 \left(F(x_0) - F^*\right)^{\frac{1-\nu}{2+\nu}}\}. \tag{2.1.42}$$

This inequality is obviously valid for $k = 0$. Assume it is also valid for some $k \geq 0$. Then, by definition of H_{k+1} (see step 3 of the algorithm), we have $H_{k+1} = H_k 2^{i_k - 1}$. There are two cases. 1) $i_k = 0$. Then $H_{k+1} < H_k$. By monotonicity of $\bigl\{F(x_k)\bigr\}_{k\geq 0}$ and by induction, we get:

$$\begin{aligned} H_{k+1} \left(F(x_{k+1}) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad &< \quad H_k \left(F(x_k) - F^*\right)^{\frac{1-\nu}{2+\nu}} \\ &\leq \quad \max\{\vartheta_\nu,\ H_0 \left(F(x_0) - F^*\right)^{\frac{1-\nu}{2+\nu}}\}. \end{aligned}$$

2) $i_k > 0$. Then applying Lemma 2.1.10 with $H := H_k 2^{i_k - 1} = H_{k+1}$

and $x := x_k$, we have:

$$H_{k+1}\left(F(x_{k+1}) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad = \quad H\left(F(T_{2H}(x)) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad \overset{(2.1.34)}{\leq} \quad \vartheta_\nu.$$

Thus, (2.1.42) is true by induction. Choosing H_0 small enough (2.1.37), we have:

$$2H_k\left(F(x_k) - F^*\right)^{\frac{1-\nu}{2+\nu}} \quad \leq \quad 2\vartheta_\nu, \qquad 0 \leq k \leq K. \tag{2.1.43}$$

From Lemma 2.1.9 we know, that one of the two following estimates is true (denote $\delta_k := F(x_k) - F^*$):

1. $F(x_k) - F(x_{k+1}) \geq \alpha \cdot \delta_k \;\Leftrightarrow\; \delta_{k+1} \leq (1-\alpha)\cdot\delta_k,$ *or*
2. $F(x_k) - F(x_{k+1}) \geq \beta \cdot \delta_{k+1} \quad\Leftrightarrow\quad \delta_{k+1} \leq (1+\beta)^{-1}\delta_k \leq (1 - \min\{\beta, 1\}/2)\delta_k,$

where

$$\alpha \quad \overset{\text{def}}{=} \quad \tfrac{1+\nu}{2+\nu}\cdot\min\Bigl\{\bigl(\tfrac{(1+\nu)}{2(2+\nu)\omega_\nu}\bigr)^{\frac{1}{1+\nu}},\ 1\Bigr\},$$

and

$$\begin{aligned}\beta \quad &\overset{\text{def}}{=} \quad \bigl(\tfrac{2+\nu}{1+\nu}\bigr)^{\frac{3(1+\nu)}{2(2+\nu)}}\cdot\frac{(\sigma_{2+\nu})^{\frac{3}{2(2+\nu)}}}{3(2\vartheta_\nu)^{1/2}}\\ &\overset{(2.1.32)}{=} \quad \tfrac{2+\nu}{1+\nu}\cdot\tfrac{2^{1/2}\cdot((1+\nu)(2+\nu))^{\frac{1}{1+\nu}}}{6^{3/2}\cdot(8+\nu)^{(1-\nu)/(2+2\nu)}}\cdot\bigl(\tfrac{1}{\omega_\nu}\bigr)^{\frac{1}{1+\nu}}.\end{aligned}$$

It remains to notice, that $\alpha \geq \min\bigl\{\beta, 1\bigr\}/2$. Thus we obtain (2.1.39).

Finally, let us estimate the total number of the oracle calls N_K during the first K iterations. At each iteration the oracle is called $i_k + 1$ times, and we have $H_{k+1} = H_k 2^{i_k - 1}$. Therefore,

$$\begin{aligned} N_K \quad &= \quad \textstyle\sum_{k=0}^{K-1}(i_k+1) \;=\; \sum_{k=0}^{K-1}\Bigl(\log_2\frac{H_{k+1}}{H_k} + 2\Bigr)\\ &= \quad 2K + \log_2 H_K - \log_2 H_0\\ &\overset{(2.1.43),(2.1.38)}{\leq} \quad 2K + \log_2\tfrac{\vartheta_\nu}{\varepsilon^{(1-\nu)/(2+\nu)}} - \log_2 H_0. \qquad \Box\end{aligned}$$

Note that condition (2.1.37) for the initial choice of H_0 can be seen as a definition of the moment, after which we can guarantee the linear rate of convergence (2.1.39). In practice we can launch algorithm (2.1.22) with *arbitrary* $H_0 > 0$. There are two possible options: either the method halves

H_k at every step in the beginning, so H_k becomes small very quickly, or this value is increased at least once, and the required bound is guaranteed by Lemma 2.1.10. It can be easily proved, that this initial phase requires no more than $K_0 = \lceil \log_2 \frac{H_0 \varepsilon^{(1-\nu)/(1+\nu)}}{\vartheta_\nu} \rceil$ oracle calls.

2.1.4 Discussion

Let us discuss the global complexity results, provided by Theorem 2.1.11 for the Cubic Regularization of the Newton Method with the adaptive adjustment of the regularization parameter.

For the class of twice continuously differentiable μ-strongly convex functions with Lipschitz continuous gradients, it is well known that the classical Gradient Method needs

$$\mathcal{O}\Bigl(\tfrac{L_1}{\mu} \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr) \tag{2.1.44}$$

iterations for computing an ε-solution of the problem (e.g. [114]). As it was shown in [23], this result is shared by a variant of Cubic Regularization of the Newton Method. This is much better than the bound $\mathcal{O}\bigl(\bigl(\frac{L_1}{\mu}\bigr)^2 \log \frac{F(x_0)-F^*}{\varepsilon} \bigr)$, known for the Damped Newton Method (see Section 1.4.2).

For the class of uniformly convex functions of degree $q = 2 + \nu$ characterized by a Hölder continuous Hessian of degree $\nu \in [0,1]$ we have proved the following parametric estimates:

$$\mathcal{O}\Bigl(\max\bigl\{ \bigl(\omega_\nu\bigr)^{\frac{1}{1+\nu}}, 1 \bigr\} \cdot \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr),$$

where $\omega_\nu \overset{\mathrm{def}}{=} \frac{\mathcal{H}_\nu}{\sigma_{2+\nu}}$ is the *condition number* of degree ν. However, in practice we may not know exactly an appropriate value of the parameter ν. It is important that our algorithm automatically adjusts to the best possible complexity bound:

$$\mathcal{O}\Bigl(\max\bigl\{ \inf_{\nu \in [0,1]} \bigl(\omega_\nu\bigr)^{\frac{1}{1+\nu}}, \; 1 \; \bigr\} \cdot \log \tfrac{F(x_0) - F^*}{\varepsilon} \Bigr). \tag{2.1.45}$$

Note that for the functional class: $\forall x \in \operatorname{dom} \psi$ ($\mu B \preceq \nabla^2 f(x) \preceq L_1 B$), we have

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \quad \leq \quad L_1 - \mu, \qquad \forall x, y \in \operatorname{dom} \psi.$$

Thus, $\mathcal{H}_0 \leq L_1 - \mu$ and $\omega_0 \leq \frac{L_1 - \mu}{\mu}$. So we can conclude that estimate (2.1.45) is better than (2.1.44). Moreover, addition to our objective an *arbitrary* convex quadratic function does not change any of $\mathcal{H}_\nu$, $\nu \in [0, 1]$. Thus it can only improve the condition number ω_ν, while the ratio L_1/μ may become arbitrarily bad. It confirms an intuition, that a natural Newton-type minimization scheme should not be affected by any quadratic parts of the objective, and the notion of *well-conditioned* and *ill-conditioned* problems for second-order methods should be different from that of for first-order ones.

In the recent paper [61], a linear rate of convergence was also proven for the accelerated second-order scheme, with the complexity bound

$$\mathcal{O}\bigl(\max\{(\omega_\nu)^{\frac{1}{2+\nu}}, 1\} \cdot \log \tfrac{\mathcal{H}_\nu D_0^{2+\nu}}{\varepsilon}\bigr). \tag{2.1.46}$$

This is a better rate than (2.1.45). However, the method requires to know the parameter ν and the constant of uniform convexity. Thus, one theoretical question remains open: is it possible to construct *universal* second-order scheme matching (2.1.46) in the uniformly convex case?

It would be also interesting to generalize our results onto the Tensor Methods of arbitrary order. The situation with the methods of higher order is more difficult. One of the big issues is that Taylor's polynomial of degree $p \geq 3$ is generally nonconvex. We need to regularize the polynomial by the power of norm, with sufficiently big regularizing coefficient (see Section 1.5.1). Thus, using any adaptive strategy for the regularization coefficient, one need to ensure that the subproblem remains tractable. In our theoretical analysis, we also rely on the convexity of the model (in particular, in Lemma 2.1.9).

Note that one of the conditions to prove the universality of the method is (2.1.38), which bounds from below the residuals for all test points of the optimization process. This raises the question of *termination criterion* for the method. A working strategy is to do as many iterations of the method as our budget allows. An interesting open problem is to develop efficiently computable accuracy certificates for the Cubic Newton.

By looking at the definitions of $\mathcal{H}_\nu$ and $\sigma_{2+\nu}$, we can see that, for all $x, y \in \operatorname{dom} \psi, x \neq y$,

$$\mathcal{H}_\nu \quad \geq \quad \tfrac{\|\nabla^2 f(x) - \nabla^2 f(y)\|}{\|x-y\|^\nu}, \qquad \tfrac{1}{\sigma_{2+\nu}} \quad \geq \quad \tfrac{\|x-y\|^{2+\nu}}{\langle \nabla f(x) - \nabla f(y), x-y \rangle},$$

and

$$\begin{aligned} \omega_\nu \;=\; \tfrac{\mathcal{H}_\nu}{\sigma_{2+\nu}} \;&\geq\; \tfrac{\|\nabla^2 f(x) - \nabla^2 f(y)\| \cdot \|x-y\|^2}{\langle \nabla f(x) - \nabla f(y), x - y\rangle} \\ &\geq\; \tfrac{\langle (\nabla^2 f(x) - \nabla^2 f(y))(x-y), x-y\rangle}{\langle \nabla f(x) - \nabla f(y), x-y\rangle}. \end{aligned}$$

The last fraction does not depend on any particular ν and does not fix any norms. So, for any twice-differentiable convex function, we can define the following number:

$$\omega \;\overset{\mathrm{def}}{=}\; \sup\limits_{\substack{x,y \in \mathrm{dom}\,\psi \\ x \neq y}} \tfrac{\langle (\nabla^2 f(x) - \nabla^2 f(y))(x-y), x-y\rangle}{\langle \nabla f(x) - \nabla f(y), x-y\rangle}.$$

If it is finite ($\omega < +\infty$), then it could serve as an indicator of the *second-order non-degeneracy*, for which we have an upper bound: $\omega_\nu \geq \omega$, $\nu \in [0,1]$.

2.2 Local Convergence of Tensor Methods

One of the classical results about Newton's Method is its local quadratic convergence (see Section 1.4.1). For the Cubic Newton Method, its local superlinear convergence was justified in [124].

This part of the thesis is aimed to study local convergence of high-order methods, generalizing corresponding results from [124] in several ways. We establish local superlinear convergence of the Tensor Method (1.5.1) of degree $p \geq 2$, in the case when the composite objective is uniformly convex of arbitrary degree q from the interval $2 \leq q < p+1$. For strongly convex functions ($q = 2$), this gives the local convergence of order p.

We recall the definition of the Tensor step in Section 2.2.1. Then, we declare some of its properties, which are required for our analysis.

In Section 2.2.2, we prove local superlinear convergence of the Tensor Method in the function value and in the norm of minimal subgradient, under the assumption of uniform convexity of the objective.

In Section 2.2.3, we discuss the global behavior of the method and justify sublinear and linear global rates of convergence for convex and uniformly convex cases respectively.

One application of our developments is provided in Section 2.2.4. We show how local convergence can be applied for computing an inexact step in proximal methods. A global sublinear rate of convergence for the resulting scheme is also given.

2.2.1 Main Inequalities

For solving the composite minimization problem,

$$\min_x \Big\{ F(x) \;\; = \;\; f(x) + \psi(x) \Big\},$$

let us define one step $T = T_H(x)$ of the *regularized composite Tensor Method* of degree $p \geq 2$

$$T_H(x) \;\; \overset{\text{def}}{=} \;\; \operatorname*{argmin}_y \Big\{ \Omega_p(f, x; y) + \tfrac{H}{(p+1)!} \|y - x\|^{p+1} + \psi(y) \Big\}. \qquad (2.2.1)$$

Theorem 1.5.1 states that for

$$\boxed{H \;\geq\; pL_p} \qquad (2.2.2)$$

the auxiliary optimization problem in (2.2.1) is *convex*. This condition is crucial for the implementability of the Tensor Method and we always assume it is to be satisfied.

Then, we can write down the first-order optimality condition for the auxiliary subproblem in (2.2.1):

$$\begin{array}{l} \langle \nabla \Omega_p(f, x; T) + \frac{H}{p!} \|T - x\|^{p-1} B(T - x), y - T \rangle \\ \;+\; \psi(y) \;\;\geq\;\; \psi(T), \end{array} \qquad (2.2.3)$$

for all $y \in \operatorname{dom} \psi$. In other words, for vector

$$\psi(T) \;\; \overset{\text{def}}{=} \;\; -\Big(\nabla \Omega_p(f, x; T) + \tfrac{H}{p!} \|T - x\|^{p-1} B(T - x) \Big) \qquad (2.2.4)$$

we have $\psi'(T) \overset{(2.2.3)}{\in} \partial \psi(T)$. This fact explains our notation

$$F'(T) \;\; \overset{\text{def}}{=} \;\; \nabla f(T) + \psi'(T) \;\; \in \;\; \partial F(T). \qquad (2.2.5)$$

Let us present some properties of the point $T = T_H(x)$. First of all, we need some bounds for the norm of vector $F'(T)$. Note that

$$\begin{array}{l} \Big\| F'(T) + \frac{H}{p!} \|T - x\|^{p-1} B(T - x) \Big\|_* \\ \overset{(2.2.4)}{=} \;\; \Big\| \nabla f(T) - \nabla \Omega_p(f, x; T) \Big\|_* \;\; \overset{(1.3.6)}{\leq} \;\; \frac{L_p}{p!} \|T - x\|^p. \end{array} \qquad (2.2.6)$$

Consequently,

$$\|F'(T)\|_* \quad \leq \quad \tfrac{L_p+H}{p!}\|T-x\|^p. \tag{2.2.7}$$

Secondly, we use the following lemma.

Lemma 2.2.1. *Let $\beta > 1$ and $H = \beta L_p$. Then*

$$\begin{array}{rl} & \langle F'(T), x-T\rangle \\ \geq & \left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{1}{p}} \cdot \|F'(T)\|_*^{\frac{p+1}{p}} \cdot \tfrac{(\beta^2-1)^{\frac{p-1}{2p}}}{\beta} \cdot \tfrac{p}{(p^2-1)^{\frac{p-1}{2p}}}. \end{array} \tag{2.2.8}$$

In particular, if $\beta = p$, then

$$\langle F'(T), x-T\rangle \quad \geq \quad \left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{1}{p}} \cdot \|F'(T)\|_*^{\frac{p+1}{p}}. \tag{2.2.9}$$

Proof. Denote $r = \|T-x\|$, $h = \frac{H}{p!}$, and $l = \frac{L_p}{p!}$. Then inequality (2.2.6) can be written as follows:

$$\|F'(T) + hr^{p-1}B(T-x)\|_*^2 \quad \leq \quad l^2 r^{2p}.$$

This means that

$$\langle F'(T), x-T\rangle \quad \geq \quad \tfrac{1}{2hr^{p-1}}\|F'(T)\|_*^2 + \tfrac{r^{2p}(h^2-l^2)}{2hr^{p-1}}. \tag{2.2.10}$$

Denote

$$a \;=\; \tfrac{1}{2h}\|F'(T)\|_*^2, \quad b \;=\; \tfrac{h^2-l^2}{2h}, \quad \tau \;=\; r^{p-1}, \quad \alpha \;=\; \tfrac{p+1}{p-1}.$$

Then inequality (2.2.10) can be rewritten as follows:

$$\langle F'(T), x-T\rangle \quad \geq \quad \tfrac{a}{\tau} + b\tau^\alpha \;\geq\; \min_{t>0}\left\{\tfrac{a}{t} + bt^\alpha\right\} \;=\; (1+\alpha)\left(\tfrac{a}{\alpha}\right)^{\frac{\alpha}{1+\alpha}} b^{\frac{1}{1+\alpha}}.$$

By taking into account that $1+\alpha = \frac{2p}{p-1}$ and $\frac{\alpha}{1+\alpha} = \frac{p+1}{2p}$, and by using the actual meaning of a, b, and α, we get

$$\begin{aligned} \langle F'(T), x - T \rangle & \geq \frac{2p}{p-1} \cdot \frac{\|F'(T)\|_*^{\frac{p+1}{p}}}{(2h)^{\frac{p+1}{2p}}} \cdot \frac{(p-1)^{\frac{p+1}{2p}}}{(p+1)^{\frac{p+1}{2p}}} \cdot \frac{(h^2-l^2)^{\frac{p-1}{2p}}}{(2h)^{\frac{p-1}{2p}}} \\ & = \|F'(T)\|_*^{\frac{p+1}{p}} \cdot \frac{(h^2-l^2)^{\frac{p-1}{2p}}}{h} \cdot \frac{p}{(p+1)^{\frac{p+1}{2p}}(p-1)^{\frac{p-1}{2p}}} \\ & = \|F'(T)\|_*^{\frac{p+1}{p}} \cdot \frac{(h^2-l^2)^{\frac{p-1}{2p}}}{h} \cdot \frac{p}{(p^2-1)^{\frac{p-1}{2p}}(p+1)^{\frac{1}{p}}}. \end{aligned}$$

It remains to note that

$$\frac{(h^2-l^2)^{\frac{p-1}{2p}}}{h} = \frac{(H^2-L_p^2)^{\frac{p-1}{2p}}}{H} \cdot (p!)^{\frac{1}{p}} = \frac{(\beta^2-1)^{\frac{p-1}{2p}}}{\beta} \cdot \left(\frac{p!}{L_p}\right)^{\frac{1}{p}}.$$

□

2.2.2 Local Convergence

We analyse now the local behavior of the *Regularized Composite Tensor Method* (RCTM):

$$\boxed{x_0 \in \operatorname{dom}\psi, \quad x_{k+1} = T_H(x_k), \quad k \geq 0} \tag{2.2.11}$$

This is algorithm (1.5.1) with fixed regularization parameter. In order to prove local superlinear convergence of this scheme, we need one more assumption.

Assumption 2.2.2. For all $x, y \in \operatorname{dom}\psi$ and for all $F'(x) \in \partial F(x)$, $F'(y) \in \partial F(y)$, it holds:

$$\langle F'(x) - F'(y), x - y \rangle \geq \sigma_q \|x - y\|^q, \tag{2.2.12}$$

for some $q \geq 2$ and $\sigma_q > 0$. Hence, we assume that the full objective in the composite problem is uniformly convex of degree q (see Lemma 2.0.1).

Inequality (2.2.12) gives us the following local convergence rate for RCTM.

Theorem 2.2.3. *Let $\sigma_q > 0$ for some $q \geq 2$. Then for the sequence $\{x_k\}_{k \geq 0}$*

generated by method (2.2.11) *with* $H := pL_p$*, we have*

$$\begin{array}{rl} & F(x_{k+1}) - F^* \\ \leq & (q-1)q^{\frac{p-q+1}{q-1}} \left(\frac{1}{\sigma_q}\right)^{\frac{p+1}{q-1}} \left(\frac{L_p+H}{p!}\right)^{\frac{q}{q-1}} \left[F(x_k) - F^*\right]^{\frac{p}{q-1}}. \end{array} \tag{2.2.13}$$

Proof. Indeed, for any $k \geq 0$ we have

$$\begin{array}{rcl} F(x_k) - F^* & \geq & F(x_k) - F(x_{k+1}) \\ & \overset{(2.0.1)}{\geq} & \langle F'(x_{k+1}), x_k - x_{k+1} \rangle + \frac{\sigma_q}{q} \|x_k - x_{k+1}\|^q \\ & \overset{(2.2.8)}{\geq} & \frac{\sigma_q}{q} \|x_k - x_{k+1}\|^q \overset{(2.2.7)}{\geq} \frac{\sigma_q}{q} \left(\frac{p!}{L_p+H} \|F'(x_{k+1})\|_* \right)^{\frac{q}{p}} \\ & \overset{(2.0.3)}{\geq} & \frac{\sigma_q}{q} \left(\frac{p!}{L_p+H} \right)^{\frac{q}{p}} \left(\frac{q\,\sigma_q^{\frac{1}{q-1}}}{q-1} (F(x_{k+1}) - F^*) \right)^{\frac{q-1}{p}}. \end{array}$$

And this is exactly inequality (2.2.13). □

Corollary 2.2.4. *If $p > q - 1$, then method (2.2.11) has local superlinear rate of convergence.*

Proof. Indeed, in this case $\frac{p}{q-1} > 1$. □

For example, if $q = 2$ (strongly convex function) and $p = 2$ (Cubic Regularization of the Newton Method), then the rate of convergence is quadratic. If $q = 2$, and $p = 3$, then the local rate of convergence is cubic, etc.

Let us study now the local convergence of the method (2.2.11) in terms of the norm of gradient. For any $x \in \operatorname{dom} \psi$ denote

$$\eta(x) \overset{\text{def}}{=} \min_{g \in \partial\psi(x)} \|\nabla f(x) + g\|_*. \tag{2.2.14}$$

If $\partial\psi(x) = \varnothing$, we set $\eta(x) = +\infty$.

Theorem 2.2.5. *Let $\sigma_q > 0$ for some $q \geq 2$. Then for the sequence $\{x_k\}_{k\geq 0}$ generated by method* (2.2.11) *with* $H := pL_p$*, we have*

$$\eta(x_{k+1}) \quad \leq \quad \|F'(x_{k+1})\|_* \leq \frac{L_p+H}{p!} \left[\frac{1}{\sigma_q}\, \eta(x_k)\right]^{\frac{p}{q-1}}. \tag{2.2.15}$$

Proof. Indeed, in view of inequality (2.2.12), we have

$$\begin{aligned} &\langle \nabla f(x_k) + g_k, x_k - x_{k+1} \rangle \\ &\geq \quad \langle F'(x_{k+1}), x_k - x_{k+1} \rangle + \sigma_q \| x_k - x_{k+1} \|^q \\ &\overset{(2.2.8)}{\geq} \quad \sigma_q \| x_k - x_{k+1} \|^q, \end{aligned}$$

where g_k is an arbitrary vector from $\partial \psi(x_k)$. Therefore, we conclude that

$$\eta(x_k) \quad \geq \quad \sigma_q \| x_k - x_{k+1} \|^{q-1}.$$

It remains to use inequality (2.2.7). □

As we can see, the condition for superlinear convergence of the method (2.2.11) in terms of the norm of the gradient is the same as in Corollary 2.2.4: we need to have $\frac{p}{q-1} > 1$, that is $p > q-1$. Moreover, the local rate of convergence has the same order as that for the residual of the function value.

According to Theorem 2.2.3, the region of superlinear convergence of RCTM in terms of the function value is as follows:

$$\mathcal{Q} \quad = \quad \left\{ x: \; F(x) - F^* \; \leq \; \tfrac{1}{q} \cdot \left(\tfrac{\sigma_q^{p+1}}{(q-1)^{q-1}} \cdot \left(\tfrac{p!}{L_p + H} \right)^q \right)^{\frac{1}{p-q+1}} \right\}. \qquad (2.2.16)$$

Alternatively, by Theorem 2.2.5, in terms of the norm of minimal subgradient (2.2.14), the region of superlinear convergence looks as follows:

$$\mathcal{G} \quad = \quad \left\{ x: \; \eta(x) \; \leq \; \left(\sigma_q^p \cdot \left(\tfrac{p!}{L_p + H} \right)^{q-1} \right)^{\frac{1}{p-q+1}} \right\}. \qquad (2.2.17)$$

Note that these sets can be very different. Indeed, set $\mathcal{Q}$ is a closed and convex neighborhood of the point x^*. At the same time, the structure of the set $\mathcal{G}$ can be very complex since in general the function $\eta(x)$ is discontinuous. Let us look at simple example where $\psi(x) = \mathrm{Ind}_Q(x)$, the indicator function of a closed convex set Q.

Example 2.2.6. Consider the following optimization problem:

$$\min_{x \in \mathbb{R}^2} \left\{ f(x) : \; \|x\|^2 \overset{\text{def}}{=} (x^{(1)})^2 + (x^{(2)})^2 \leq 1 \right\}, \qquad (2.2.18)$$

with

$$f(x) \quad = \quad \tfrac{\sigma_2}{2}\|x-\bar{x}\|^2 + \tfrac{2\sigma_3}{3}\|x-\bar{x}\|^3,$$

for some fixed $\sigma_2, \sigma_3 > 0$ and $\bar{x} = (0, -2) \in \mathbb{R}^2$.

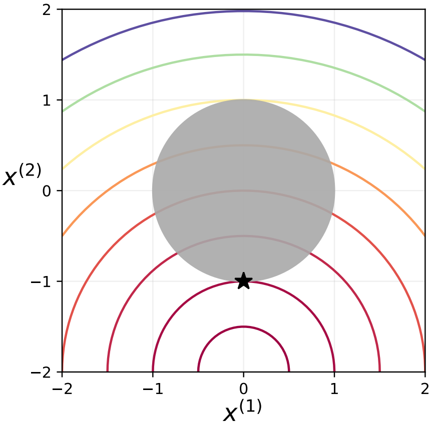


Figure 2.1: Level sets for constrained optimization problem (2.2.18).

We have

$$\nabla f(x) = r(x) \cdot (x^{(1)}, x^{(2)} + 2),$$

where $r : \mathbb{R}^2 \to \mathbb{R}$ is

$$r(x) = \sigma_2 + 2\sigma_3\|x-\bar{x}\|.$$

Note that f is uniformly convex of degree $q = 2$ with constant σ_2, and for $q = 3$ with constant σ_3 (see Lemma 4.2.3 in [117]). Moreover, we have for any $\nu \in [0, 1]$:

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y), x - y \rangle \quad &\geq \quad \sigma_2\|x-y\|^2 + \sigma_3\|x-y\|^3 \\ &\geq \quad \min_{t \geq 0}\Big\{ \tfrac{\sigma_2}{t^\nu} + \sigma_3 t^{1-\nu} \Big\} \cdot \|x-y\|^{2+\nu} \\ &\geq \quad \sigma_2^{1-\nu}\sigma_3^{\nu} \cdot \|x-y\|^{2+\nu}. \end{aligned}$$

Hence, the function is uniformly convex of any degree $q \in [2, 3]$ (note that it also follows from inequality (2.1.2), which was justified by a more general reasoning). At the same time, the Hessian of f is Lipschitz continuous with constant $L_2 = 4\sigma_3$ (see Lemma 4.2.4 in [117]).

Clearly, in this problem $x^* = (0, -1)$, and it can be written in the composite form with

$$\psi(x) \quad = \quad \begin{cases} +\infty, & \text{if } \|x\| > 1, \\ 0, & \text{otherwise.} \end{cases}$$

Note that for $x \in \operatorname{dom} \psi \equiv \{x : \; \|x\| \leq 1\}$, we have

$$\partial\psi(x) \; = \; \begin{cases} 0, & \text{if } \|x\| < 1, \\ \{\gamma x, \, \gamma \geq 0\}, & \text{if } \|x\| = 1. \end{cases}$$

Therefore, if $\|x\| < 1$, then $\eta(x) = \|\nabla f(x)\| \geq \sigma_2$. If $\|x\| = 1$, then

$$\begin{aligned} \eta^2(x) \quad & \overset{(2.2.14)}{=} \quad \min_{\gamma \geq 0} \Big\{ \big[(r(x) + \gamma)x^{(1)}\big]^2 + \big[(r(x) + \gamma)x^{(2)} + 2r(x)\big]^2 \Big\} \\ & = \quad \min_{\gamma \geq 0} \Big\{ (r(x) + \gamma)^2 + 4r(x)(r(x) + \gamma)x^{(2)} + 4r^2(x) \Big\} \\ & = \quad \begin{cases} 4r^2(x)(1 - (x^{(2)})^2), & \text{if } x^{(2)} \leq -\frac{1}{2}, \\ r^2(x)(5 + 4x^{(2)}), & \text{otherwise.} \end{cases} \end{aligned}$$

Thus, in any neighbourhood of x^*, $\eta(x)$ vanishes only along the boundary of the feasible set. □

So, a natural question that arises is how Tensor Method (2.2.11) could come to the region $\mathcal{G}$. The answer follows from the inequalities derived in Section 2.2.1. Indeed,

$$\|F'(x_{k+1})\|_* \quad \overset{(2.2.7)}{\leq} \quad \tfrac{L_p + H}{p!} \|x_k - x_{k+1}\|^p,$$

and

$$\begin{aligned} F(x_k) - F(x_{k+1}) \quad & \geq \quad \langle F'(x_{k+1}), x_k - x_{k+1} \rangle \\ & \overset{(2.2.9)}{\geq} \quad \Big(\tfrac{p!}{(p+1)L_p} \Big)^{\frac{1}{p}} \cdot \|F'(x_{k+1})\|_*^{\frac{p+1}{p}}. \end{aligned}$$

Thus, at some moment the norm $\|F'(x_k)\|_*$ will be small enough to enter $\mathcal{G}$.

2.2.3 Global Complexity Bounds

Let us briefly discuss the global complexity bounds of method (2.2.11), namely the number of iterations required for coming from an arbitrary initial point $x_0 \in \operatorname{dom} \psi$ to the region $\mathcal{Q}$.

First, note that for every step $T = T_H(x)$ of the method with parameter $H \geq pL_p$, we have

$$\begin{aligned} F(T) \quad &\overset{(1.3.5)}{\leq} \quad \Omega_p(f,x;T) + \tfrac{H}{(p+1)!}\|T-x\|^{p+1} + \psi(T) \\ &\overset{(2.2.1)}{=} \quad \min_{y\in\mathbb{E}}\Big\{\Omega_p(f,x;y) + \tfrac{H}{(p+1)!}\|y-x\|^{p+1} + \psi(y)\Big\} \\ &\overset{(1.3.5)}{\leq} \quad \min_{y\in\mathbb{E}}\Big\{F(y) + \tfrac{H+L_p}{(p+1)!}\|y-x\|^{p+1}\Big\}. \end{aligned}$$

Therefore,

$$F(T(x)) - F^* \quad \leq \quad \tfrac{H+L_p}{(p+1)!}\|x-x^*\|^{p+1}, \qquad \forall x \in \operatorname{dom}\psi, \tag{2.2.19}$$

with $x^* \overset{\text{def}}{=} \operatorname{argmin}_y F(y)$, which exists by our assumption. Denote by D_0 the maximal radius of the initial level set of the objective, which we assume to be finite:

$$D_0 \quad \overset{\text{def}}{=} \quad \sup_{x\in\operatorname{dom}\psi}\Big\{\|x-x^*\| : \; F(x) \leq F(x_0)\Big\} \quad < \quad +\infty.$$

Then, by monotonicity of method (2.2.11) and by convexity we conclude that

$$\frac{1}{D_0}\Big(F(x_{k+1}) - F^*\Big) \;\leq\; \frac{1}{D_0}\langle F'(x_{k+1}), x_{k+1} - x^*\rangle \;\leq\; \|F'(x_{k+1})\|_*. \tag{2.2.20}$$

In the general convex case for the Tensor Method, we can prove the global sublinear rate of convergence of the order $\mathcal{O}(1/k^p)$ [118]. For completeness of presentation, let us prove an extension of this result onto the composite case. Note that the theorems from this section are valid for $p = 1$ (the Gradient Method) as well.

Theorem 2.2.7. *For method* (2.2.11) *with* $H := pL_p$ *we have*

$$F(x_k) - F^* \quad \leq \quad \tfrac{(p+1)(2p)^p}{p!} \cdot \tfrac{L_p D_0^{p+1}}{(k-1)^p}, \qquad k \geq 2. \tag{2.2.21}$$

Proof. Indeed, in view of (2.2.9) and (2.2.20), we have, for every $k \geq 0$, that

$$\begin{aligned} F(x_k) - F(x_{k+1}) &\geq \langle F'(x_{k+1}), x_k - x_{k+1} \rangle \\ &\overset{(2.2.9)}{\geq} \left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{1}{p}} \cdot \|F'(x_{k+1})\|_*^{\frac{p+1}{p}} \\ &\overset{(2.2.20)}{\geq} \left(\tfrac{p!}{(p+1)L_p D_0^{p+1}}\right)^{\frac{1}{p}} \cdot \Big(F(x_{k+1}) - F^*\Big)^{\frac{p+1}{p}}. \end{aligned}$$

Denoting $\delta_k = F(x_k) - F^*$ and $C = \left(\frac{p!}{(p+1)L_p D_0^{p+1}}\right)^{\frac{1}{p}}$, we obtain the following recurrence:

$$\delta_k - \delta_{k+1} \;\geq\; C\delta_{k+1}^{\frac{p+1}{p}}, \qquad k \geq 0, \tag{2.2.22}$$

or for $\mu_k = C^p \delta_k \overset{(2.2.19)}{\leq} 1$, as follows:

$$\mu_k - \mu_{k+1} \;\geq\; \mu_{k+1}^{\frac{p+1}{p}}, \qquad k \geq 0.$$

Then, Lemma 1.1 from [60] provides us with the following guarantee:

$$\mu_k \;\leq\; \left(\tfrac{p(1+\mu_1^{1/p})}{k-1}\right)^p \;\leq\; \left(\tfrac{2p}{k-1}\right)^p, \quad k \geq 2.$$

Therefore,

$$\delta_k \;=\; \tfrac{\mu_k}{C^p} \;\leq\; \left(\tfrac{2p}{C(k-1)}\right)^p \;=\; \tfrac{(p+1)(2p)^p}{p!} \cdot \tfrac{L_p D_0^{p+1}}{(k-1)^p}, \qquad k \geq 2.$$

□

For a given degree $q \geq 2$ of uniform convexity with $\sigma_q > 0$, and for RCTM of order $p \geq q-1$, let us denote by $\bar{\omega}_{p,q}$ the following *condition number*:

$$\bar{\omega}_{p,q} \;\overset{\text{def}}{=}\; \tfrac{p+1}{p!} \cdot \left(\tfrac{q-1}{q}\right)^{q-1} \cdot \tfrac{L_p D_0^{p-q+1}}{\sigma_q}.$$

Then, we come to the following conclusion.

Corollary 2.2.8. *In order to attain the region $\mathcal{Q}$ it is enough to perform*

$$\left\lceil 2p \cdot \left(\frac{q^q}{(q-1)^{q-1}} \cdot \bar{\omega}_{p,q}^{\frac{p+1}{p}} \right)^{\frac{1}{p-q+1}} \right\rceil + 2 \tag{2.2.23}$$

iterations of the method.

Proof. Plugging (2.2.16) into (2.2.21). □

We can improve this estimate, knowing that the objective is globally uniformly convex (2.2.12). Then the linear rate of convergence arises at the first state, till the entering in the region $\mathcal{Q}$.

Theorem 2.2.9. *Let $\sigma_q > 0$ with $q \le p+1$. Then for method (2.2.11) with $H := pL_p$, we have*

$$F(x_k) - F^* \quad \le \quad \exp\left(-\frac{k}{1+\bar{\omega}_{p,q}^{1/p}}\right) \cdot \bigl(F(x_0) - F^*\bigr), \qquad k \ge 1. \qquad (2.2.24)$$

Therefore, for a given $\varepsilon > 0$ to achieve $F(x_K) - F^* \le \varepsilon$, it is enough to set

$$K \quad = \quad \left\lceil (1 + \bar{\omega}_{p,q}^{1/p}) \cdot \log \frac{F(x_0)-F^*}{\varepsilon} \right\rceil + 1. \qquad (2.2.25)$$

Proof. Indeed, for every $k \ge 0$

$$\begin{array}{rcl} F(x_k) - F(x_{k+1}) & & \\ & \ge & \langle F'(x_{k+1}), x_k - x_{k+1} \rangle \\ & \overset{(2.2.9)}{\ge} & \left(\frac{p!}{(p+1)L_p}\right)^{\frac{1}{p}} \cdot \|F'(x_{k+1})\|_*^{\frac{p+1}{p}} \\ & = & \left(\frac{p!}{(p+1)L_p}\right)^{\frac{1}{p}} \cdot \|F'(x_{k+1})\|_*^{\frac{p-q+1}{p}} \cdot \|F'(x_{k+1})\|_*^{\frac{q}{p}} \\ & \overset{(2.2.20),(2.0.3)}{\ge} & \left(\frac{p!}{p+1} \cdot \frac{\sigma_q}{L_p D_0^{p-q+1}}\right)^{\frac{1}{p}} \cdot \left(\frac{q}{q-1}\right)^{\frac{q-1}{p}} \cdot \Bigl(F(x_{k+1}) - F^*\Bigr) \\ & = & \left(\frac{1}{\bar{\omega}_{p,q}}\right)^{\frac{1}{p}} \cdot \Bigl(F(x_{k+1}) - F^*\Bigr). \end{array}$$

Denoting $\delta_k = F(x_k) - F^*$, we obtain

$$\delta_{k+1} \quad \le \quad \frac{\bar{\omega}_{p,q}^{1/p}}{1+\bar{\omega}_{p,q}^{1/p}} \cdot \delta_k \;\le\; \exp\left(-\frac{1}{1+\bar{\omega}_{p,q}^{1/p}}\right) \cdot \delta_k, \qquad k \ge 1. \qquad \square$$

We see that, for RCTM with $p \ge 2$ minimizing the uniformly convex objective of degree $q \le p+1$, the condition number $\bar{\omega}_{p,q}^{1/p}$ is the main factor in the global complexity estimates (2.2.23) and (2.2.25). Since in general this number may be arbitrarily big, complexity estimate $\tilde{\mathcal{O}}(\bar{\omega}_{p,q}^{1/p})$ in (2.2.25) is much better than the estimate $\mathcal{O}(\bar{\omega}_{p,q}^{(p+1)/(p(p-q+1))})$ in (2.2.23) because of relation $\frac{p+1}{p-q+1} \geq 1$.

These global bounds can be improved, by using the *universal* (see the previous Section 2.1) and the *accelerated* [111, 61, 63, 54, 145] high-order schemes.

High-order tensor methods for minimizing the gradient norm have been developed in [13, 48]. These methods achieve near-optimal global convergence rates, and can be used for coming into the region $\mathcal{G}$ (2.2.17). Note that for the composite minimization problems, some modification of these methods is required, which ensures minimization of the *subgradient* norm.

Finally, let us mention recent results [122, 76], where it was shown that it is possible to implement the third-order schemes by using only second-order oracle information (see also Section 1.5.2). Hence, it is interesting to investigate the local behaviour of the high-order methods under approximate condition on the derivatives, which we keep as an open question for the further research.

2.2.4 Application to Proximal Methods

Let us discuss now a general approach, which uses the local convergence of the methods for justifying the global performance of proximal iterations.

The Proximal-Point algorithm [133] is one of the classical iterative methods in theoretical optimization. This method, as applied to minimizing a convex function $F : \operatorname{dom} F \to \mathbb{R}$, consists of solving at each iteration the following subproblem:

$$x_{k+1} \quad = \quad \operatorname*{argmin}_{x} \Bigl\{ a_{k+1} F(x) + \tfrac{1}{2}\|x - x_k\|^2 \Bigr\}, \qquad k \geq 0, \tag{2.2.26}$$

where $\{a_k\}_{k\geq 1}$ is a sequence of positive coefficients, related to the iteration counter.

Of course, in general, we can hope only to use an inexact solution to the subproblem (2.2.26). The questions of practical implementations and possible generalizations of the proximal method, are still in the area of intensive research (see, for example [65, 144, 142, 140]).

One simple observation on the subproblem (2.2.26) is that it is 1-strongly convex. Therefore, if we were be able to pick an initial point from the region of superlinear convergence (2.2.16) or (2.2.17), we could minimize it very quickly by RCTM of degree $p \geq 2$ up to arbitrary accuracy. In this section, we are going to investigate this approach. For the resulting scheme, we will prove the global rate of convergence of order $\tilde{\mathcal{O}}(1/k^{\frac{p+1}{2}})$.

Denote by Φ_{k+1} the regularized objective from (2.2.26):

$$\begin{array}{rcl}\Phi_{k+1}(x) & \overset{\text{def}}{=} & a_{k+1}F(x) + \frac{1}{2}\|x - x_k\|^2 \\ \\ & = & a_{k+1}f(x) + \frac{1}{2}\|x - x_k\|^2 + a_{k+1}\psi(x).\end{array}$$

We fix a sequences of accuracies $\{\delta_k\}_{k\geq 1}$ and relax the assumption on exact minimization in (2.2.26). Now, at every step we need to find a point x_{k+1} and corresponding subgradient vector $g_{k+1} \in \partial\Phi_{k+1}(x_{k+1})$ with bounded norm:

$$\|g_{k+1}\|_* \quad \leq \quad \delta_{k+1}. \tag{2.2.27}$$

Denote

$$F'(x_{k+1}) \quad \overset{\text{def}}{=} \quad \tfrac{1}{a_{k+1}}(g_{k+1} - B(x_{k+1} - x_k)) \;\in\; \partial F(x_{k+1}).$$

The following global convergence result holds for the general proximal algorithm with inexact minimization criterion (2.2.27).

Theorem 2.2.10. *Assume that there exist a minimum $x^* \in \operatorname{dom}\psi$ of the problem. Then, for any $k \geq 1$, we have*

$$\begin{array}{l}\sum\limits_{i=1}^{k} a_i(F(x_i) - F^*) + \frac{1}{2}\sum\limits_{i=1}^{k} a_i^2\|F'(x_i)\|_*^2 + \frac{1}{2}\|x_k - x^*\|^2 \\ \\ \leq \quad R_k(\delta) \quad \overset{\text{def}}{=} \quad \frac{1}{2}\left(\|x_0 - x^*\| + \sum\limits_{i=1}^{k}\delta_i\right)^2.\end{array} \tag{2.2.28}$$

Proof. First, let us prove that for all $k \geq 0$ and for every $x \in \operatorname{dom}\psi$, we have

$$\tfrac{1}{2}\|x_0 - x\|^2 + \sum_{i=1}^{k} a_i F(x) \quad \geq \quad \tfrac{1}{2}\|x_k - x\|^2 + C_k(x), \tag{2.2.29}$$

where

$$C_k(x) \quad \overset{\text{def}}{=} \quad \sum_{i=1}^{k}\left(a_i F(x_i) + \tfrac{a_i^2}{2}\|F'(x_i)\|_*^2 + \langle g_i, x - x_{i-1}\rangle - \tfrac{\delta_i^2}{2}\right).$$

This is obviously true for $k = 0$. Let it hold for some $k \geq 0$. Consider the step number $k+1$ of the inexact proximal method.

By condition (2.2.27), we have

$$\| a_{k+1} F'(x_{k+1}) + B(x_{k+1} - x_k) \|_*^2 \quad \leq \quad \delta_{k+1}^2.$$

Equivalently,

$$\begin{array}{rl} & \langle a_{k+1} F'(x_{k+1}), x_k - x_{k+1} \rangle \\ \geq & \frac{a_{k+1}^2}{2} \| F'(x_{k+1}) \|_*^2 + \frac{1}{2} \| x_{k+1} - x_k \|^2 - \frac{\delta_{k+1}^2}{2}. \end{array} \tag{2.2.30}$$

Therefore, by using the inductive assumption and strong convexity of $\Phi_{k+1}(\cdot)$, we conclude

$$\begin{array}{rl} \frac{1}{2}\|x_0 - x\|^2 + \sum\limits_{i=1}^{k+1} a_i F(x) & = \ \frac{1}{2}\|x_0 - x\|^2 + \sum\limits_{i=1}^{k} a_i F(x) + a_{k+1} F(x) \\ \overset{(2.2.29)}{\geq} & \Phi_{k+1}(x) + C_k(x) \\ \geq & \Phi_{k+1}(x_{k+1}) + \langle g_{k+1}, x - x_{k+1} \rangle + \frac{1}{2}\|x_{k+1} - x\|^2 + C_k(x) \\ = & a_{k+1} F(x_{k+1}) + \frac{1}{2}\|x_{k+1} - x_k\|^2 + \langle g_{k+1}, x_k - x_{k+1} \rangle \\ + & \langle g_{k+1}, x - x_k \rangle + \frac{1}{2}\|x_{k+1} - x\|^2 + C_k(x) \\ = & a_{k+1} F(x_{k+1}) + \langle a_{k+1} F'(x_{k+1}), x_k - x_{k+1} \rangle - \frac{1}{2}\|x_{k+1} - x_k\|^2 \\ + & \langle g_{k+1}, x - x_k \rangle + \frac{1}{2}\|x_{k+1} - x\|^2 + C_k(x) \\ \overset{(2.2.30)}{\geq} & a_{k+1} F(x_{k+1}) + \frac{a_{k+1}^2}{2}\|F'(x_{k+1})\|_*^2 - \frac{\delta_{k+1}^2}{2} \\ + & \langle g_{k+1}, x - x_k \rangle + \frac{1}{2}\|x_{k+1} - x\|^2 + C_k(x) \\ = & \frac{1}{2}\|x_{k+1} - x\|^2 + C_{k+1}(x). \end{array}$$

Thus, inequality (2.2.29) is valid for all $k \geq 0$.

Now, by plugging $x \equiv x^*$ into (2.2.29), we have

$$\begin{array}{rl} \sum\limits_{i=1}^{k} a_i(F(x_i) - F^*) + \frac{1}{2}\sum\limits_{i=1}^{k} a_i^2 \|F'(x_i)\|_*^2 + \frac{1}{2}\|x_k - x^*\|^2 & \\ \leq & \frac{1}{2}\|x_0 - x^*\|^2 + \frac{1}{2}\sum\limits_{i=1}^{k}\delta_i^2 + \sum\limits_{i=1}^{k}\langle g_i, x_{i-1} - x^*\rangle \\ \overset{(2.2.27)}{\leq} & \frac{1}{2}\|x_0 - x^*\|^2 + \frac{1}{2}\sum\limits_{i=1}^{k}\delta_i^2 + \sum\limits_{i=1}^{k}\delta_i\|x_{i-1} - x^*\| \\ \overset{\text{def}}{=} & \alpha_k. \end{array} \tag{2.2.31}$$

In order to finish the proof, it is enough to show that $\alpha_k \leq R_k(\delta)$.

Indeed,

$$\begin{array}{rcl} \alpha_{k+1} & = & \alpha_k + \frac{1}{2}\delta_{k+1}^2 + \delta_{k+1}\|x_k - x^*\| \\ & \overset{(2.2.31)}{\leq} & \alpha_k + \frac{1}{2}\delta_{k+1}^2 + \delta_{k+1}\sqrt{2\alpha_k} \\ & = & \left(\sqrt{\alpha_k} + \frac{1}{\sqrt{2}}\delta_{k+1}\right)^2. \end{array}$$

Therefore,

$$\begin{array}{rcl} \sqrt{\alpha_k} & \leq & \sqrt{\alpha_{k-1}} + \frac{1}{\sqrt{2}}\delta_k \;\leq\; \ldots \;\leq\; \sqrt{\alpha_0} + \frac{1}{\sqrt{2}}\sum\limits_{i=1}^{k}\delta_i \\ & = & \frac{1}{\sqrt{2}}\left(\|x_0 - x^*\| + \sum\limits_{i=1}^{k}\delta_i\right) \;=\; \sqrt{R_k(\delta)}. \qquad \Box \end{array}$$

Now, we are ready to use the result on the local superlinear convergence of RCTM in the norm of subgradient (Theorem 2.2.5), in order to minimize $\Phi_{k+1}(\cdot)$ at every step of inexact proximal method.

Note that

$$\partial\Phi_{k+1}(x) \;=\; a_{k+1}\partial F(x) + B(x - x_k),$$

and it is natural to start minimization process from the previous point x_k, for which $\partial\Phi_{k+1}(x_k) = a_{k+1}\partial F(x_k)$. Let us also notice, that the Lipschitz constant of the p-th derivative ($p \geq 2$) of the smooth part of Φ_{k+1} is $a_{k+1}L_p$.

By using our previous notation, one step of RCTM can be written as follows:

$$\begin{aligned} T_H(\Phi_{k+1}, z) \quad &\overset{\text{def}}{=} \quad \arg\min_{y \in \mathbb{E}} \Big\{ a_{k+1}\Omega_p(f, z; y) + \tfrac{H}{(p+1)!}\|y - z\|^{p+1} \\ &\qquad\qquad + a_{k+1}\psi(y) + \tfrac{1}{2}\|y - x_k\|^2 \Big\}, \end{aligned}$$

where $H = a_{k+1} p L_p$. Then, a sufficient condition for $z = x_k$ to be in the region of superlinear convergence (2.2.17) is

$$a_{k+1}\|F'(x_k)\|_* \quad \leq \quad \left(\tfrac{p!}{a_{k+1}(p+1)L_p} \right)^{\frac{1}{p-1}},$$

or, equivalently

$$a_{k+1} \quad \leq \quad \left(\tfrac{1}{\|F'(x_k)\|_*} \right)^{\frac{p-1}{p}} \left(\tfrac{p!}{(p+1)L_p} \right)^{\frac{1}{p}}.$$

To be sure that x_k is strictly inside the region, we can pick:

$$\boxed{a_{k+1} \quad = \quad \left(\tfrac{1}{2\|F'(x_k)\|_*} \right)^{\frac{p-1}{p}} \left(\tfrac{p!}{(p+1)L_p} \right)^{\frac{1}{p}}} \tag{2.2.32}$$

Note, that this rule requires fixing an initial subgradient $F'(x_0) \in \partial F(x_0)$, in order to choose a_1.

Finally, we apply the following steps:

$$\boxed{z_0 \;=\; x_k, \quad z_{t+1} \quad = \quad T_H(\Phi_{k+1}, z_t), \quad t \geq 0} \tag{2.2.33}$$

We can estimate the required number of these iterations as follows.

Lemma 2.2.11. *At every iteration $k \geq 0$ of the inexact proximal method, in order to achieve $\|\Phi'_{k+1}(z_t)\|_* \leq \delta_{k+1}$, it is enough to perform*

$$t_k \quad = \quad \left\lceil \tfrac{1}{\log_2 p} \cdot \log_2 \log_2 \left(\tfrac{2D_k(\delta)}{\delta_{k+1}} \right) \right\rceil \tag{2.2.34}$$

steps of RCTM (2.2.33), where

$$D_k(\delta) \quad \overset{\text{def}}{=} \quad \max\Big\{ \|x_0 - x^*\| + \sum_{i=1}^{k} \delta_i, \left(\tfrac{p!\|F'(x_0)\|_*}{(p+1)L_p 2^{p-1}} \right)^{\frac{1}{p}} \Big\}$$

Proof. According to (2.2.15), one step of RCTM (2.2.33) provides us with the following guarantee in terms of the subgradients of our objective $\Phi_{k+1}(\cdot)$:

$$\|\Phi'_{k+1}(z_t)\|_* \quad \leq \quad \tfrac{a_{k+1}(p+1)L_p}{p!}\|\Phi'_{k+1}(z_{t-1})\|_*^p, \tag{2.2.35}$$

where we used in (2.2.15) the values $q = 2$, $\sigma_q = 1$, $a_{k+1}L_p$ for the Lipschitz constant of the p-th derivative of the smooth part of Φ_{k+1}, and $H = a_{k+1}pL_p$.

Denote $\beta \equiv \left(\frac{a_{k+1}(p+1)L_p}{p!}\right)^{\frac{1}{p-1}} \overset{(2.2.32)}{=} \left(\frac{(p+1)L_p}{2\cdot p!\cdot\|F'(x_k)\|_*}\right)^{\frac{1}{p}}$. Then, from (2.2.35) we have

$$\begin{aligned} \beta\|\Phi'_{k+1}(z_t)\|_* \quad & \leq \quad \left(\beta\|\Phi'_{k+1}(z_{t-1})\|_*\right)^p \\ & \leq \quad \ldots \quad \leq \quad \left(\beta\|\Phi'_{k+1}(z_0)\|_*\right)^{p^t} \\ & = \quad (\beta a_{k+1}\|F'(x_k)\|_*)^{p^t} \\ & = \quad \left(a_{k+1}^{\frac{p}{p-1}}\left(\tfrac{(p+1)L_p}{p!}\right)^{\frac{1}{p-1}}\|F'(x_k)\|_*\right)^{p^t} \\ & \overset{(2.2.32)}{=} \quad \left(\tfrac{1}{2}\right)^{p^t}. \end{aligned} \tag{2.2.36}$$

Therefore, for

$$\begin{aligned} t \quad & \geq \quad \log_p \log_2\left(\tfrac{1}{\beta\delta_{k+1}}\right) \\ & = \quad \tfrac{1}{\log_2 p}\cdot\log_2\log_2\left(\tfrac{1}{\delta_{k+1}}\left(\tfrac{2\cdot p!\cdot\|F'(x_k)\|_*}{(p+1)L_p}\right)^{\frac{1}{p}}\right), \end{aligned} \tag{2.2.37}$$

it holds $\|\Phi'_{k+1}(z_t)\|_* \leq \delta_{k+1}$. To finish the proof, let us estimate $\|F'(x_k)\|_*$ from above. We have

$$\begin{aligned} & 2^{\frac{3p-2}{p}}\left(\tfrac{(p+1)L_p}{p!}\right)^{\frac{2}{p}} R_k(\delta) \\ & \overset{(2.2.28)}{\geq} \quad 2^{\frac{2(p-1)}{p}}\left(\tfrac{(p+1)L_p}{p!}\right)^{\frac{2}{p}}\sum_{i=1}^{k} a_i^2\|F'(x_i)\|_*^2 \\ & \overset{(2.2.32)}{=} \quad \sum_{i=1}^{k}\|F'(x_{i-1})\|_*^{\frac{2(1-p)}{p}}\|F'(x_i)\|_*^2. \end{aligned} \tag{2.2.38}$$

Thus, for every $1 \leq i \leq k$ it holds

$$\|F'(x_i)\|_* \overset{(2.2.38)}{\leq} \|F'(x_{i-1})\|_*^{\rho} \cdot \mathcal{D}, \tag{2.2.39}$$

with $\mathcal{D} \equiv R_k^{1/2}(\delta)\left(\frac{(p+1)L_p}{p!}\right)^{\frac{1}{p}} 2^{\frac{3p-2}{2p}}$, and $\rho \equiv \frac{p-1}{p}$. Therefore,

$$\begin{array}{rcl}
\|F'(x_k)\|_* & \overset{(2.2.39)}{\leq} & \|F'(x_0)\|_*^{\rho^k} \cdot \mathcal{D}^{1+\rho+\rho^2+\cdots+\rho^{k-1}} \\[2ex]
& = & \|F'(x_0)\|_* \cdot \left(\|F'(x_0)\|_*^{\rho^k-1} \cdot \mathcal{D}^{\frac{1-\rho^k}{1-\rho}}\right) \\[2ex]
& = & \|F'(x_0)\|_* \cdot \left(\frac{\mathcal{D}^p}{\|F'(x_0)\|_*}\right)^{1-\rho^k} \\[2ex]
& \leq & \|F'(x_0)\|_* \cdot \max\{\frac{\mathcal{D}^p}{\|F'(x_0)\|_*}, 1\} \\[2ex]
& = & \max\left\{\frac{(p+1)L_p 2^{p-1}}{p!}\left(\|x_0 - x^*\| + \sum\limits_{i=1}^{k}\delta_i\right)^p, \; \|F'(x_0)\|_*\right\}.
\end{array}$$

Substitution of this bound into (2.2.37) gives (2.2.34). □

Let us prove now the rate of convergence for the outer iterations. This is a direct consequence of Theorem 2.2.10 and the choice (2.2.32) of the coefficients $\{a_k\}_{k\geq 1}$.

Lemma 2.2.12. *Let for a given $\varepsilon > 0$,*

$$F(x_k) - F^* \quad \geq \quad \varepsilon, \qquad 1 \leq k \leq K. \tag{2.2.40}$$

Then for every $1 \leq k \leq K$, we have

$$F(\bar{x}_k) - F^* \quad \leq \quad \frac{L_p\left(\|x_0-x^*\|+\sum_{i=1}^k \delta_i\right)^{p+1}}{k^{\frac{p+1}{2}}} \frac{(p+1)2^{p-2}V_k(\varepsilon)}{p!}, \tag{2.2.41}$$

where $\bar{x}_k \overset{\text{def}}{=} \frac{\sum_{i=1}^k a_i x_i}{\sum_{i=1}^k a_i}$, *and* $V_k(\varepsilon) \overset{\text{def}}{=} \left(\frac{\|F'(x_0)\|_* \cdot (\|x_0-x^*\|+\sum_{i=1}^k \delta_i)}{\varepsilon}\right)^{\frac{p-1}{k}}$.

Proof. By using the inequality between the arithmetic and geometric means, we obtain

$$\begin{aligned} R_k(\delta) & \overset{(2.2.28)}{\geq} \tfrac{1}{2}\sum_{i=1}^{k} a_i^2 \|F'(x_i)\|_*^2 \\ & \overset{(2.2.32)}{=} \tfrac{1}{8}\left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{2}{p-1}} \sum_{i=1}^{k} \frac{a_i^2}{a_{i+1}^{\frac{2p}{p-1}}} \\ & \geq \tfrac{k}{8}\left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{2}{p-1}} \left(\prod_{i=1}^{k} \frac{a_i^2}{a_{i+1}^{\frac{2p}{p-1}}}\right)^{\frac{1}{k}} \qquad (2.2.42) \\ & = \tfrac{k}{8}\left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{2}{p-1}} \left(\tfrac{a_1}{a_{k+1}}\right)^{\frac{2p}{(p-1)k}} \left(\prod_{i=1}^{k} a_i\right)^{\frac{-2}{(p-1)k}} \\ & \geq \frac{k^{\frac{p+1}{p-1}}}{8}\left(\tfrac{p!}{(p+1)L_p}\right)^{\frac{2}{p-1}} \left(\tfrac{a_1}{a_{k+1}}\right)^{\frac{2p}{(p-1)k}} \left(\sum_{i=1}^{k} a_i\right)^{\frac{-2}{p-1}}. \end{aligned}$$

Therefore,

$$\begin{aligned} F(\bar{x}_k) - F^* & \leq \frac{1}{\sum\limits_{i=1}^{k} a_i} \sum_{i=1}^{k} a_i (F(x_i) - F^*) \overset{(2.2.28)}{\leq} \frac{R_k(\delta)}{\sum\limits_{i=1}^{k} a_i} \\ & \overset{(2.2.42)}{\leq} \frac{R_k(\delta)^{\frac{p+1}{2}}}{k^{\frac{p+1}{2}}} \frac{(p+1)L_p}{p!} \left(\frac{a_{k+1}}{a_1}\right)^{\frac{p}{k}} 8^{\frac{p-1}{2}} \\ & = \frac{L_p\left(\|x_0-x^*\|+\sum_{i=1}^{k}\delta_i\right)^{p+1}}{k^{\frac{p+1}{2}}} \frac{(p+1)2^{p-2}}{p!} \left(\frac{\|F'(x_0)\|_*}{\|F'(x_k)\|_*}\right)^{\frac{p-1}{k}}, \end{aligned}$$

where the first inequality holds by convexity. At the same time, we have

$$\begin{aligned} \|F'(x_k)\|_* & \geq \frac{\langle F'(x_k), x_k - x^*\rangle}{\|x_k - x^*\|} \geq \frac{F(x_k)-F^*}{\|x_k-x^*\|} \\ & \overset{(2.2.40)}{\geq} \frac{\varepsilon}{\|x_k-x^*\|} \overset{(2.2.28)}{\geq} \frac{\varepsilon}{\|x_0-x^*\|+\sum_{i=1}^{k}\delta_i}. \end{aligned}$$

Thus, $\left(\frac{\|F'(x_0)\|_*}{\|F'(x_k)\|_*}\right)^{\frac{p-1}{k}} \leq V_k(\varepsilon)$ and we obtain (2.2.41). □

Remark 2.2.13. Note that $\left(\frac{1}{\varepsilon}\right)^{\frac{p-1}{k}} = \exp\left(\frac{p-1}{k}\ln\frac{1}{\varepsilon}\right)$. Therefore after $k =$

$\mathcal{O}\left(\ln \frac{1}{\varepsilon}\right)$ iterations, the factor $V_k(\varepsilon)$ is bounded by an absolute constant.

Since the local convergence of RCTM is very fast (2.2.34), we can choose the inner accuracies $\{\delta_i\}_{i\geq 1}$ small enough, to have the right hand side of (2.2.41) being of the order $\tilde{\mathcal{O}}(1/k^{\frac{p+1}{2}})$. Let us present a precise statement.

Theorem 2.2.14. *Let $\delta_k \equiv \frac{c}{k^s}$ for fixed absolute constants $c > 0$ and $s > 1$. Let for a given $\varepsilon > 0$, we have*

$$F(x_k) - F^* \quad \geq \quad \varepsilon, \qquad 1 \leq k \leq K.$$

Then, for every k such that $\ln \frac{\|F'(x_0)\|_* R}{\varepsilon} \leq k \leq K$, we get

$$F(\bar{x}_k) - F^* \quad \leq \quad \tfrac{L_p R^{p+1}}{k^{\frac{p+1}{2}}} \tfrac{(p+1)2^{p-2}\exp(p-1)}{p!}, \tag{2.2.43}$$

where

$$R \quad \stackrel{\mathrm{def}}{=} \quad \|x_0 - x^*\| + \tfrac{cs}{s-1}.$$

The total number of oracle calls N_k during the first k iterations is bounded as follows:

$$N_k \quad \leq \quad k \cdot \Bigl(1 + \tfrac{1}{\log_2 p} \log_2 \log_2 \tfrac{2Dk^s}{c}\Bigr),$$

where

$$D \quad \stackrel{\mathrm{def}}{=} \quad \max\Bigl\{R, \, \Bigl(\tfrac{p!\|F'(x_0)\|_*}{(p+1)L_p 2^{p-1}}\Bigr)^{\frac{1}{p}}\Bigr\}.$$

Proof. First, observe that

$$\sum_{i=1}^{k} \delta_i \quad \overset{(1.3.9)}{\leq} \quad \tfrac{cs}{s-1}.$$

Thus, we obtain (2.2.43) directly from the bound (2.2.41) and by the fact that

$$\begin{aligned} V_k(\varepsilon) \quad &\equiv \quad \Bigl(\tfrac{\|F'(x_0)\|_* R}{\varepsilon}\Bigr)^{\frac{p-1}{k}} \quad = \quad \exp\Bigl(\tfrac{p-1}{k} \log \tfrac{\|F'(x_0)\|_* R}{\varepsilon}\Bigr) \\ &\leq \quad \exp(p-1), \end{aligned}$$

when $k \geq \ln \frac{\|F'(x_0)\|_* R}{\varepsilon}$.

Then,

$$\begin{aligned} N_k & \overset{(2.2.34)}{\leq} \sum_{i=1}^{k} \left\lceil \tfrac{1}{\log_2 p} \log_2 \log_2 \tfrac{2D}{\delta_i} \right\rceil \;\leq\; k + \tfrac{1}{\log_2 p} \sum_{i=1}^{k} \log_2 \log_2 \tfrac{2Di^s}{c} \\ & \leq \quad k + \tfrac{1}{\log_2 p} \sum_{i=1}^{k} \log_2 \log_2 \tfrac{2Dk^s}{c} \;=\; k \cdot \Bigl(1 + \tfrac{1}{\log_2 p} \log_2 \log_2 \tfrac{2Dk^s}{c} \Bigr). \end{aligned}$$

□

Note that we were able to justify the global performance of the scheme by using only the local convergence results for the inner method. It is interesting to compare our approach with the recent results on the path-following second-order methods [49].

We can drop the logarithmic components in the complexity bounds by using the *hybrid proximal methods* (see [100] and [98]), where at each iteration only one step of RCTM is performed. The resulting rate of convergence there is $\mathcal{O}(1/k^{\frac{p+1}{2}})$, without any extra logarithmic factors. However, this rate is worse than the rate $\mathcal{O}(1/k^p)$ provided by the Theorem 2.2.7 for the primal iterations of RCTM (2.2.11).

Chapter 3

Contraction Technique in Convex Optimization

For a differentiable function f, we can consider its *contraction*, which is the function

$$x \quad \mapsto \quad f(\gamma x + (1-\gamma)\bar{x}),$$

where $\bar{x}$ is a fixed point (usually the current iterate of a method), and $\gamma \in (0,1)$ is a contraction parameter.

Let us denote the new function by $g(x) := f(\gamma x + (1-\gamma)\bar{x})$. Then, its derivatives are as follows:

$$\begin{array}{rcl} Dg(x) & = & \gamma Df(\gamma x + (1-\gamma)\bar{x}), \\[2ex] D^2 g(x) & = & \gamma^2 D^2 f(\gamma x + (1-\gamma)\bar{x}), \\ & \ldots & \\ D^k g(x) & = & \gamma^k D^k f(\gamma x + (1-\gamma)\bar{x}). \end{array}$$

The smoothness characteristics of the objective (i.e. the Lipschitz constants) are defined by using the derivatives. Hence, we can hope that the smoothness properties of $g(\cdot)$ can be better than those of the initial function. The result of employing the contracted objective should be combined with the progress made by an optimization algorithm up to the current iterate $\bar{x}$.

In this chapter, we investigate this idea by developing new contracting algorithms for Smooth Convex Optimization. It appears that the contraction technique is somewhat complementary to the proximal approach.

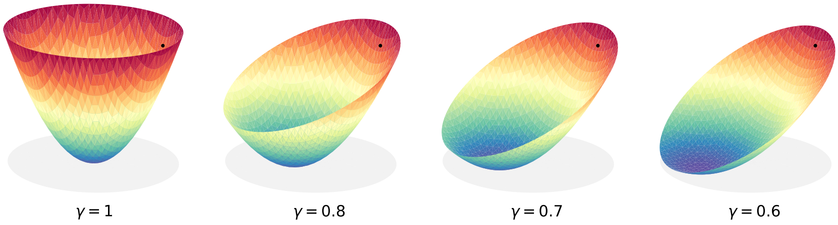


Figure 3.1: Contractions of a convex quadratic function.

3.1 Affine-Invariant Contracting-Point Methods

In the last years, we have seen an increasing interest in new frameworks for derivation and justification different methods for Convex Optimization, provided with a worst-case complexity analysis (see, for example, [7, 95, 28, 116, 118, 54, 77, 122, 120]). It turns out that the accelerated proximal tensor methods [6, 118] can be naturally explained through the framework of high-order proximal-point schemes [120] requiring a solution to nontrivial auxiliary problem at every iteration.

This possibility serves as a departure point for the results presented in this part of the thesis. Indeed, the main drawback of proximal tensor methods consists in the necessity of using a fixed Euclidean structure for measuring distances between points. However, the multi-dimensional Taylor polynomials are defined by directional derivatives, which are affine-invariant objects. Can we construct a family of tensor methods, which do not depend on the choice of the coordinate system in the space of variables? Our results give a positive answer on this question.

Our framework extends the initial results presented in [116], where it was shown that the classical Frank-Wolfe algorithm can be generalized onto the case of the composite objective function [114] by using a contraction of the feasible set towards the current test point. This operation was also used in [116] for justifying a second-order method with contraction, which looks similar to the classical trust-region schemes [30], but with asymmetric trust region.

We significantly improve the convergence rates for the second-order methods, and extend the contraction technique onto the whole family of tensor methods. However, in the vein of [120], we start first from analyzing a conceptual scheme solving at each iteration an auxiliary optimization problem formulated in terms of the initial objective function.

In Section 3.1.1, we present a general framework of Contracting-Point methods. We provide two conceptual variants of our scheme for different conditions of inexactness for the solution to the subproblem: using a point with small residual in the function value, and another one using a stronger condition which involves the gradients. For both schemes we establish global bounds for the functional residual of the initial problem. These bounds lead to global convergence guarantees under a suitable choice of the parameters. For the scheme with the second condition of inexactness, we also provide a computable accuracy certificate. It can be used to estimate the functional residual directly within the method.

Section 3.1.2 contains smoothness conditions, which are useful to analyse affine-invariant high-order schemes. We present some basic inequalities and examples, related to the new definitions.

In Section 3.1.3, we show how to implement one iteration of our methods by computing an (inexact) affine-invariant tensor step. For the methods of degree $p \geq 1$, we establish global convergence in the functional residual of the order $\mathcal{O}(1/k^p)$, where k is the iteration counter. For $p = 1$, this recovers well-known result about global convergence of the classical Frank-Wolfe algorithm [52, 116]. For $p = 2$, we obtain the new algorithm called Contracting-Point Newton Method. Our analysis also works in the case, when the corresponding subproblem is solved inexactly.

In Section 3.1.4, we discuss our results and highlight some open questions for the future research.

3.1.1 Conceptual Contracting-Point Methods

We are interested in solving the composite convex minimization problem, with an additional assumption that the feasible set is *bounded*. Hence,

$$F^* \overset{\text{def}}{=} \min_{x \in \operatorname{dom} \psi} \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\},$$

where $\psi : \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ is a *simple* proper closed convex function with *bounded domain*, and function $f(x)$ is convex and p (≥ 1) times continuously differentiable at every point $x \in \operatorname{dom} \psi \subseteq \operatorname{dom} f$.

In this section, we propose a conceptual optimization scheme for solving this problem. At each step of our method, we choose a contracting coefficient $\gamma_k \in (0,1]$ restricting the nontrivial part of our objective $f(\cdot)$ onto a *contracted domain*. At the same time, the domain for the composite part remains unchanged.

Namely, at point $x_k \in \operatorname{dom} \psi$, define

$$S_k(y) \quad \overset{\text{def}}{=} \quad \gamma_k \psi\big(x_k + \tfrac{1}{\gamma_k}(y - x_k)\big), \quad y = x_k + \gamma_k(v - x_k), \quad v \in \operatorname{dom} \psi.$$

Note that $S_k(y) = \gamma_k \psi(v)$. Consider the following *exact* iteration:

$$\boxed{\begin{array}{rcl} v_{k+1}^* & \in & \operatorname*{Argmin}_v \Big\{ f(y) + S_k(y) : \\ & & \qquad\qquad y = (1-\gamma_k)x_k + \gamma_k v, \ v \in \operatorname{dom} \psi \Big\}, \\ x_{k+1}^* & = & (1-\gamma_k)x_k + \gamma_k v_{k+1}^*. \end{array}} \tag{3.1.1}$$

Of course, when $\gamma_k = 1$, exact step from (3.1.1) solves the initial problem. However, we are going to look at the *inexact* minimizer. In this case, the choice of $\{\gamma_k\}_{k \geq 0}$ should take into account the efficiency of solving the auxiliary subproblem.

Denote by $F_k(\cdot)$ the objective in the auxiliary problem (3.1.1), that is

$$F_k(y) \quad \overset{\text{def}}{=} \quad f(y) + S_k(y), \quad y \ = \ (1-\gamma_k)x_k + \gamma_k v, \quad v \in \operatorname{dom} \psi.$$

We are going to use the point $\bar{x}_{k+1} = (1-\gamma_k)x_k + \gamma_k \bar{v}_{k+1}$ with $\bar{v}_{k+1} \in \operatorname{dom} \psi$ having a *small residual* in the function value:

$$F_k(\bar{x}_{k+1}) - F_k(x_{k+1}^*) \quad \leq \quad \delta_{k+1}, \tag{3.1.2}$$

with some fixed $\delta_{k+1} \geq 0$.

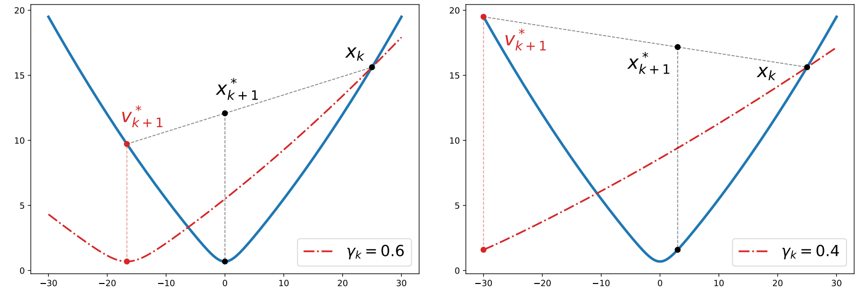


Figure 3.2: Iteration of the conceptual Contracting-Point method for different values of contracting coefficient γ_k, minimizing $f(\cdot)$ over segment $\mathrm{dom}\,\psi \equiv [-30, 30]$.

Lemma 3.1.1. *For all $k \geq 0$ and $v \in \mathrm{dom}\,\psi$, we have*

$$F(\bar{x}_{k+1}) \quad \leq \quad (1-\gamma_k)F(x_k) + \gamma_k F(v) + \delta_{k+1}. \tag{3.1.3}$$

Proof. Indeed, for any $v \in \mathrm{dom}\,\psi$, we have

$$\begin{array}{rcl} F_k(\bar{x}_{k+1}) & \overset{(3.1.2)}{\leq} & F_k(x^*_{k+1}) + \delta_{k+1} \\ & \overset{(3.1.1)}{\leq} & f((1-\gamma_k)x_k + \gamma_k v) + S_k((1-\gamma_k)x_k + \gamma_k v) + \delta_{k+1} \\ & \leq & (1-\gamma_k)f(x_k) + \gamma_k f(v) + \gamma_k \psi(v) + \delta_{k+1}. \end{array}$$

Therefore,

$$\begin{array}{rcl} F(\bar{x}_{k+1}) & = & F_k(\bar{x}_{k+1}) + \psi(\bar{x}_{k+1}) - \gamma_k \psi(\bar{v}_{k+1}) \\ & \leq & (1-\gamma_k)f(x_k) + \gamma_k F(v) + \delta_{k+1} + \psi(\bar{x}_{k+1}) - \gamma_k \psi(\bar{v}_{k+1}) \\ & \leq & (1-\gamma_k)F(x_k) + \gamma_k F(v) + \delta_{k+1}. \end{array}$$

□

Let us write down our method in algorithmic form.

```
Conceptual Contracting-Point Method, I

Initialization. Choose $x_0 \in \mathrm{dom}\, \psi$.

Iteration $k \geq 0$.

    1: Choose $\gamma_k \in (0, 1]$.

    2: For some $\delta_{k+1} \geq 0$, find $\bar{x}_{k+1}$ satisfying (3.1.2).

    3: If $F(\bar{x}_{k+1}) \leq F(x_k)$, then set $x_{k+1} = \bar{x}_{k+1}$.
       Else choose $x_{k+1} = x_k$.
```
(3.1.4)

In Step 3 of this method, we add a simple test for ensuring monotonicity in the function value. This step is optional.

It is more convenient to describe the rate of convergence of this scheme with respect to another sequence of parameters. Let us introduce an arbitrary sequence of positive numbers $\{a_k\}_{k \geq 1}$ and denote $A_k \overset{\text{def}}{=} \sum_{i=1}^{k} a_i$. Then, we can define the contracting coefficients as follows

$$\gamma_k \quad \overset{\text{def}}{=} \quad \tfrac{a_{k+1}}{A_{k+1}}. \tag{3.1.5}$$

Theorem 3.1.2. *For all points of sequence $\{x_k\}_{k \geq 0}$, generated by algorithm (3.1.4), we have the following relation:*

$$A_k F(x_k) \quad \leq \quad A_k F^* + B_k, \quad \textit{with} \quad B_k \overset{\text{def}}{=} \textstyle\sum_{i=1}^{k} A_i \delta_i. \tag{3.1.6}$$

Proof. Indeed, for $k = 0$, we have $A_k = 0$, $B_k = 0$. Hence, (3.1.6) is valid.

Assume it is valid for some $k \geq 0$. Then

$$\begin{array}{rcl}
A_{k+1}F(x_{k+1}) & \overset{\text{Step 3}}{\leq} & A_{k+1}F(\bar{x}_{k+1}) \\[1em]
& \leq & A_{k+1}\Big((1-\gamma_k)F(x_k) + \gamma_k F^* + \delta_{k+1}\Big) \\[1em]
& \overset{(3.1.5)}{=} & A_k F(x_k) + a_{k+1}F^* + A_{k+1}\delta_{k+1} \\[1em]
& \overset{(3.1.6)}{\leq} & A_{k+1}F^* + B_{k+1}. \qquad \square
\end{array}$$

From bound (3.1.6), we can see, that

$$F(x_k) - F^* \quad \leq \quad \tfrac{1}{A_k}\textstyle\sum_{i=1}^k A_i \delta_i, \qquad k \geq 1. \tag{3.1.7}$$

Hence, the actual rate of convergence of method (3.1.4) depends on the growth of coefficients $\{A_k\}_{k\geq 1}$ *relatively* to the level of inaccuracies $\{\delta_k\}_{k \geq 1}$. Potentially, this rate can be arbitrarily high. Since we did not assume anything yet about our objective function, this means that we just retransmitted the complexity of solving the initial problem onto a lower level, the level of computing the point $\bar{x}_{k+1}$, satisfying the condition (3.1.2). We are going to discuss different possibilities for that in Sections 3.1.3 and 4.2.

Now, let us endow method (3.1.4) with a computable *accuracy certificate*. For this purpose, for a sequence of given test points $\{\bar{x}_k\}_{k\geq 1} \subset \mathrm{dom}\,\psi$, we introduce the following *Estimating Function* (see [117]):

$$\varphi_k(v) \quad \overset{\text{def}}{=} \quad \sum_{i=1}^k a_i\big[f(\bar{x}_i) + \langle \nabla f(\bar{x}_i), v - \bar{x}_i\rangle + \psi(v)\big].$$

By convexity of $f(\cdot)$, we have $A_k F(v) \geq \varphi_k(v)$ for all $v \in \mathrm{dom}\,\psi$. Hence, for all $k \geq 1$, we can get the following bound for the functional residual:

$$\begin{array}{rcl}
F(x_k) - F^* \;\leq\; \ell_k & \overset{\text{def}}{=} & F(x_k) - \frac{1}{A_k}\varphi_k^*, \\[1em]
\varphi_k^* & \overset{\text{def}}{=} & \min\limits_{v \in \mathrm{dom}\,\psi} \varphi_k(v).
\end{array} \tag{3.1.8}$$

The complexity of computing the value of ℓ_k usually does not exceed the complexity of computing the next iterate of our method since it requires just one call of the *linear minimization oracle*. Let us show that an appropriate rate of decrease of the estimates ℓ_k can be guaranteed by sufficiently accurate steps of the method (3.1.1).

For that, we need a stronger condition on point $\bar{x}_{k+1}$, that is, for arbitrary $v \in \operatorname{dom} \psi$

$$\begin{array}{rcl} \langle \nabla f(\bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle + \psi(v) & \geq & \psi(\bar{v}_{k+1}) - \frac{1}{\gamma_k} \delta_{k+1}, \\ \bar{x}_{k+1} & = & (1-\gamma_k) x_k + \gamma_k \bar{v}_{k+1}, \end{array} \tag{3.1.9}$$

with some $\delta_{k+1} \geq 0$. Note that, for $\delta_{k+1} = 0$, condition (3.1.9) ensures the exactness of the corresponding step of method (3.1.1).

Let us consider now the following algorithm.

```
Conceptual Contracting-Point Method, II

Initialization. Choose $x_0 \in \operatorname{dom} \psi$.

Iteration $k \geq 0$.

1: Choose $\gamma_k \in (0, 1]$.

2: For some $\delta_{k+1} \geq 0$, find $\bar{x}_{k+1}$ satisfying (3.1.9).

3: If $F(\bar{x}_{k+1}) \leq F(x_k)$, then set $x_{k+1} = \bar{x}_{k+1}$.
   Else choose $x_{k+1} = x_k$.
```

(3.1.10)

This scheme differs from the previous method (3.1.4) only in the characteristic condition (3.1.9) for the next test point.

Theorem 3.1.3. *For all points of the sequence $\{x_k\}_{k \geq 0}$, generated by algorithm* (3.1.10)*, we have*

$$\varphi_k^* \quad \geq \quad A_k F(x_k) - B_k, \qquad k \geq 0. \tag{3.1.11}$$

Proof. For $k = 0$, relation (3.1.11) is valid since both sides are zeros. Assume that (3.1.11) holds for some $k \geq 0$. Then, for any $v \in \mathrm{dom}\,\psi$, we have

$$\begin{array}{rcl}
\varphi_{k+1}(v) & \equiv & \varphi_k(v) + a_{k+1}\big[f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), v - \bar{x}_{k+1}\rangle + \psi(v)\big] \\[1em]
& \overset{(3.1.11)}{\geq} & A_k F(x_k) - B_k \\[1em]
& & \quad + \, a_{k+1}\big[f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), v - \bar{x}_{k+1}\rangle + \psi(v)\big] \\[1em]
& \overset{(*)}{\geq} & A_{k+1}\big[f(\bar{x}_{k+1}) + \langle \nabla f(\bar{x}_{k+1}), \tfrac{a_{k+1}v + A_k x_k}{A_{k+1}} - \bar{x}_{k+1}\rangle\big] \\[1em]
& & \quad + \, A_k\psi(x_k) + a_{k+1}\psi(v) - B_k \\[1em]
& = & A_{k+1} f(\bar{x}_{k+1}) + a_{k+1}\big[\langle \nabla f(\bar{x}_{k+1}), v - \bar{v}_{k+1}\rangle + \psi(v)\big] \\[1em]
& & \quad + \, A_k\psi(x_k) - B_k \\[1em]
& \overset{(3.1.9)}{\geq} & A_{k+1} f(\bar{x}_{k+1}) + a_{k+1}\psi(\bar{v}_{k+1}) + A_k\psi(x_k) - B_{k+1} \\[1em]
& \overset{(**)}{\geq} & A_{k+1} F(\bar{x}_{k+1}) - B_{k+1} \\[1em]
& \overset{\text{Step 3}}{\geq} & A_{k+1} F(x_{k+1}) - B_{k+1}.
\end{array}$$

Here, the inequalities $(*)$ and $(**)$ are justified by convexity of $f(\cdot)$ and $\psi(\cdot)$, correspondingly. Thus, (3.1.11) is proved for all $k \geq 0$. □

Combining now (3.1.8) with (3.1.11), we obtain

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \tfrac{1}{A_k}\textstyle\sum_{i=1}^{k} A_i \delta_i, \quad k \geq 1. \tag{3.1.12}$$

We see that the right hand side in (3.1.12) is the same, as that one in (3.1.7). However, this convergence is stronger, since it provides a bound for the accuracy certificate ℓ_k.

3.1.2 Affine-Invariant High-Order Smoothness Condition

We are going to describe the efficiency of solving the auxiliary problem in (3.1.1). For that we use *affine-invariant* characteristics of variation of function $f(\cdot)$ over the compact convex sets. For a convex set Q, positive integer $p \geq 1$, and $\nu \in [0,1]$, define

$$\begin{aligned} \Delta_Q^{(p,\nu)}(f) \quad &\overset{\text{def}}{=} \quad \sup_{\substack{x,v\in Q,\\ t\in(0,1]}} \tfrac{1}{t^{p+\nu}} \Big| f(x+t(v-x)) - f(x) \\ &\qquad\qquad - \sum_{i=1}^{p} \tfrac{t^i}{i!} D^i f(x)[v-x]^i \Big|. \end{aligned} \tag{3.1.13}$$

Note that for $p=1$ and $\nu=1$ this characteristic was considered in [73] for the analysis of the classical Frank-Wolfe algorithm.

It is clear that for any $0 \leq \nu_1 < \nu_2 \leq 1$, we have

$$\Delta_Q^{(p,\nu_1)}(f) \quad \leq \quad \Delta_Q^{(p,\nu_2)}(f).$$

In many situations, it is more convenient to use an upper bound for $\Delta_Q^{(p,1)}(f)$, which is a full variation of the $(p+1)$th derivative over the given set Q:

$$\mathcal{V}_Q^{(p+1)}(f) \quad \overset{\text{def}}{=} \quad \sup_{x,y,v\in Q} \Big| D^{p+1} f(y)[v-x]^{p+1} \Big|. \tag{3.1.14}$$

Indeed, by Taylor formula, we have

$$\begin{aligned} &\tfrac{1}{t^{p+1}} \Big[f(x+t(v-x)) - f(x) - \sum_{i=1}^{p} \tfrac{t^i}{i!} D^i f(x)[v-x]^i \Big] \\ = \quad &\tfrac{1}{p!} \int_0^1 (1-\tau)^p D^{p+1} f(x+\tau t(v-x))[v-x]^{p+1} d\tau. \end{aligned}$$

Hence,

$$\Delta_Q^{(p,1)}(f) \quad \leq \quad \tfrac{1}{(p+1)!} \mathcal{V}_Q^{(p+1)}(f). \tag{3.1.15}$$

Sometimes, in order to exploit a *primal-dual* structure of the problem, we need to work with the dual objects (gradients), as in method (3.1.10). In this case, we need to characterize the variation of the gradient $\nabla f(\cdot)$ over the set Q:

$$\begin{aligned}\Gamma_Q^{(p,\nu)}(f) \quad &\overset{\text{def}}{=} \quad \sup_{\substack{x,y,v \in Q, \\ t \in (0,1]}} \tfrac{1}{t^{p+\nu-1}} \Big| \langle \nabla f(x+t(v-x)) - \nabla f(x) \\ & \qquad - \sum_{i=2}^{p} \tfrac{t^{i-1}}{(i-1)!} D^i f(x)[v-x]^{i-1}, v-y\rangle \Big|.\end{aligned} \tag{3.1.16}$$

Since

$$\begin{aligned} & \tfrac{1}{t}\Big[f(x+t(v-x)) - f(x) - \sum_{i=1}^{p} \tfrac{t^i}{i!} D^i f(x)[v-x]^i \Big] \\ = \quad & \tfrac{1}{t}\Big[\int_0^1 \langle \nabla f(x+\tau t(v-x)), t(v-x)\rangle d\tau - \sum_{i=1}^{p} \tfrac{t^i}{i!} D^i f(x)[v-x]^i \Big] \\ = \quad & \int_0^1 \langle \nabla f(x+\tau t(v-x)) - \sum_{i=1}^{p} \tfrac{(\tau t)^{i-1}}{(i-1)!} D^i f(x)[v-x]^{i-1}, v-x\rangle d\tau, \end{aligned}$$

we conclude that

$$\Delta_Q^{(p,\nu)}(f) \quad \leq \quad \tfrac{1}{p+\nu} \Gamma_Q^{(p,\nu)}(f). \tag{3.1.17}$$

At the same time, by Taylor formula, we get

$$\begin{aligned} & \tfrac{1}{t^p}\Big[\nabla f(x+t(v-x)) - \nabla f(x) - \sum_{i=2}^{p} \tfrac{t^{i-1}}{(i-1)!} D^i f(x)[v-x]^{i-1} \Big] \\ & = \tfrac{1}{(p-1)!} \int_0^1 (1-\tau)^{p-1} D^{p+1} f(x+\tau t(v-x))[v-x]^p d\tau. \end{aligned} \tag{3.1.18}$$

Therefore, again we have an upper bound in terms of the variation of the $(p+1)$th derivative, that is

$$\begin{aligned} \Gamma_Q^{(p,1)}(f) \quad & \overset{(3.1.18)}{\leq} \quad \tfrac{1}{p!} \sup_{x,y,z,v \in Q} \Big| \langle D^{p+1} f(z)[v-x]^p, v-y\rangle \Big| \\ & \leq \quad \tfrac{2(p+1)^p}{(p!)^2} \mathcal{V}_Q^{(p+1)}(f). \end{aligned} \tag{3.1.19}$$

See Proposition A.1 in Appendix for the proof of the last inequality. Hence, the value of $\mathcal{V}_Q^{(p+1)}(f)$ is the biggest one. However, in many cases it is more convenient.

Example 3.1.4. Let $\| \cdot \|$ be an arbitrary norm defined on the primal vector space $\mathbb{E}$, and let $Q \subset \mathbb{E}$ be a compact convex set with diameter

$$\mathscr{D} \;=\; \mathscr{D}_{\|\cdot\|}(Q) \;\overset{\text{def}}{=}\; \max_{x,y \in Q} \|x-y\| \;<\; +\infty.$$

Let W be an open set containing it: $Q \subset W \subseteq \mathbb{E}$. Assume that function f is $(p+1)$-times continuously differentiable on W, and its p-th derivative is Lipschitz continuous on W (w.r.t. $\| \cdot \|$) with constant L_p:

$$\begin{aligned} &\|D^p f(x) - D^p f(y)\| \\ &\overset{\text{def}}{=} \max_{h_1,\dots,h_p \in \mathbb{E}} \Big\{ |D^p f(x)[h_1,\dots,h_p] - D^p f(y)[h_1,\dots,h_p]| \,:\, \forall i(\|h_i\| \leq 1) \Big\} \\ &\leq L_p \|y-x\|, \qquad \forall x,y \in W. \end{aligned}$$

Then, we have

$$\mathcal{V}_Q^{(p+1)}(f) \;\leq\; L_p \mathscr{D}^{p+1}.$$

Consequently,

$$\Delta_Q^{(p,1)} \;\overset{(3.1.15)}{\leq}\; \tfrac{1}{(p+1)!} \mathcal{V}_Q^{(p+1)}(f) \;\leq\; \tfrac{1}{(p+1)!} L_p \mathscr{D}^{p+1}$$

and

$$\Gamma_Q^{(p,1)}(f) \;\overset{(3.1.19)}{\leq}\; \tfrac{2(p+1)^p}{(p!)^2} \mathcal{V}_Q^{(p+1)}(f) \;\leq\; \tfrac{2(p+1)^p}{(p!)^2} L_p \mathscr{D}^{p+1}. \qquad \square$$

Example 3.1.5. Assume in the previous example that the p-th derivative of f is Hölder continuous of degree $\nu \in [0,1]$ on Q with some constant $\mathcal{H}_{p,\nu}$:

$$\|D^p f(x) - D^p f(y)\| \;\leq\; \mathcal{H}_{p,\nu} \|x-y\|^{\nu}, \qquad \forall x,y \in Q.$$

Then, we have

$$\Gamma_Q^{(p,\nu)}(f) \;\leq\; \tfrac{1}{(p-1)!} \mathcal{H}_{p,\nu} \mathscr{D}^{p+\nu}$$

and hence

$$\Delta_Q^{(p,\nu)} \;\overset{(3.1.17)}{\leq}\; \tfrac{1}{p+\nu} \Gamma_Q^{(p,\nu)}(f) \;\leq\; \tfrac{1}{(p-1)! \cdot (p+\nu)} \mathcal{H}_{p,\nu} \mathscr{D}^{p+\nu}. \qquad \square$$

In some situations we can obtain much better estimates.

Example 3.1.6. Let $A \succeq 0$, and $f(x) = \frac{1}{2}\langle Ax, x\rangle$ with

$$x \in \mathbb{S}_n \quad \overset{\mathrm{def}}{=} \quad \{x \in \mathbb{R}^n_+ : \sum\limits_{i=1}^{n} x^{(i)} = 1\}.$$

For measuring distances in the standard simplex, we choose ℓ_1-norm:

$$\|h\| \quad = \quad \sum\limits_{i=1}^{n} |h^{(i)}|, \quad h \in \mathbb{R}^n.$$

In this case, $\mathscr{D} = \mathscr{D}_{\|\cdot\|}(\mathbb{S}_n) = 2$, and $L_1 = \max\limits_{1 \leq i \leq n} A^{(i,i)}$. On the other hand,

$$\begin{aligned} \mathcal{V}^{(2)}_{\mathbb{S}_n}(f) \quad &= \quad \max\limits_{1 \leq i,j \leq n} \langle A(e_i - e_j), e_i - e_j\rangle \\ &\leq \quad \max\limits_{1 \leq i,j \leq n} [2\langle Ae_i, e_i\rangle + 2\langle Ae_j, e_j\rangle] \;=\; 4L_1, \end{aligned}$$

where e_k denotes the kth coordinate vector in $\mathbb{R}^n$. Thus, $\mathcal{V}^{(2)}_{\mathbb{S}_n} \leq L_1 \mathscr{D}^2$.

However, for some matrices, the value $\mathcal{V}^{(2)}_{\mathbb{S}_n}(f)$ can be much smaller than $L_1\mathscr{D}^2$. Indeed, let $A = aa^T$ for some $a \in \mathbb{R}^n$. Then $L_1 = \max\limits_{1 \leq i \leq n} (a^{(i)})^2$, and

$$\mathcal{V}^{(2)}_{\mathbb{S}_n}(f) \quad = \quad \left[\max\limits_{1 \leq i \leq n} a^{(i)} - \min\limits_{1 \leq i \leq n} a^{(i)}\right]^2,$$

which can be much smaller than $4L_1$. □

Example 3.1.7. Let given vectors $a_1, \dots, a_m$ span the whole $\mathbb{R}^n$. Consider the objective

$$f(x) \quad = \quad \log\left(\sum\limits_{k=1}^{m} e^{\langle a_k, x\rangle}\right), \qquad x \in \mathbb{S}_n.$$

Then, it holds (see Example 1.3.5 for the first inequality):

$$\begin{aligned} \langle \nabla^2 f(x)h, h\rangle \quad &\leq \quad \max\limits_{1 \leq k,l \leq m} \langle a_k - a_l, h\rangle^2 \\ &\leq \quad \max\limits_{1 \leq k,l \leq m} \|a_k - a_l\|^2_\infty \|h\|^2_1, \qquad h \in \mathbb{R}^n. \end{aligned}$$

Therefore, in ℓ_1-norm we have $L_1 = \max\limits_{1 \leq k,l \leq m} \max\limits_{1 \leq i \leq n} \left[a_k^{(i)} - a_l^{(i)}\right]^2$. At the same time,

$$\begin{aligned}\mathcal{V}^{(2)}_{\mathbb{S}_n}(f) &= \sup_{x \in \mathbb{S}_n} \max_{1 \leq i,j \leq n} \langle \nabla^2 f(x)(e_i - e_j), e_i - e_j \rangle \\ &\leq \max_{1 \leq k,l \leq m} \max_{1 \leq i,j \leq n} \Big[\big(a_k^{(i)} - a_k^{(j)}\big) - \big(a_l^{(i)} - a_l^{(j)}\big) \Big]^2 .\end{aligned}$$

The last expression is the maximal difference between variations of the coordinates. It can be much smaller than $L_1 \mathscr{D}^2 = 4L_1$.

Moreover, we have (see Example 2.1.1):

$$|D^3 f(x)[h]^3| \quad \leq \quad \max_{1 \leq k,l \leq m} |\langle a_k - a_l, h \rangle|^3, \qquad h \in \mathbb{R}^n.$$

Hence, we obtain

$$\mathcal{V}^{(3)}_{\mathbb{S}_n}(f) \quad \leq \quad \max_{1 \leq k,l \leq m} \max_{1 \leq i,j \leq n} \Big| \big(a_k^{(i)} - a_k^{(j)}\big) - \big(a_l^{(i)} - a_l^{(j)}\big) \Big|^3 . \qquad \Box$$

3.1.3 Contracting-Point Tensor Methods

In this section, we show how to implement Contracting-Point Methods, by using affine-invariant tensor steps. At each iteration of (3.1.1), we approximate $f(\cdot)$ by its Taylor's polynomial of degree $p \geq 1$ around the current point x_k:

$$f(y) \quad \approx \quad \Omega_p(f, x_k; y) \quad \overset{\text{def}}{=} \quad f(x_k) + \sum_{i=1}^{p} \tfrac{1}{i!} D^i f(x_k)[y - x_k]^i.$$

Thus, we need to solve the following auxiliary problem:

$$\begin{gathered}\min_{v} \Big\{ M_k(y) \overset{\text{def}}{=} \Omega_p(f, x_k; y) + S_k(y) : \\ y = (1 - \gamma_k) x_k + \gamma_k v, \; v \in \operatorname{dom} \psi \Big\}.\end{gathered} \tag{3.1.20}$$

Note that this global minimum M_k^* is well defined since $\operatorname{dom} \psi$ is bounded. Let us take

$$\bar{x}_{k+1} \quad = \quad (1 - \gamma_k) x_k + \gamma_k \bar{v}_{k+1},$$

where $\bar{v}_{k+1}$ is an inexact solution to (3.1.20) in the following sense:

$$M_k(\bar{x}_{k+1}) - M_k^* \quad \leq \quad \xi_{k+1}. \tag{3.1.21}$$

Then, this point serves as a good candidate for the inexact step of our method.

Theorem 3.1.8. *Let $\xi_{k+1} \leq c\gamma_k^{p+\nu}$, for some arbitrary constants $c \geq 0$ and $\nu \in [0,1]$. Then*

$$F_k(\bar{x}_{k+1}) - F_k^* \quad \leq \quad \delta_{k+1},$$

for $\delta_{k+1} = (c + 2\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f))\gamma_k^{p+\nu}$.

Proof. Indeed, for $y = x_k + \gamma_k(v - x_k)$ with arbitrary $v \in \mathrm{dom}\,\psi$, we have

$$\begin{aligned}
F_k(y) \quad &= \quad f(y) + S_k(y) \\
&\overset{(3.1.13)}{\geq} \quad \Omega_p(f, x_k; y) + S_k(y) - \Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f)\gamma_k^{p+\nu} \\
&\overset{(3.1.21)}{\geq} \quad \Omega_p(f, x_k; \bar{x}_{k+1}) + S_k(\bar{x}_{k+1}) - (c + \Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f))\gamma_k^{p+\nu} \\
&\overset{(3.1.13)}{\geq} \quad f(\bar{x}_{k+1}) + S_k(\bar{x}_{k+1}) - (c + 2\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f))\gamma_k^{p+\nu} \\
&= \quad F_k(\bar{x}_{k+1}) - \delta_{k+1}.
\end{aligned}$$

□

Thus, we come to the following minimization scheme.

Contracting-Point Tensor Method, I

(3.1.22)

```
Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$, $c \geq 0$.

Iteration $k \geq 0$.

1: Choose $\gamma_k \in (0, 1]$.

2: For some $\xi_{k+1} \leq c\gamma_k^{p+1}$, find $\bar{x}_{k+1}$ satisfying (3.1.21).

3: If $F(\bar{x}_{k+1}) \leq F(x_k)$, then set $x_{k+1} = \bar{x}_{k+1}$.
   Else choose $x_{k+1} = x_k$.
```

Note that we do not fix any particular ν in our scheme, because we want to have a *universal* method that does not depend on the degree of Hölder continuity.

For $p = 1$ and $\psi(\cdot)$ being an indicator function of a compact convex set, this is the well-known Frank-Wolfe algorithm [52]. Inexact version of the Frank-Wolfe algorithm was analysed in [73].

Straightforward consequence of our observations is the following

Theorem 3.1.9. *Let $\gamma_k = \frac{p+1}{k+p+1}$. Then, for all iterations $\{x_k\}_{k\geq 1}$ generated by method (3.1.22), we have*

$$F(x_k) - F^* \quad \leq \quad \tfrac{6(p+1)^{p+\nu}}{2-\nu} \cdot (c + 2\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}) \cdot k^{-(p+\nu-1)}, \qquad \forall \nu \in [0,1].$$

Proof. Let us choose $A_k = k \cdot (k+1) \cdot \ldots \cdot (k+p)$. Then, $a_{k+1} = A_{k+1} - A_k = \frac{(p+1)A_{k+1}}{k+p+1}$, and

$$\gamma_k \quad = \quad \tfrac{a_{k+1}}{A_{k+1}} \;=\; \tfrac{p+1}{k+p+1}.$$

Combining (3.1.7) with Theorem 3.1.8, we have

$$F(x_k) - F^* \quad \leq \quad \tfrac{(c+2\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f))}{A_k} \sum\limits_{i=1}^{k} \tfrac{a_i^{p+\nu}}{A_i^{p+\nu-1}}, \qquad k \geq 1.$$

Since

$$\begin{aligned}
\tfrac{1}{A_k} \sum\limits_{i=1}^{k} \tfrac{a_i^{p+\nu}}{A_i^{p+\nu-1}} \quad &= \quad \tfrac{1}{A_k} \sum\limits_{i=1}^{k} \tfrac{(p+1)^{p+\nu} A_i}{(i+p)^{p+\nu}} \quad \leq \quad \tfrac{(p+1)^{p+\nu}}{A_k} \sum\limits_{i=1}^{k} (i+p)^{1-\nu} \\
&\leq \quad \tfrac{(p+1)^{p+\nu}}{A_k} \int\limits_0^{k+1} (\tau + p)^{1-\nu} d\tau \quad = \quad \tfrac{(p+1)^{p+\nu}(k+p+1)^{2-\nu}}{(2-\nu)A_k} \\
&\leq \quad \tfrac{6(p+1)^{p+\nu}}{2-\nu} \cdot \tfrac{(k+p+1)^{2-\nu}}{k^{p-1}\cdot(k+p+1)^2} \quad \leq \quad \tfrac{6(p+1)^{p+\nu}}{2-\nu} \cdot \tfrac{1}{k^{p+\nu-1}},
\end{aligned}$$

we get the required inequality. □

Hence, in order to find an ε-solution to the problem: $F(x_K) - F^* \leq \varepsilon$, it is enough to do

$$K \quad = \quad \inf\limits_{\nu \in [0,1]} \Big[\tfrac{6(p+1)^{p+\nu}}{2-\nu} \cdot \tfrac{(c+2\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f))}{\varepsilon} \Big]^{\frac{1}{p+\nu-1}}$$

iterations of the method. And an appropriate choice of parameter c is the value $\Delta^{(p,\nu)}_{\mathrm{dom}\,\psi}(f)$ for some $\nu \in [0,1]$. In practice, it seems reasonable to use a small constant for the value of c.

It is important that the required level of accuracy ξ_{k+1} for solving the subproblem is not static: it is changing with iterations. Indeed, from the practical perspective, there is no need to use high accuracy during the first iterations, but it is natural to improve our precision while approaching the optimum. Inexact proximal-type tensor methods with dynamic inner accuracies are studied in Section 4.1 of Chapter 4.

Let us note that the objective $M_k(y)$ from (3.1.20) is generally nonconvex for $p \geq 3$, and it may be nontrivial to look for its global minimum. Because of that, we propose an alternative condition for the next point. It requires just to find an inexact *stationary point* of $\Omega_p(f, x_k; y)$. That is a point $\bar{x}_{k+1}$, satisfying, for all $v \in \operatorname{dom} \psi$

$$\begin{array}{rcl} \langle \nabla \Omega_p(f, x_k; \bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle + \psi(v) & \geq & \psi(\bar{v}_{k+1}) - \frac{1}{\gamma_k}\xi_{k+1}, \\ \bar{x}_{k+1} & = & (1-\gamma_k)x_k + \gamma_k \bar{v}_{k+1}, \end{array} \tag{3.1.23}$$

for some tolerance value $\xi_{k+1} \geq 0$.

Theorem 3.1.10. *Let point $\bar{x}_{k+1}$ satisfy condition (3.1.23) with*

$$\xi_{k+1} \quad \leq \quad c\gamma_k^{p+\nu},$$

for some constants $c \geq 0$ and $\nu \in [0,1]$. Then it satisfies inexact condition (3.1.9) of the Conceptual Contracting-Point Method with

$$\delta_{k+1} \quad = \quad (c + \Gamma^{(p,\nu)}_{\operatorname{dom}\psi}(f))\gamma_k^{p+\nu}.$$

Proof. Indeed, for any $v \in \operatorname{dom} \psi$, we have

$$\begin{array}{rl} & \langle \nabla f(\bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle + \psi(v) \\ = & \langle \nabla \Omega_p(f, x_k; \bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle + \psi(v) \\ & + \quad \langle \nabla f(\bar{x}_{k+1}) - \Omega_p(f, x_k; \bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle \\ \overset{(3.1.23)}{\geq} & \psi(\bar{v}_{k+1}) - c\gamma_k^{p+\nu-1} + \langle \nabla f(\bar{x}_{k+1}) - \Omega_p(f, x_k; \bar{x}_{k+1}), v - \bar{v}_{k+1} \rangle \\ \overset{(3.1.16)}{\geq} & \psi(\bar{v}_{k+1}) - (c + \Gamma^{(p)}_{\operatorname{dom}\psi}(f))\gamma_k^{p+\nu-1} \quad = \quad \psi(\bar{v}_{k+1}) - \frac{1}{\gamma_k}\delta_{k+1}. \quad \square \end{array}$$

Now, changing inexactness condition (3.1.21) in method (3.1.22) by condition (3.1.23), we come to the following algorithm.

Contracting-Point Tensor Method, II

```
Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$, $c \geq 0$.

Iteration $k \geq 0$.

1: Choose $\gamma_k \in (0, 1]$.

2: For some $\xi_{k+1} \leq c\gamma_k^{p+1}$, find $\bar{x}_{k+1}$ satisfying (3.1.23).

3: If $F(\bar{x}_{k+1}) \leq F(x_k)$, then set $x_{k+1} = \bar{x}_{k+1}$.

   Else choose $x_{k+1} = x_k$.
```

(3.1.24)

Its convergence analysis is straightforward.

Theorem 3.1.11. *Let* $A_k \overset{\mathrm{def}}{=} k \cdot (k+1) \cdot \ldots \cdot (k+p)$, *and consequently* $\gamma_k = \frac{p+1}{k+p+1}$. *Then, for all iterations* $\{x_k\}_{k \geq 1}$ *of method* (3.1.24), *we have*

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \tfrac{6(p+1)^{p+\nu}}{2-\nu} \cdot (c + \Gamma^{(p)}_{\mathrm{dom}\,\psi}(f)) \cdot k^{-(p+\nu-1)},$$

for any $\nu \in [0, 1]$.

Proof. Combining inequality (3.1.12) with the statement of Theorem 3.1.10, we have

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \tfrac{c + \Gamma^{(p)}_{\mathrm{dom}\,\psi}(f)}{A_k} \sum_{i=1}^{k} \tfrac{a_i^{p+\nu}}{A_i^{p+\nu-1}}, \qquad k \geq 1.$$

It remains to use the same reasoning, as in the proof of Theorem 3.1.9. □

To finish this section, let us discuss the affine invariance of our new methods. In the exact form, iterations of the Contracting-Point Tensor

Method can be rewritten as follows, for $k \geq 0$:

$$x_{k+1} \quad \in \quad \operatorname*{Argmin}_x \Big\{ \Omega_p(f, x_k; x) + \gamma_k \psi(x_k + \tfrac{1}{\gamma_k}(x - x_k)) \Big\}, \tag{3.1.25}$$

when applied to the composite objective $F(x) = f(x) + \psi(x)$.

Let $A : \mathbb{E} \to \mathbb{E}$ be a nondegenerate linear operator. Let $b \in \mathbb{E}$ be a fixed vector. Consider the new functions

$$\tilde{f}(y) \quad := \quad f(Ay + b), \qquad \tilde{\psi}(y) \quad := \quad \psi(Ay + b), \qquad y \in \mathbb{E}.$$

Note that, for any $h \in \mathbb{E}$,

$$D^p \tilde{f}(y)[h]^p \quad = \quad D^p f(Ay + b)[Ah]^p. \tag{3.1.26}$$

We can prove the following simple statement.

Proposition 3.1.12. *Let sequence $\{x_k\}_{k \geq 0}$ be generated by method (3.1.25). Consider the corresponding $\{y_k\}_{k \geq 0}$ defined by*

$$y_k \quad := \quad A^{-1}(x_k - b), \qquad k \geq 0.$$

Then the latter sequence satisfies the iterations of the Contracting-Point Tensor Method applied to the new objective $\tilde{F}(y) := \tilde{f}(y) + \tilde{\psi}(y)$. Thus

$$y_{k+1} \quad \in \quad \operatorname*{Argmin}_y \Big\{ \Omega_p(\tilde{f}, y_k; y) + \gamma_k \tilde{\psi}(y_k + \tfrac{1}{\gamma_k}(y - y_k)) \Big\}. \tag{3.1.27}$$

Proof. Let us fix arbitrary $y \in \mathbb{E}$. Set $x = Ay + b$. Then,

$$\begin{aligned} \Omega_p(\tilde{f}, y_k; y) \quad &= \quad \tilde{f}(y_k) + \sum_{i=1}^{p} \tfrac{1}{i!} D^i \tilde{f}(y_k)[y - y_k]^i \\ &\overset{(3.1.26)}{=} \quad f(Ay_k + b) + \sum_{i=1}^{p} \tfrac{1}{i!} D^i f(Ay_k + b)[A(y - y_k)]^i \\ &= \quad f(x_k) + \sum_{i=1}^{p} \tfrac{1}{i!} D^i f(x_k)[x - x_k]^i \\ &= \quad \Omega_p(f, x_k; x). \end{aligned}$$

For the composite term, we have

$$\begin{array}{rcl}\tilde{\psi}(y_k + \frac{1}{\gamma_k}(y - y_k)) & = & \psi(A(y_k + \frac{1}{\gamma_k}(y - y_k)) + b) \\ & = & \psi(x_k + \frac{1}{\gamma_k}(x - x_k)).\end{array}$$

Hence,

$$\begin{array}{rl} & \Omega_p(\tilde{f}, y_k; y_{k+1}) + \gamma_k \tilde{\psi}(y_k + \frac{1}{\gamma_k}(y_{k+1} - y_k)) \\ = & \Omega_p(f, x_k; x_{k+1}) + \gamma_k \psi(x_k + \frac{1}{\gamma_k}(x_{k+1} - x_k)) \\ \overset{(3.1.25)}{\leq} & \Omega_p(f, x_k; x) + \gamma_k \psi(x_k + \frac{1}{\gamma_k}(x - x_k)) \\ = & \Omega_p(\tilde{f}, y_k; y) + \gamma_k \tilde{\psi}(y_k + \frac{1}{\gamma_k}(y - y_k)),\end{array}$$

which justifies (3.1.27). □

We see that the sequence generated by the method is invariant with respect to affine transformations of variables.

3.1.4 Discussion

We have presented a new general framework of Contracting-Point methods, which can be used for developing affine-invariant optimization algorithms of different order. For the methods of order $p \geq 1$, we prove the following global convergence rate:

$$F(x_k) - F^* \quad \leq \quad \mathcal{O}\big(1/k^p\big), \quad k \geq 1.$$

This is the same rate, as that of the basic high-order Proximal-Point scheme [120]. However, the methods from this section are free from using the norms or any other characteristic parameters of the problem. This nice property makes Contracting-Point methods favourable for solving optimization problems over the sets with a non-Euclidean geometry (e.g. over the simplex or over a general convex polytope).

At the same time, it is known that in Euclidean case, the prox-type methods can be accelerated, achieving $\mathcal{O}(1/k^{p+1})$ global rate of convergence [6, 118, 120]. Using additional one-dimensional search at each iteration, this rate can be improved up to $\mathcal{O}(1/k^{\frac{3p+1}{2}})$ (see [54, 120]). The latter rate is shown to be optimal [4, 117]. To the best of our knowledge, the lower bounds for high-order methods in general non-Euclidean case remain unknown. However, the worst-case oracle complexity of the classical Frank-Wolfe algorithm (the case $p = 1$ in our framework) is proven to be near-optimal for smooth minimization over $\| \cdot \|_{\infty}$-balls [67].

Another open question is a possibility of efficient implementation of our methods for the case $p \geq 3$. In absence of an explicit regularizer (contrary to the prox-type methods), the subproblem in (3.1.20) can be nonconvex. Hence, it seems hard to find its global minimizer. We hope that for some problem classes, it is still feasible to satisfy the inexact stationarity condition (3.1.23) by a reasonable amount of computations. We keep this question for further investigation.

3.2 Global Lower Second-Order Models

We have seen in Section 3.1.3 that one iteration of the Contracting-Point method can be implemented by using the p-th order Taylor's approximation of the smooth part of the objective. For $p = 2$, this results in a new second-order optimization scheme, called Contracting Newton Method.

This algorithm possesses the global convergence guarantee; it needs $\mathcal{O}\bigl(\varepsilon^{-1/2}\bigr)$ second-order oracle calls to solve the composite problem with *bounded domain* up to ε-accuracy in the functional residual (Theorem 3.1.9). This is the same rate as that of the Cubic Newton in a general convex case.

There are several differences between these two algorithms though. The Cubic Newton Method uses a global *upper* approximation model, which is the second-order Taylor's polynomial augmented by a cubic term. The regularizer is the third power of the Euclidean norm, and thus the method is no longer affine-invariant. However, the Cubic Newton has the global linear rate in the uniformly convex case (Section 2.1), and the local superlinear convergence (Section 2.2).

At the same time, the Contracting Newton Method is *affine-invariant*. It does not depend on a particular norm and the corresponding Lipschitz constants. In this part of the thesis, we deeply investigate the properties of the contracting second-order schemes.

First, we develop a new global second-order *lower* model of the objective function, introduced in Section 3.2.1.

Then, we provide the Contracting Newton Method with a new interpretation, incorporating this model into optimization schemes (Section 3.2.2).

For convex functions with a Hölder continuous Hessian of degree $\nu \in [0,1]$ (w.r.t. an arbitrary norm) we re-establish a global convergence rate for our algorithm of the order $\mathcal{O}(1/k^{1+\nu})$. When the composite component is strongly convex, we show $\mathcal{O}(1/k^{2+2\nu})$ global rate for a universal scheme, and global linear rate if the parameters of the problem class are known. We also provide different trust-region interpretations for our method.

In Section 3.2.3, we present aggregated second-order models, accumulating information into *quadratic Estimating Function*. Based on this, we develop an alternative optimization process, called Aggregating Newton Method. For this algorithm, we establish the global convergence of the same order $\mathcal{O}(1/k^{1+\nu})$ as for the Contracting Newton Method in the general convex case.

Section 3.2.4 contains numerical experiments. In Section 3.2.5, we discuss our results.

Recall that our goal is to solve the composite convex minimization problem:

$$\min_{x \in \mathrm{dom}\, \psi} \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\}.$$

Let us fix an arbitrary (possibly non-Euclidean) norm $\| \cdot \|$ on $\mathbb{E}$. We denote by $\mathscr{D}$ the corresponding diameter of $\mathrm{dom}\, \psi$:

$$\mathscr{D} \quad \overset{\mathrm{def}}{=} \quad \sup_{x,y \in \mathrm{dom}\, \psi} \|x - y\|. \tag{3.2.1}$$

In this section, our main assumption on the problem is that $\mathrm{dom}\, \psi$ is *bounded*:

$$\mathscr{D} \quad < \quad +\infty.$$

The most important example of ψ is $\{0, +\infty\}$-indicator of a simple compact convex set $Q = \mathrm{dom}\, \psi$. In particular, for a ball in $\| \cdot \|_p$-norm with $p \geq 1$ on $\mathbb{E} := \mathbb{R}^n$, this is

$$\psi(x) \quad = \quad \begin{cases} 0, & \|x\|_p := \Big(\sum_{i=1}^n |x^{(i)}|^p \Big)^{1/p} \leq \frac{\mathscr{D}}{2}, \\ +\infty, & \text{else}. \end{cases} \tag{3.2.2}$$

From the machine learning perspective, $\mathscr{D}$ is usually considered as a *regularization parameter* in this setting.

Having fixed the norm $\| \cdot \|$ for the primal space, the *dual* norm can be defined in the standard way,

$$\|s\|_* \overset{\text{def}}{=} \max_{h \in \mathbb{E}} \bigl\{ \langle s, h \rangle \ : \ \|h\| \leq 1 \bigr\}, \qquad s \in \mathbb{E}^*.$$

The dual norm is necessary for measuring the size of the gradients. For a linear operator $A : \mathbb{E} \to \mathbb{E}^*$, we use the corresponding induced operator norm, defined as

$$\|A\| \overset{\text{def}}{=} \max_{h \in \mathbb{E}} \bigl\{ \|Ah\|_* \ : \ \|h\| \leq 1 \bigr\}.$$

3.2.1 Second-Order Lower Model of Objective Function

To characterize the complexity of our problem, assume that the Hessian of f is Hölder continuous of degree $\nu \in [0,1]$ on $\operatorname{dom} \psi$, i.e., that

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq \mathcal{H}_\nu \|x - y\|^\nu, \qquad \forall x, y \in \operatorname{dom} \psi. \tag{3.2.3}$$

The actual parameters of this problem class may be unknown. However, we assume that for *some* $\nu \in [0,1]$ inequality (3.2.3) is satisfied with corresponding constant $0 \leq \mathcal{H}_\nu < +\infty$. The direct consequence of (3.2.3) is the following global bounds for Taylor's approximation, for all $x, y \in \operatorname{dom} \psi$

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\|_* \leq \tfrac{\mathcal{H}_\nu \|y-x\|^{1+\nu}}{1+\nu}, \tag{3.2.4}$$

$$\begin{array}{l} |f(y) - f(x) - \langle \nabla f(x), y - x \rangle - \frac{1}{2} \langle \nabla^2 f(x)(y-x), y-x \rangle| \\ \leq \quad \frac{\mathcal{H}_\nu \|y-x\|^{2+\nu}}{(1+\nu)(2+\nu)}. \end{array} \tag{3.2.5}$$

Recall, that in addition to (3.2.3), we assume that f is *convex*:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \qquad x, y \in \operatorname{dom} \psi. \tag{3.2.6}$$

Employing both smoothness and convexity, we are able to enhance this global lower bound, as follows.

Lemma 3.2.1. *For all* $x, y \in \operatorname{dom}\psi$ *and* $t \in [0,1]$, *it holds*

$$\begin{array}{rcl} f(y) & \geq & f(x) + \langle \nabla f(x), y - x\rangle + \frac{t}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle \\ & & - \frac{t^{1+\nu}\mathcal{H}_\nu \|y-x\|^{2+\nu}}{(1+\nu)(2+\nu)}. \end{array} \tag{3.2.7}$$

Proof. Let us prove the following bound, for all $x, y \in \operatorname{dom}\psi$ and $t \in [0,1]$

$$\begin{array}{l} \langle \nabla f(y) - \nabla f(x), y - x\rangle \\ \geq\ t\langle \nabla^2 f(x)(y-x), y-x\rangle - \frac{t^{1+\nu}\mathcal{H}_\nu \|y-x\|^{2+\nu}}{1+\nu}. \end{array} \tag{3.2.8}$$

For $t = 1$ it follows from (3.2.4). Therefore, we may assume that $t < 1$. Let us take $z_t \overset{\text{def}}{=} x + t(y-x)$. Then, by convexity of f, we have

$$\begin{array}{rcl} \langle \nabla f(y), y - x\rangle & = & \frac{1}{1-t}\langle \nabla f(y), y - z_t\rangle \\ & \geq & \frac{1}{1-t}\langle \nabla f(z_t), y - z_t\rangle \ = \ \langle \nabla f(z_t), y - x\rangle. \end{array}$$

Now, by Hölder continuity of the Hessian we get

$$\begin{array}{cl} \multicolumn{2}{l}{\langle \nabla f(z_t), y - x\rangle} \\ \overset{(3.2.4)}{\geq} & \langle \nabla f(x), y - x\rangle + \langle \nabla^2 f(x)(z_t - x), y - x\rangle - \frac{\mathcal{H}_\nu \|z_t - x\|^{1+\nu}\|y-x\|}{1+\nu} \\ = & \langle \nabla f(x), y - x\rangle + t\langle \nabla^2 f(x)(y-x), y-x\rangle - \frac{t^{1+\nu}\mathcal{H}_\nu \|y-x\|^{2+\nu}}{1+\nu}. \end{array}$$

Thus we prove (3.2.8). Then, the claim of the lemma can be obtained by simple integration:

$$\begin{array}{cl} \multicolumn{2}{l}{f(y) - f(x) - \langle \nabla f(x), y - x\rangle} \\ = & \int\limits_0^1 \langle \nabla f(z_\tau) - \nabla f(x), y - x\rangle d\tau \\ \overset{(3.2.8)}{\geq} & \int\limits_0^1 t\tau\langle \nabla^2 f(x)(y-x), y-x\rangle - \frac{(t\tau)^{1+\nu}\mathcal{H}_\nu \|y-x\|^{2+\nu}}{1+\nu} d\tau \\ = & \frac{t}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle - \frac{t^{1+\nu}\mathcal{H}_\nu \|y-x\|^{2+\nu}}{(1+\nu)(2+\nu)}. \qquad \square \end{array}$$

Note that the right-hand side of (3.2.7) is concave in $t \in [0,1]$, and for $t = 0$ we obtain the standard first-order lower bound. The maximization of (3.2.7) over t gives

$$f(y) \quad \geq \quad f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{\bar{\gamma}_{x,y}}{2} \langle \nabla^2 f(x)(y-x), y - x \rangle, \qquad (3.2.9)$$

with

$$\bar{\gamma}_{x,y} \quad \overset{\text{def}}{=} \quad \tfrac{\nu}{1+\nu} \min \Bigl\{ 1, \tfrac{(2+\nu)\langle \nabla^2 f(x)(y-x), y-x \rangle}{2\mathcal{H}_\nu \|y-x\|^{2+\nu}} \Bigr\}^{\frac{1}{\nu}}, \quad x \neq y, \quad \nu \in (0,1].$$

Thus, (3.2.9) is always *tighter* than (3.2.6), employing additional *global second-order information.* The relationship between them is shown in Figure 3.3. Hence, it seems natural to incorporate the second-order lower bounds into optimization schemes.

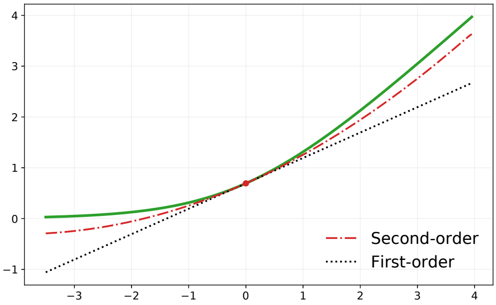


Figure 3.3: Global lower bounds for the logistic regression loss, $f(x) = \log(1 + \exp(x))$.

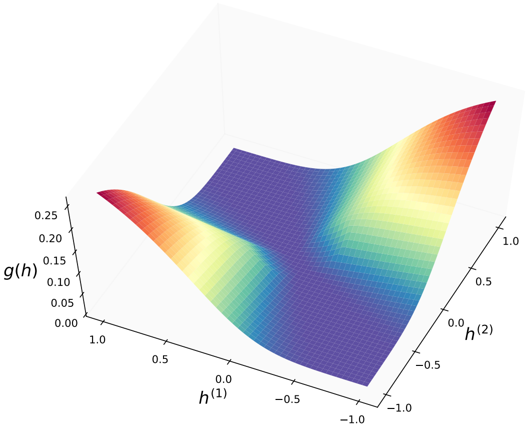


Figure 3.4: The graph of $g(h) = \langle \nabla^2 f(0)h, h\rangle \cdot \min\Bigl\{1, \frac{3\langle \nabla^2 f(0)h, h\rangle}{4\|h\|^3}\Bigr\}, h \in \mathbb{R}^2$, where $f(x) = \log\bigl(\exp(x^{(1)}) + \exp(x^{(2)})\bigr)$, whose Hessian is Lipschitz continuous with constant $L_2 = 2$ in the standard Euclidean norm (Example 1.3.5).

3.2.2 Contracting-Point Newton Methods

Let us introduce an optimization scheme which is based on global second-order lower bounds.

Note that the right hand side of (3.2.9) is nonconvex in y (see Figure 3.4). Hence, it can hardly be used directly in a computational algorithm. To tackle this issue, we use a sequence of contracting coefficients $\{\gamma_k\}_{k\geq 0}$. Each coefficient $\gamma_k \in (0,1]$ can be seen as an appropriate substitute of $\bar{\gamma}_{x,y}$ in (3.2.9). Then we minimize the corresponding global lower bound augmented by the composite component $\psi(\cdot)$. The next point is taken as a convex combination of the minimizer and the current point.

It appears that the iterations of such scheme coincide with the steps of the Contracting-Point Tensor Method (Section 3.1.3) for the particular instance $p = 2$.

Let us present the method in the algorithmic form. For simplicity, we consider the case when the method uses the *exact* solution to the subproblem.

```
Contracting-Point Newton Method, I

Initialization. Choose $x_0 \in \mathrm{dom}\, \psi$.

Iteration $k \geq 0$.

1: Pick up $\gamma_k \in (0, 1]$.                                    (3.2.10)

2: Compute
    $v_{k+1} \in \mathrm{Argmin}_y \Big\{ \langle \nabla f(x_k), y - x_k \rangle$
        $+ \frac{\gamma_k}{2} \langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \psi(y) \Big\}$.

3: Set $x_{k+1} = x_k + \gamma_k (v_{k+1} - x_k)$.
```

There is a clear connection of this method with Frank-Wolfe algorithm (the case $p = 1$ of the Contracting-Point Tensor Method). Indeed, instead of the standard first-order approximation (3.2.6), we use the lower global quadratic model. Thus, as compared with the gradient methods, every iteration of algorithm (3.2.10) is more expensive. However, this is a standard situation with the second-order schemes (see the below discussion on the iteration complexity). At the same time, our method is *affine-invariant*, since it does not depend on the norms.

It is clear that we obtain iterations of the classical Newton's method when $\gamma_k \equiv 1$. Its local quadratic convergence for composite optimization problems was established in Theorem 1.4.1. However, for the global convergence, we need to adjust the contracting coefficients accordingly. To state the global convergence result, let us define the following linear *Estimating Functions*:

$$\begin{aligned} \phi_k(x) &\overset{\mathrm{def}}{=} \textstyle\sum_{i=1}^{k} a_i \big[f(x_i) + \langle \nabla f(x_i), x - x_i \rangle + \psi(x) \big], \\ \phi_k^* &\overset{\mathrm{def}}{=} \min_x \phi_k(x), \end{aligned} \tag{3.2.11}$$

for the sequence of test points $\{x_k : x_k \in \operatorname{dom}\psi\}_{k\geq 1}$ and positive scaling coefficients $\{a_k\}_{k\geq 1}$. We relate them with contracting coefficients, as follows

$$\gamma_k \overset{\text{def}}{=} \tfrac{a_{k+1}}{A_{k+1}}, \qquad A_k \overset{\text{def}}{=} \textstyle\sum_{i=1}^k a_i. \tag{3.2.12}$$

We denote by $\mu \geq 0$ the constant of strong convexity of $\psi(\cdot)$. We allow $\mu = 0$ in the following auxiliary lemma, in order to cover both the general convex and the strongly convex cases. Thus, it holds

$$\psi(y) \quad \geq \quad \psi(x) + \langle \psi'(x), y - x\rangle + \tfrac{\mu}{2}\|y - x\|^2, \tag{3.2.13}$$

for all $x, y \in \operatorname{dom}\psi$ and for all $\psi'(x) \in \partial\psi(x)$.

Lemma 3.2.2. *For the sequences $\{x_k\}_{k\geq 1}$ and $\{v_k\}_{k\geq 1}$, produced by algorithm (3.2.10), we have*

$$A_k F(x_k) \quad \leq \quad \phi_k(x) \; + \; B_k(x), \qquad x \in \operatorname{dom}\psi, \tag{3.2.14}$$

with

$$\begin{aligned} B_k(x) \quad \overset{\text{def}}{=} \quad & \sum_{i=1}^k \Big[\tfrac{\mathcal{H}_\nu a_i^{2+\nu} \|x - v_i\| \cdot \|x_{i-1} - v_i\|^{1+\nu}}{(1+\nu) A_i^{1+\nu}} \\ & \qquad - \tfrac{\mu a_i \|x - v_i\|^2}{2} - \tfrac{\mu a_i A_{i-1} \|x_{i-1} - v_i\|^2}{2A_i} \Big]. \end{aligned} \tag{3.2.15}$$

Proof. Let us prove (3.2.14) by induction.

It obviously holds for $k = 0$, since $A_0 := 0$, $\phi_0(x) \equiv 0$, and $B_0(x) \equiv 0$ by definition.

Assume that it holds for the current $k \geq 0$, and consider the next iterate. Stationary condition for the method step is

$$\begin{aligned} & \langle \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k), x - v_{k+1}\rangle + \psi(x) \\ & \geq \; \psi(v_{k+1}) + \tfrac{\mu}{2}\|x - v_{k+1}\|^2, \end{aligned} \tag{3.2.16}$$

for all $x \in \operatorname{dom}\psi$.

Then, we have

$$\begin{aligned}
\phi_{k+1}(x) \quad &\equiv \quad a_{k+1}\big[f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1}\rangle + \psi(x)\big] + \phi_k(x) \\
&\overset{(3.2.14)}{\geq} \quad a_{k+1}\big[f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1}\rangle + \psi(x)\big] + A_k F(x_k)) \\
&\qquad - B_k(x) \\
&\overset{(*)}{\geq} \quad A_{k+1}\big[f(x_{k+1}) + \langle \nabla f(x_{k+1}), \tfrac{a_{k+1}x + A_k x_k}{A_{k+1}} - x_{k+1}\rangle\big] + a_{k+1}\psi(x) \\
&\qquad + A_k\psi(x_k) - B_k(x) \\
&= \quad A_{k+1} f(x_{k+1}) + a_{k+1}\langle \nabla f(x_{k+1}), x - v_{k+1}\rangle + a_{k+1}\psi(x) \\
&\qquad + A_k\psi(x_k) - B_k(x) \\
&= \quad A_{k+1} f(x_{k+1}) \\
&\qquad + a_{k+1}\big[\langle \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k), x - v_{k+1}\rangle + \psi(x)\big] \\
&\qquad + a_{k+1}\langle \nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k)(x_{k+1} - x_k), x - v_{k+1}\rangle \\
&\qquad + A_k\psi(x_k) - B_k(x),
\end{aligned}$$

where $(*)$ refers to convexity of f. Hence,

$$\begin{aligned}
\phi_{k+1}(x) \quad &\overset{(3.2.16),(3.2.4)}{\geq} \quad A_{k+1} f(x_{k+1}) + a_{k+1}\big[\psi(v_{k+1}) + \tfrac{\mu}{2}\|x - v_{k+1}\|^2\big] \\
&\qquad - \tfrac{\mathcal{H}_\nu a_{k+1}^{2+\nu}\|x - v_{k+1}\| \cdot \|v_{k+1} - x_k\|^{1+\nu}}{(1+\nu)A_{k+1}^{1+\nu}} + A_k\psi(x_k) - B_k(x) \\
&\overset{(**)}{\geq} \quad A_{k+1}F(x_{k+1}) + \tfrac{\mu a_{k+1}\|x - v_{k+1}\|^2}{2} + \tfrac{\mu a_{k+1} A_k}{2A_{k+1}}\|x_k - v_{k+1}\|^2 \\
&\qquad - \tfrac{\mathcal{H}_\nu a_{k+1}^{2+\nu}\|x - v_{k+1}\| \cdot \|v_{k+1} - x_k\|^{1+\nu}}{(1+\nu)A_{k+1}^{1+\nu}} + A_k\psi(x_k) - B_k(x) \\
&\equiv \quad A_{k+1}F(x_{k+1}) - B_{k+1}(x),
\end{aligned}$$

where $(**)$ stands for strong convexity of ψ. Thus we have (3.2.14) established for all $k \geq 0$. □

Theorem 3.2.3. *Let $A_k := k^3$, and consequently, $\gamma_k := 1 - \bigl(\frac{k}{k+1}\bigr)^3 = \mathcal{O}\bigl(\frac{1}{k}\bigr)$. Then for the sequence $\{x_k\}_{k\geq 1}$ generated by algorithm (3.2.10), we have*

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \overset{\text{def}}{=} \quad F(x_k) - \tfrac{\phi_k^*}{A_k} \quad \leq \quad \mathcal{O}\bigl(\tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\bigr). \qquad (3.2.17)$$

Proof. First, by convexity of f we have, for all $x \in \operatorname{dom}\psi$

$$\phi_k(x) \quad \leq \quad A_k F(x).$$

Therefore, for the solution x^* of our problem: $F^* = F(x^*)$, it holds

$$F(x_k) - F^* \quad \leq \quad F(x_k) - \tfrac{\phi_k(x^*)}{A_k} \quad \leq \quad \ell_k \quad \overset{\text{def}}{=} \quad F(x_k) - \tfrac{\phi_k^*}{A_k},$$

and this is the first part of (3.2.17).

At the same time, by Lemma 3.2.2, and using boundness of the domain, we have

$$\begin{aligned} \phi_k^* \quad &:= \quad \min_{x \in \operatorname{dom}\psi} \Bigl\{ \phi_k(x) \Bigr\} \quad \overset{(3.2.14)}{\geq} \quad \min_{x \in \operatorname{dom}\psi} \Bigl\{ A_k F(x_k) - B_k(x) \Bigr\} \\ &\geq \quad A_k F(x_k) - \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu} \sum_{i=1}^{k} \tfrac{a_i^{2+\nu}}{A_i^{1+\nu}} \end{aligned}$$

Therefore, for the choice $A_k := k^3$, we finally obtain

$$\begin{aligned} \ell_k \quad &\leq \quad \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)A_k} \sum_{i=1}^{k} \tfrac{a_i^{2+\nu}}{A_i^{1+\nu}} \quad = \quad \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)k^3} \sum_{i=1}^{k} \tfrac{(i^3-(i-1)^3)^{2+\nu}}{i^{3(1+\nu)}} \\ &\leq \quad \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)k^3} \sum_{i=1}^{k} \tfrac{3^{2+\nu} i^{2(2+\nu)}}{i^{3(1+\nu)}} \quad = \quad \tfrac{3^{2+\nu}\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)k^3} \sum_{i=1}^{k} i^{1-\nu} \\ &= \quad \mathcal{O}\bigl(\tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\bigr). \end{aligned}$$

□

For the case $\nu = 1$ (convex functions with Lipschitz continuous Hessian), estimate (3.2.17) gives the convergence rate of the order $\mathcal{O}(\frac{1}{k^2})$. This rate was proven in Theorem 3.1.9 for the general Contracting-Point Tensor Method with $p = 2$.

In accordance to (3.2.17), in order to obtain ε-accuracy in functional residual, $F(x_K) - F^* \leq \varepsilon$, it is enough to perform

$$K \quad = \quad \mathcal{O}\Bigl(\inf_{\nu \in [0,1]} \bigl(\tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{\varepsilon} \bigr)^{1/(1+\nu)} \Bigr) \tag{3.2.18}$$

iterations of algorithm (3.2.10). In [60], there were proposed first *universal* second-order methods (which do not depend on parameters ν and $\mathcal{H}_\nu$ of the problem class), having complexity guarantees of the same order (3.2.18). These methods are based on the Cubic regularization and an adaptive search for estimating the regularization parameter at every iteration (see Section 2.1).

It is important that algorithm (3.2.10) is both universal and affine-invariant. Additionally, convergence result (3.2.17) provides us with a sequence $\{\ell_k\}_{k \geq 1}$ of computable *accuracy certificates*, which can be used as a stopping criterion of the method.

Now, let us assume that the composite component is *strongly convex* with parameter $\mu > 0$. In this situation, we are able to improve convergence estimate (3.2.17), as follows.

Theorem 3.2.4. *Let $A_k := k^5$, and consequently, $\gamma_k := 1 - \left(\frac{k}{k+1}\right)^5 = \mathcal{O}\left(\frac{1}{k}\right)$. Then for the sequence $\{x_k\}_{k \geq 1}$ generated by algorithm (3.2.10), we have*

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \mathcal{O}\left(\frac{\mathcal{H}_\nu \mathscr{D}^\nu}{\mu} \cdot \frac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{2+2\nu}} \right). \tag{3.2.19}$$

Moreover, if the second-order condition number

$$\bar{\omega}_\nu \quad \overset{\text{def}}{=} \quad \left[\frac{\mathcal{H}_\nu \mathscr{D}^\nu}{(1+\nu)\mu} \right]^{\frac{1}{1+\nu}} \tag{3.2.20}$$

is known, then, defining $A_k := (1 + \bar{\omega}_\nu^{-1})^k$, $k \geq 1$, $A_0 := 0$, and $\gamma_k := \frac{1}{1+\bar{\omega}_\nu}$, $k \geq 1$, $\gamma_0 := 1$, we obtain the global linear rate of convergence

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \exp\left(-\frac{k-1}{1+\bar{\omega}_\nu}\right) \cdot \frac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu}. \tag{3.2.21}$$

Proof. Starting from the same reasoning, as in the proof of Theorem 3.2.3, we get

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \overset{\text{def}}{=} \quad F(x_k) - \frac{\phi_k^*}{A_k}.$$

Let us denote by u_k the minimum of the Estimating Function ϕ_k. Thus,

$$\ell_k \quad = \quad F(x_k) - \frac{\phi_k(u_k)}{A_k} \quad \overset{(3.2.14)}{\leq} \quad \frac{1}{A_k} B_k(u_k) \quad \equiv \quad \frac{1}{A_k} \sum_{i=1}^k B_k^{(i)},$$

with

$$\begin{aligned} B_k^{(i)} \quad &\overset{\text{def}}{=} \quad a_i \left[\frac{\mathcal{H}_\nu a_i^{1+\nu} \|u_k - v_i\| \cdot \|x_{i-1} - v_i\|^{1+\nu}}{(1+\nu) A_i^{1+\nu}} - \frac{\mu \|u_k - v_i\|^2}{2} \right] \\ &\qquad - \frac{\mu a_i A_{i-1} \|x_{i-1} - v_i\|^2}{2A_i} \\ &\leq \quad a_i \max_{t \geq 0} \left\{ \frac{\mathcal{H}_\nu a_i^{1+\nu} \|x_{i-1} - v_i\|^{1+\nu} t}{(1+\nu) A_i^{1+\nu}} - \frac{\mu t^2}{2} \right\} \qquad (3.2.22) \\ &\qquad - \frac{\mu a_i A_{i-1} \|x_{i-1} - v_i\|^2}{2A_i} \\ &= \quad \frac{a_i}{2\mu} \left(\frac{\mathcal{H}_\nu a_i^{1+\nu} \|x_{i-1} - v_i\|^{1+\nu}}{(1+\nu) A_i^{1+\nu}} \right)^2 - \frac{\mu a_i A_{i-1} \|x_{i-1} - v_i\|^2}{2A_i}. \end{aligned}$$

Therefore, for the choice $A_k := k^5$, we have

$$\begin{aligned}
\ell_k &\leq \tfrac{1}{A_k}\sum_{i=1}^{k}\tfrac{a_i}{2\mu}\Big(\tfrac{\mathcal{H}_\nu a_i^{1+\nu}\|x_{i-1}-v_i\|^{1+\nu}}{(1+\nu)A_i^{1+\nu}}\Big)^2 \leq \tfrac{\mathcal{H}_\nu^2\mathscr{D}^{2(1+\nu)}}{2\mu(1+\nu)^2A_k}\sum_{i=1}^{k}\tfrac{a_i^{2(1+\nu)+1}}{A_i^{2(1+\nu)}}\\
&= \tfrac{\mathcal{H}_\nu^2\mathscr{D}^{2(1+\nu)}}{2\mu(1+\nu)^2k^5}\sum_{i=1}^{k}\tfrac{(i^5-(i-1)^5)^{2(1+\nu)+1}}{i^{10(1+\nu)}} \leq \tfrac{5^{2(1+\nu)+1}\mathcal{H}_\nu^2\mathscr{D}^{2(1+\nu)}}{2\mu(1+\nu)^2k^5}\sum_{i=1}^{k}i^{2-2\nu}\\
&= \mathcal{O}\big(\tfrac{\mathcal{H}_\nu\mathscr{D}^\nu}{\mu}\cdot\tfrac{\mathcal{H}_\nu\mathscr{D}^{2+\nu}}{k^{2+2\nu}}\big).
\end{aligned}$$

Thus we have justified (3.2.19). To obtain the linear rate (3.2.21), we set

$$A_k \;:=\; (1+\bar{\omega}_\nu^{-1})^k, \qquad k \geq 1,$$

and $A_0 := 0$. So, $a_1 = A_1$ and

$$a_i \;=\; A_i - A_{i-1} \;=\; \bar{\omega}_\nu^{-1}A_{i-1}, \qquad i \geq 2.$$

Therefore, for the values $\{B_k^{(i)}\}_{i=1}^k$, we have

$$B_k^{(1)} \;\leq\; a_1\tfrac{\mathcal{H}_\nu\mathscr{D}^{2+\nu}}{1+\nu} \;=\; A_1\tfrac{\mathcal{H}_\nu\mathscr{D}^{2+\nu}}{1+\nu},$$

and

$$\begin{aligned}
B_k^{(i)} &\overset{(3.2.22)}{\leq} \tfrac{\mathcal{H}_\nu^2\mathscr{D}^{2\nu}\|x_{i-1}-v_i\|^2a_i^{3+2\nu}}{2\mu(1+\nu)^2A_i^{2+2\nu}} - \tfrac{\mu a_iA_{i-1}\|x_{i-1}-v_i\|^2}{2A_i}\\
&= \tfrac{\mu a_iA_{i-1}\|x_{i-1}-v_i\|^2}{2A_i}\Big(\Big[\tfrac{\mathcal{H}_\nu\mathscr{D}^\nu}{(1+\nu)\mu}\Big]^2\tfrac{a_i^{2+2\nu}}{A_i^{1+2\nu}A_{i-1}} - 1\Big)\\
&\leq \tfrac{\mu a_iA_{i-1}\|x_{i-1}-v_i\|^2}{2A_i}\Big(\Big[\tfrac{\mathcal{H}_\nu\mathscr{D}^\nu}{(1+\nu)\mu}\Big]^2\Big[\tfrac{a_i}{A_{i-1}}\Big]^{2(1+\nu)} - 1\Big)\\
&= 0, \qquad 2 \leq i \leq k,
\end{aligned}$$

since by our choice

$$\tfrac{a_i}{A_{i-1}} \;=\; \bar{\omega}_\nu^{-1} \;\overset{(3.2.20)}{=}\; \Big[\tfrac{(1+\nu)\mu}{\mathcal{H}_\nu\mathscr{D}^\nu}\Big]^{\frac{1}{1+\nu}}.$$

Finally, we obtain

$$\begin{aligned} \ell_k \quad &\leq \quad \tfrac{1}{A_k} B_k^{(1)} \quad \leq \quad \tfrac{A_1}{A_k} \cdot \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu} \quad = \quad \tfrac{1}{(1+\bar{\omega}_\nu^{-1})^{k-1}} \cdot \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu} \\ &\leq \quad \exp\bigl(-\tfrac{k-1}{1+\bar{\omega}_\nu}\bigr) \cdot \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu}. \end{aligned}$$

□

Remark 3.2.5. Note that for the strongly convex function with bounded domain, we have, for every $q \geq 2$:

$$\begin{aligned} \langle \psi'(y) - \psi'(x), y - x \rangle \quad &\overset{(3.2.13)}{\geq} \quad \mu \|y - x\|^2 \quad = \quad \tfrac{\mu \|y-x\|^q}{\|y-x\|^{q-2}} \\ &\overset{(3.2.1)}{\geq} \quad \tfrac{\mu \|y-x\|^q}{\mathscr{D}^{q-2}} \quad = \quad \sigma \|y - x\|^q, \end{aligned}$$

with $\sigma_q := \frac{\mu}{\mathscr{D}^{q-2}}$. Therefore, such a function is also *uniformly convex* for arbitrary $q \geq 2$, and definition (3.2.20) of the condition number is consistent with (2.1.8). Namely, it holds:

$$\bar{\omega}_\nu \quad = \quad \bigl(\tfrac{1}{1+\nu} \cdot \omega_\nu\bigr)^{\frac{1}{1+\nu}}.$$

□

According to estimate (3.2.21), in order to get ε-accuracy in function value, it is enough to perform

$$K \quad = \quad \mathcal{O}\bigl((1 + \bar{\omega}_\nu) \cdot \log \tfrac{F(x_0) - F^*}{\varepsilon}\bigr)$$

iterations of the method. Hence, condition number $\bar{\omega}_\nu$ plays the role of the main complexity factor. This rate corresponds to that one of Cubically Regularized Newton Method (Theorem 2.1.11).

At the same time, there exists a second variant of Contracting-Point Newton Method, where the next point is defined by minimization of the full second-order model for the smooth component augmented by the composite term over the *contracted domain* (this explains the names of our methods).

```
Contracting-Point Newton Method, II

Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$.

Iteration $k \geq 0$.

1: Pick up $\gamma_k \in (0,1]$.

2: Denote
   $\bar{S}_k(y) \overset{\mathrm{def}}{=} \begin{cases} \psi(y), & y \in \gamma_k \mathrm{dom}\,\psi + (1-\gamma_k)x_k, \\ +\infty, & \text{else}. \end{cases}$

3: Compute
   $x_{k+1} \in \mathop{\mathrm{Argmin}}_y \Big\{ \langle \nabla f(x_k), y - x_k \rangle + \frac{1}{2}\langle \nabla^2 f(x_k)(y-x_k), y - x_k \rangle + \bar{S}_k(y) \Big\}.$
```

(3.2.23)

Note that algorithm (3.2.10) admits similar representation as well. [1] Both methods produce the same sequences of points when $\psi(\cdot)$ is $\{0,+\infty\}$-indicator of a convex set. Otherwise, they are different.

Using the same contraction technique, it was shown in [116] that the classical Frank-Wolfe algorithm can be extended onto the case of the composite optimization problems. Additionally, the second-order *Contracting Trust-Region method* was proposed, which has the same form as algorithm (3.2.23). However, its convergence rate was established only at the level $\mathcal{O}(\frac{1}{k})$. Here, we improve its rate as follows.

Theorem 3.2.6. *Let $A_k := k^3$ and $\gamma_k := 1 - \big(\frac{k}{k+1}\big)^3 = \mathcal{O}\big(\frac{1}{k}\big)$. Then for the sequence $\{x_k\}_{k \geq 1}$ generated by algorithm (3.2.23), we have*

$$F(x_k) - F^* \quad \leq \quad \ell_k \quad \leq \quad \mathcal{O}\big(\tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\big). \tag{3.2.24}$$

[1] Indeed, it is enough to take $S_k(y) := \gamma_k \psi(x_k + \frac{1}{\gamma_k}(y - x_k))$.

Proof. The proof is very similar to that one for algorithm (3.2.10). First, the stationary condition for one iteration of algorithm (3.2.23) is

$$\begin{aligned} &\langle \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k), x - v_{k+1} \rangle + \tfrac{1}{\gamma_k}\psi\bigl(\gamma_k x + (1-\gamma_k)x_k\bigr) \\ &\geq \quad \tfrac{1}{\gamma_k}\psi(x_{k+1}), \end{aligned} \tag{3.2.25}$$

for all $x \in \operatorname{dom}\psi$ and $k \geq 0$ (compare with (3.2.16)), where

$$v_{k+1} \quad := \quad x_k + \tfrac{1}{\gamma_k}(x_{k+1} - x_k) \quad \in \quad \operatorname{dom}\psi.$$

Then, it is enough to justify by induction the following bound

$$\phi_k(x) \quad \geq \quad A_k F(x_k) - B_k, \qquad x \in \operatorname{dom}\psi, \tag{3.2.26}$$

with $B_k \overset{\text{def}}{=} \frac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{1+\nu}\sum_{i=1}^k \frac{a_i^{2+\nu}}{A_i^{1+\nu}}$. Finally, by convexity of f, we get

$$\begin{aligned} F(x_k) - F^* \qquad &\leq \qquad \ell_k \overset{\text{def}}{=} F(x_k) - \tfrac{\phi_k^*}{A_k} \\ &\overset{(3.2.26)}{\leq} \qquad \tfrac{B_k}{A_k} \quad = \quad \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)A_k}\sum_{i=1}^k \tfrac{a_i^{2+\nu}}{A_i^{1+\nu}} \\ &= \qquad \mathcal{O}\bigl(\tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\bigr), \end{aligned}$$

where the last equation holds from the choice $A_k := k^3$ (see the end of the proof of Theorem 3.2.3). □

This result is identical to Theorem 3.2.10. However, the first algorithm can be accelerated on the class of strongly convex functions (see Theorem 3.2.4). Thus, it seems that the first method is more preferable.

Finally, let us consider an example, when the composite component $\psi(\cdot)$ is an ℓ_p-ball, as in (3.2.2). Then, iterations of the method can be represented as

$$\begin{aligned} x_{k+1} \quad \in \quad x_k + \operatorname*{Argmin}_h \Bigl\{ &\langle \nabla f(x_k), h\rangle + \tfrac{1}{2}\langle \nabla^2 f(x_k)h, h\rangle \; : \\ &\|x_k + \tfrac{1}{\gamma_k}h\|_p \leq \tfrac{\mathscr{D}}{2} \Bigr\}. \end{aligned} \tag{3.2.27}$$

In this form, it looks as a variant of Trust-Region scheme. To solve the subproblem in (3.2.27), we can use Interior Point Methods (e.g. Chapter 5 in [117]). See also [30], for techniques, developed for Trust-Region schemes. Usually, complexity of this step can be estimated as $\mathcal{O}(n^3)$ arithmetic operations, which comes from the cost of computing a suitable factorization for the Hessian matrix. Alternatively, Hessian-free gradient methods can be applied, for computing an inexact step (see [19, 17]).

In Section 4.2 of Chapter 4, we present implementation and the total complexity analysis for the inexact iterations of the method (3.2.10), when each step is computed by the first-order Conditional Gradient Method. In Section 4.3 we study stochastic variants of the Contracting-Point Newton.

3.2.3 Aggregated Second-Order Models

In this section, we propose more advanced second-order models, based on the global lower bound (3.2.7).

Using the same notation as before, consider a sequence of test points $\{x_k : x_k \in \operatorname{dom}\psi\}_{k\geq 0}$ and sequences of coefficients $\{a_k\}_{k\geq 1}$, $\{\gamma_k\}_{k\geq 0}$, satisfying the relations (3.2.12).

Then, we can introduce the following *Quadratic Estimating Functions* (compare with definition (3.2.11)):

$$\begin{aligned} Q_k(x) \;&\overset{\text{def}}{=}\; \sum_{i=0}^{k-1} a_{i+1}\Big[f(x_i) + \langle \nabla f(x_i), x - x_i\rangle \\ &\qquad + \tfrac{\gamma_i}{2}\langle \nabla^2 f(x_i)(x-x_i), x - x_i\rangle + \psi(x)\Big]. \end{aligned}$$

By (3.2.7), we have the main property of Estimating Functions being satisfied. Namely, for all $x \in \operatorname{dom}\psi$

$$\begin{aligned} A_k F(x) \;&\overset{(3.2.7)}{\geq}\; Q_k(x) - \textstyle\sum_{i=0}^{k-1} \frac{a_{i+1}\gamma_i^{1+\nu}\mathcal{H}_\nu \|x-x_i\|^{2+\nu}}{(1+\nu)(2+\nu)} \\ &\overset{(3.2.1)}{\geq}\; Q_k(x) - \tfrac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)(2+\nu)} \textstyle\sum_{i=0}^{k-1} a_{i+1}\gamma_i^{1+\nu} \\ &=: \; Q_k(x) - \tfrac{C_k}{2}. \end{aligned} \tag{3.2.28}$$

Therefore, if we would be able to guarantee for our test points the relation

$$Q_k^* \;\overset{\text{def}}{=}\; \min_x Q_k(x) \;\geq\; A_k F(x_k) - \tfrac{C_k}{2}, \tag{3.2.29}$$

then we could immediately obtain the global convergence in function value.

Fortunately, relation (3.2.29) can be achieved by simple iterations.

Aggregating Newton Method

```
Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$.

Iteration $k \geq 0$.

1: Pick up $a_{k+1} > 0$.

2: Set $A_{k+1} = A_k + a_{k+1}$ and $\gamma_k = \frac{a_{k+1}}{A_{k+1}}$.        (3.2.30)

3: Update Estimating Function
     $Q_{k+1} \equiv Q_k(x) + a_{k+1}\big[f(x_k) + \langle \nabla f(x_k), x - x_k \rangle$
          $+ \frac{\gamma_k}{2}\langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle + \psi(x)\big]$.

4: Compute $v_{k+1} \in \mathop{\mathrm{Argmin}}\limits_x Q_{k+1}(x)$.

5: Set $x_{k+1} = x_k + \gamma_k(v_{k+1} - x_k)$.
```

Clearly, the most complicated part of this process is Step 4, which is computation of the minimum of Estimating Function. However, the complexity of this step remains the same, as that one for the Contracting-Point Newton Method.

We obtain the following convergence result.

Theorem 3.2.7. *For the sequence $\{x_k\}_{k \geq 1}$ generated by algorithm (3.2.30), relation (3.2.29) is satisfied. Consequently, for the choice $A_k := k^3$, we obtain*

$$\begin{array}{rcl} F(x_k) - F^* & \overset{(3.2.28)}{\leq} & F(x_k) - \frac{Q_k^*}{A_k} + \frac{C_k}{2A_k} \quad \overset{(3.2.29)}{\leq} \quad \frac{C_k}{A_k} \\ & \leq & \mathcal{O}\Big(\frac{\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\Big). \end{array} \tag{3.2.31}$$

Proof. Let us establish the relation (3.2.29) by induction. It obviously holds for $k = 0$. Assume that it is proven for the current iterate $k \geq 0$, and consider the next step:

$$\begin{aligned}
&Q_{k+1}(v_{k+1}) \\
&\quad\equiv \quad Q_k(v_{k+1}) + a_{k+1}\big[f(x_k) + \langle \nabla f(x_k), v_{k+1} - x_k\rangle \\
&\qquad\quad + \tfrac{\gamma_k}{2}\langle \nabla^2 f(x_k)(v_{k+1} - x_k), v_{k+1} - x_k\rangle + \psi(v_{k+1})\big] \\
&\quad\overset{(3.2.29)}{\geq} \quad A_k F(x_k) - \tfrac{C_k}{2} + a_{k+1}\big[f(x_k) + \langle \nabla f(x_k), v_{k+1} - x_k\rangle \\
&\qquad\quad + \tfrac{\gamma_k}{2}\langle \nabla^2 f(x_k)(v_{k+1} - x_k), v_{k+1} - x_k\rangle + \psi(v_{k+1})\big] \\
&\quad= \quad A_k \psi(x_k) - \tfrac{C_k}{2} + A_{k+1}\big[f(x_k) + \gamma_k\langle \nabla f(x_k), v_{k+1} - x_k\rangle \\
&\qquad\quad + \tfrac{\gamma_k^2}{2}\langle \nabla^2 f(x_k)(v_{k+1} - x_k), v_{k+1} - x_k\rangle] + a_{k+1}\psi(v_{k+1}) \\
&\quad= \quad A_k \psi(x_k) - \tfrac{C_k}{2} + A_{k+1}\big[f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle \\
&\qquad\quad + \tfrac{1}{2}\langle \nabla^2 f(x_k)(x_{k+1} - x_k), x_{k+1} - x_k\rangle] + a_{k+1}\psi(v_{k+1}).
\end{aligned}$$

Hence,

$$\begin{aligned}
&Q_{k+1}(v_{k+1}) \\
&\quad\overset{(3.2.5)}{\geq} \quad A_k \psi(x_k) - \tfrac{C_k}{2} + A_{k+1}\big[f(x_{k+1}) - \tfrac{\mathcal{H}_\nu \|x_{k+1} - x_k\|^{2+\nu}}{(1+\nu)(2+\nu)}\big] \\
&\qquad\quad + a_{k+1}\psi(v_{k+1}) \\
&\quad\geq \quad A_{k+1} f(x_{k+1}) - \tfrac{a_{k+1}\gamma_k^{1+\nu}\mathcal{H}_\nu \mathscr{D}^{2+\nu}}{(1+\nu)(2+\nu)} + A_{k+1}\psi(x_{k+1}) - \tfrac{C_k}{2} \\
&\quad= \quad A_{k+1} F(x_{k+1}) - \tfrac{C_{k+1}}{2}.
\end{aligned}$$

Thus, we have (3.2.29) justified for all $k \geq 0$. □

Now, for the accuracy certificate we have new expression $\bar{\ell}_k := F(x_k) - \frac{Q_k^*}{A_k} + \frac{C_k}{2A_k}$. The value of Q_k^* is available within the method directly. However, in order to compute $\bar{\ell}_k$ in practice, some estimate for C_k, which depends on the Hölder constant $\mathcal{H}_\nu$ and on the diameter $\mathscr{D}$ of the domain, is required. Note, that for the given choice of coefficients $A_k := k^3$, we have $a_k = \mathcal{O}(k^2)$ and $\gamma_k = \mathcal{O}(\frac{1}{k})$. Therefore, new information enters into the model with increasing weights, which seems natural.

3.2.4 Experiments

Let us discuss our computational results for the problem of training logistic regression model, regularized by ℓ_2-ball constraints. In this problem, the smooth part of the objective is

$$f(x) \quad := \quad \frac{1}{M} \sum_{i=1}^{M} f_i(x),$$

with $f_i(x) := \log(1 + \exp(\langle a_i, x \rangle))$. The composite part is the indicator of a Euclidean ball,

$$\psi(x) \quad := \quad \begin{cases} 0, & \|x\|_2 := \Bigl(\sum_{i=1}^n |x^{(i)}|^2 \Bigr)^{1/2} \quad \leq \quad \frac{\mathscr{D}}{2}, \\ +\infty, & \text{else.} \end{cases}$$

Diameter $\mathscr{D}$ plays the role of regularization parameter, while vectors $\{a_i : a_i \in \mathbb{R}^n\}_{i=1}^M$ are determined by the dataset[2].

We compare the performance of the Contracting Newton Method (algorithm 3.2.10) and the Aggregating Newton Method (algorithm 3.2.30) with first-order optimization schemes: Frank-Wolfe algorithm [52], the classical Gradient Method, and the Fast Gradient Method [114]. For the latter two we use a line search at each iteration, to estimate the Lipschitz constant. In all methods, we ensure monotonicity in the function value. The results are shown in Figures 3.5 – 3.8.

We see that for bigger $\mathscr{D}$, it becomes harder to solve the optimization problem. Second-order methods demonstrate good performance both in terms of the iterations, and the total computational time. [3]

According to these graphs, our second-order algorithms can be more efficient when solving ill-conditioned problems, producing the better solution within a given computational time.

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[3] CPU time was evaluated on a machine with Intel Core i5 CPU, 1.6GHz; 8 GB RAM. All methods have been implemented in C++. Operation system: macOS 10.15. Compiler: Clang 12.0.0. The source code can be found at https://github.com/doikov/contracting-newton/

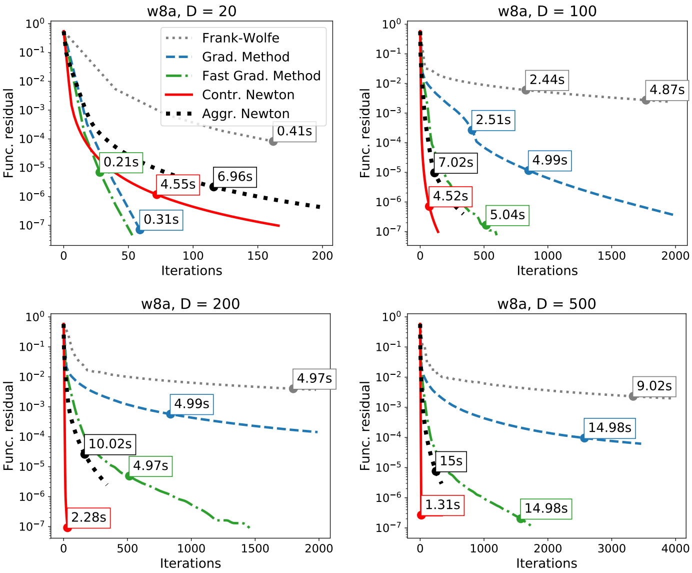


Figure 3.5: Logistic regression, *w8a* ($M = 49749, n = 300$).

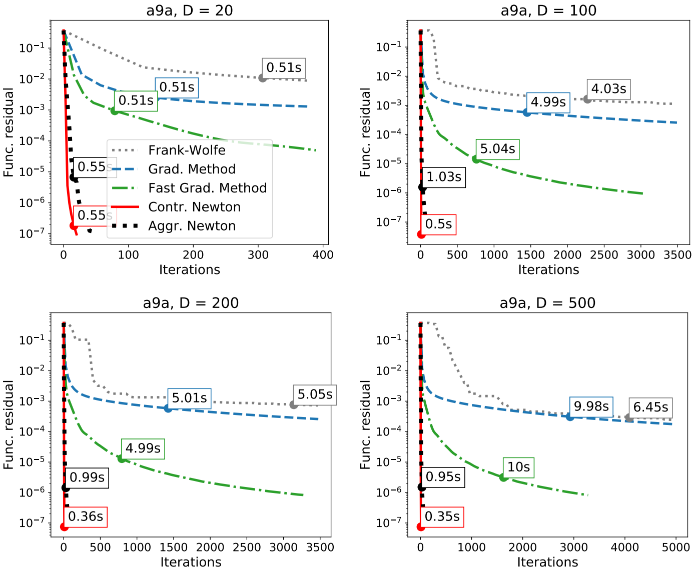


Figure 3.6: Logistic regression, *a9a* ($M = 32561, n = 123$).

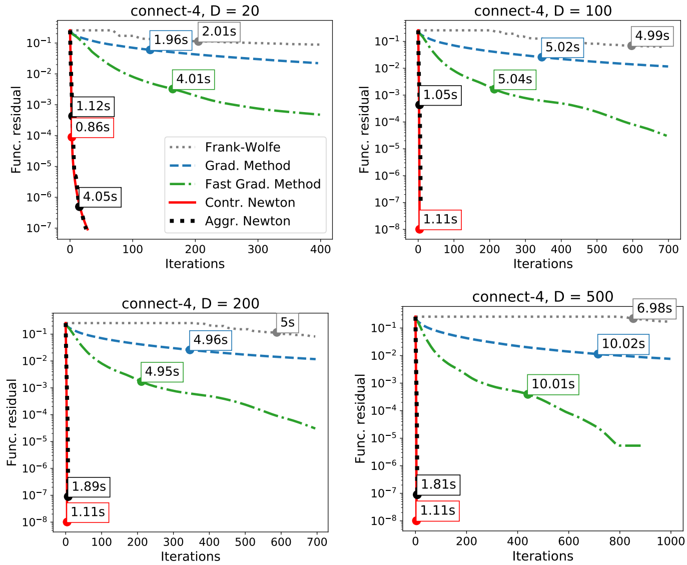


Figure 3.7: Logistic regression, *connect-4* $(M = 67557, n = 126)$.

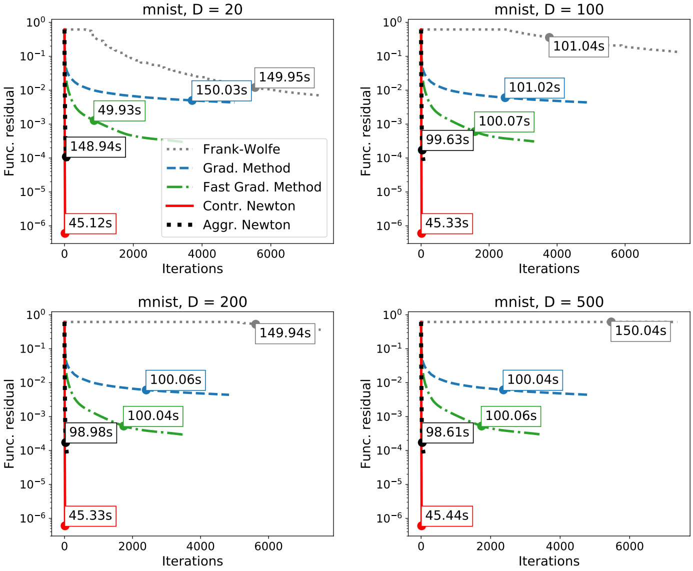


Figure 3.8: Logistic regression, *mnist* $(M = 60000, n = 780)$.

Comparing the Contracting Newton and the Aggregating Newton methods, we conclude that both algorithms show reasonably good performance in practice. The latter one works a bit slower. However, the aggregation of the Hessians helps to improve numerical stability. In Figure 3.9, we demonstrate the influence of the parameter of inner accuracy (EPS), which is the bound for the dual problem that we use in our subsolver, on the convergence of the algorithms. We see much more robust behaviour for the Aggregating Newton Method, while the first algorithm can potentially stop, or even start to diverge, if the parameter is chosen in a wrong way.

To compute one step of our second-order methods for this task, we need to solve subproblem (3.2.27) for $p = 2$. This is minimization of quadratic function over the standard Euclidean ball. First, we compute *tridiagonal* decomposition of the Hessian (it requires $\mathcal{O}(n^3)$ arithmetical operations). Then, we solve the dual to our subproblem (which is maximization of one-dimensional concave function) by classical Newton iterations (the cost of each iteration is $\mathcal{O}(n)$). For more details, see Chapter 7 in [30].

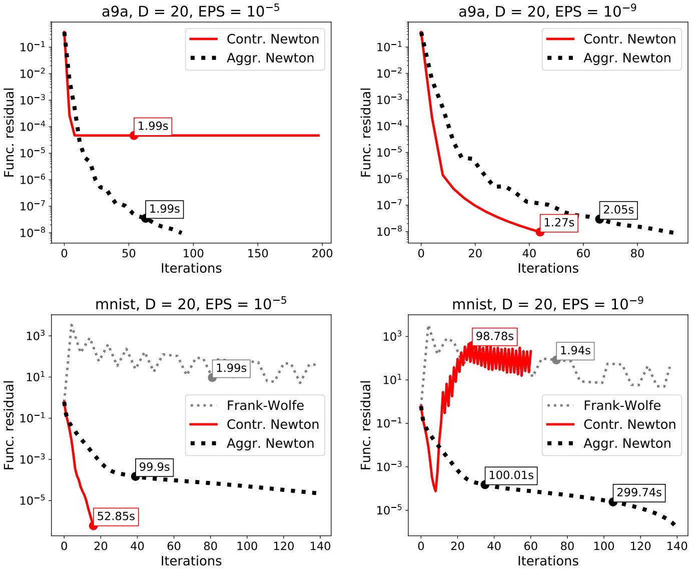


Figure 3.9: Influence of the parameter of inner accuracy.

3.2.5 Discussion

Let us discuss complexity estimates which we established in this part of the thesis. For the basic versions of the Contracting Newton Method, we have the global convergence in the functional residual of the form

$$F(x_k) - F^* \quad \leq \quad \mathcal{O}\big(\tfrac{H_\nu \mathscr{D}^{2+\nu}}{k^{1+\nu}}\big).$$

Note that the complexity parameter H_ν depends only on the *variation* of the Hessian (in arbitrary norm). It can be much smaller than the maximal eigenvalue of the Hessian, which typically appears in the rates of first-order methods. It is important that our algorithms are free from using the norms or any other particular parameters of the problem class.

At the same time, the arithmetic complexity of one step of our methods for simple sets can be estimated as the sum of the cost of computing the Hessian, and $\mathcal{O}(n^3)$ additional operations necessary to compute a suitable factorization of the matrix. For example, the cost of computing the gradient of Logistic Regression is $\mathcal{O}(Mn)$, and the Hessian is $\mathcal{O}(Mn^2)$, where M is the dataset size. Hence, it is preferable to use our algorithms with exact steps in the situation when M is much bigger than n.

When M is very big and it is expensive to compute the full gradient and Hessian at each iteration, we can use stochastic versions of our methods. We present them in Section 4.3.

3.3 Contracting Proximal Methods

Let us present an acceleration of our Contracting-Point methods by employing the *proximal idea.* It appears that having a suitable prox-function for our problem, we can achieve an accelerated rate of convergence. However, the methods are no longer affine-invariant.

Iterations of the basic Proximal-Point algorithm for minimizing a convex function $f : \mathrm{dom}\, f \to \mathbb{R}$ are as follows:

$$x_{k+1} \quad = \quad \underset{x}{\mathrm{argmin}} \Big\{ a_{k+1} f(x) + \tfrac{1}{2}\|x_k - x\|^2 \Big\}, \qquad k \geq 0, \tag{3.3.1}$$

where $\|\cdot\|$ is the Euclidean norm, and $\{a_k\}_{k\geq 0}$ is a sequence of positive coefficients.

The regularized objective in (3.3.1) is *strongly convex.* Therefore, we can hope that computing an (inexact) proximal step is usually simpler than solving the initial problem. In Section 2.2.4, we have already discussed the possibility of using the fast local convergence of high-order methods for solving the proximal subproblem.

When $f \in C^{1,1}(\mathbb{E})$ (differentiable functions with Lipschitz continuous gradient), we can set all values of the coefficients a_k equal to a positive constant. It gives a global sublinear rate of convergence of the iterations (3.3.1) in functional residual of the order $\mathcal{O}(1/k)$. This is also the rate of the Gradient Method.

For the same class of functions, we can get a faster rate of convergence of the order $\mathcal{O}(1/k^2)$ using the Fast Gradient Method [107]. It is the best possible rate achievable for the first-order black-box optimization [106]. An accelerated variant of the Proximal-Point algorithm with the optimal rate of convergence was proposed in [66] (see also [140, 93, 94, 72] for extensions and some applications).

In this part of the thesis, we present a new family of proximal-type algorithms for smooth convex optimization called *Contracting Proximal Methods*, which includes an accelerated algorithm from [66] as a particular case. It provides a systematic way for constructing faster proximal accelerated methods for high-order optimization. Thus, for the class of convex functions, whose p-th derivative is Lipschitz continuous ($p \geq 1$), our new methods achieve the $\mathcal{O}(1/k^{p+1})$-rate of convergence for the outer proximal iterations, while the inner subproblems can be efficiently solved up to desired accuracy by the basic Tensor Method.

The main difference between Contracting Proximal Methods and the classical approach (3.3.1) consists in employing the *contracted* objective function and using the *Bregman divergence* (notation $\beta_d(x;y)$) instead of the usual Euclidean norm. The exact form of our method for minimizing a convex function $f : \mathrm{dom}\, f \to \mathbb{R}$ is very simple:

$$\boxed{\begin{array}{rcl} v_{k+1} & = & \operatorname*{argmin}\limits_{x} \Big\{ A_{k+1} f\big(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\big) + \beta_d(v_k; x) \Big\}, \\[2ex] x_{k+1} & = & \tfrac{a_{k+1}v_{k+1} + A_k x_k}{A_{k+1}}, \qquad k \geq 0. \end{array}} \tag{3.3.2}$$

Thus, we use a sequence of auxiliary points $\{v_k\}_{k \geq 0}$, and the scaling coefficients $A_k \overset{\mathrm{def}}{=} \sum_{i=1}^k a_i$.

Let us illustrate the basic idea behind this construction by the simplest *Euclidean setting*, when $\beta_d(x;y) \equiv \frac{1}{2}\|x-y\|^2$. We are going to ensure at each iteration $k \geq 0$ the following condition, for all $x \in \mathrm{dom}\, f$:

$$\tfrac{1}{2}\|x_0 - x\|^2 + A_k f(x) \quad \geq \quad \tfrac{1}{2}\|v_k - x\|^2 + A_k f(x_k). \tag{3.3.3}$$

A direct consequence of (3.3.3) is the global convergence bound

$$f(x_k) - f^* \quad \leq \quad \tfrac{\|x_0 - x^*\|^2}{2A_k}. \tag{3.3.4}$$

We can propagate inequality (3.3.3) to the next iteration by a trivial observation:

$$\begin{array}{rcl} \tfrac{1}{2}\|x_0 - x\|^2 + A_{k+1} f(x) & = & \tfrac{1}{2}\|x_0 - x\|^2 + A_k f(x) + a_{k+1} f(x) \\[2ex] & \overset{(3.3.3)}{\geq} & \tfrac{1}{2}\|v_k - x\|^2 + A_k f(x_k) + a_{k+1} f(x) \\[2ex] & \geq & \tfrac{1}{2}\|v_k - x\|^2 + A_{k+1} f\big(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\big) \\[2ex] & \equiv & h_{k+1}(x), \end{array}$$

where the last inequality is due to convexity of the objective. Note that the first step of Contracting Proximal Method (3.3.2) is defined exactly as follows:

$$v_{k+1} \quad = \quad \operatorname*{argmin}_{x \in \mathbb{E}} h_{k+1}(x). \tag{3.3.5}$$

Hence, by strong convexity of $h_{k+1}(\cdot)$, we finally justify that

$$\begin{array}{rcl} h_{k+1}(x) & \geq & h_{k+1}(v_{k+1}) + \frac{1}{2}\|v_{k+1} - x\|^2 \\ \\ & \geq & A_{k+1} f(x_{k+1}) + \frac{1}{2}\|v_{k+1} - x\|^2. \end{array}$$

Thus, for the Euclidean setting, iteration (3.3.2) immediately results in the convergence guarantee (3.3.4). However, we are still free in the choice of coefficients $\{a_k\}_{k\geq 1}$. The only reason for bounding their growth consists in keeping the complexity of the optimization problem (3.3.5) at the acceptable level.[4] For $f \in C^{1,1}(\mathbb{E})$, the recommended choice of a_{k+1} corresponds to the quadratic equation [107]:

$$a_{k+1}^2 \quad = \quad \tfrac{1}{L_1}(a_{k+1} + A_k). \tag{3.3.6}$$

It is easy to see, that this choice results in the optimal $\mathcal{O}(1/k^2)$-rate of convergence for the method. On the other hand, it makes the *condition number* of the problem (3.3.5) equal to an absolute constant. Let us assume for simplicity, that f is two times continuously differentiable. Then, in view of the presence of the regularization term, $\nabla^2 h_{k+1}(x) \succeq B$. On the other hand,

$$\nabla^2 h_{k+1}(x) \quad = \quad B + \tfrac{a_{k+1}^2}{A_{k+1}} \nabla^2 f\bigl(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\bigr) \quad \overset{(3.3.6)}{\preceq} \quad 2B.$$

Hence, we are able to solve the problem (3.3.5) very efficiently by a usual gradient method (see the details in Section 3.3.3).

It is remarkable that exactly the same reasoning justifies the accelerated versions of *all* high-order Tensor Methods ($p \geq 2$). The only difference consists in the degree of the proximal term, which must be compatible with the order of optimization scheme used for solving the problem (3.3.5).

Our first-order Contracting Proximal Method for Euclidean setting (described above) produces the same sequence of points as the accelerated Proximal-Point algorithm from [66]. However, now we can employ also the Bregman divergence, which sometimes is more suitable for the geometry of our problem and ensures faster convergence.

In what follows, we recall the notion of Bregman divergence and state some of its properties in Section 3.3.1.

[4] Hence, these bounds should take into account the efficiency of the auxiliary minimization scheme used for solving the problem (3.3.5).

In Section 3.3.2, we introduce a general Contracting Proximal Method (formulated as algorithm (3.3.22)). We present its convergence analysis for a problem in composite form and arbitrary Bregman divergence. We study both convex and strongly convex cases under inexactness in proximal steps. Theorem 3.3.4 specifies how the parameters of the algorithm and inner accuracy affect the convergence rate.

In Section 3.3.3, we discuss implementation of one iteration of our method, under assumption that p-th derivative ($p \geq 1$) of the smooth part of the objective is Lipschitz continuous. We present fully-defined optimization scheme (algorithm (3.3.58)), with incorporated steps of the Tensor Methods of a certain degree. Resulting algorithm achieves the accelerated rate of convergence, with an additional logarithmic factor for the number of total oracle calls. Final complexity estimate for this scheme is given by Theorem 3.3.11 and Theorem 3.3.12.

Section 3.3.4 contains numerical experiments. Section 3.3.5 has some final remarks.

3.3.1 Bregman Divergence

We use some arbitrary (possibly non-Euclidean) norm $\| \cdot \|$ on space $\mathbb{E}$ and define the dual norm $\| \cdot \|_*$ on $\mathbb{E}^*$ in the standard way,

$$\|s\|_* \overset{\mathrm{def}}{=} \sup_{h \in \mathbb{E}} \{ \langle s, h \rangle : \|h\| \leq 1 \}, \qquad s \in \mathbb{E}^*.$$

Let us fix arbitrary differentiable strictly convex function $d : \mathrm{dom}\, \psi \to \mathbb{R}$, which we call *prox function.* Then, we denote by $\beta_d(x; y)$ the corresponding *Bregman divergence* [15], centered at x:

$$\beta_d(x; y) \overset{\mathrm{def}}{=} d(y) - d(x) - \langle \nabla d(x), y - x \rangle.$$

Recall that function d is called *uniformly convex* of degree $p+1$ (with respect to the norm $\| \cdot \|$) with constant $\sigma_{p+1}(d) > 0$, if it holds for all $x, y \in \mathrm{dom}\, d$:

$$\beta_d(x; y) \geq \tfrac{\sigma_{p+1}(d)}{p+1} \|x - y\|^{p+1}. \tag{3.3.7}$$

The main example, which naturally appears in the Tensor Methods and which we use in Section 3.3.3, is the following prox function.

Example 3.3.1.

$$d(x) \equiv \tfrac{1}{p+1} \|x - x_0\|^{p+1},$$

for some $p \geq 1$. For the Euclidean norm this prox function is *uniformly convex* of degree $p+1$ with constant 2^{1-p} (see Lemma 2.1.4), so it holds:

$$\beta_d(x;y) \quad \geq \quad \tfrac{2^{1-p}}{p+1}\|x-y\|^{p+1}, \qquad x, y \in \mathbb{E}. \tag{3.3.8}$$

□

For more examples of available prox functions see [7, 95].

The definition of Bregman divergence can be extended to the case of a nondifferentiable function ψ by specifying a particular subgradient $\psi'(x) \in \partial\psi(x)$:

$$\beta_\psi(x, \psi'(x); y) \quad \overset{\mathrm{def}}{=} \quad \psi(y) - \psi(x) - \langle \psi'(x), y - x \rangle.$$

However, we will use simpler notation $\beta_\psi(x;y)$ if no ambiguity arise.

We say that function ψ is *strongly convex with respect to d* (see [95]) with constant $\sigma_d(\psi) > 0$, if it holds for all $x, y \in \mathrm{dom}\,\psi$ and for all $\psi'(x) \in \partial\psi(x)$

$$\beta_\psi(x, \psi'(x); y) \quad \geq \quad \sigma_d(\psi)\beta_d(x;y). \tag{3.3.9}$$

Inequality (3.3.9) always holds with $\sigma_d(\psi) = 0$ just by convexity. An interesting illustration of this concept is given by a regularized Taylor polynomial of degree 3 for a convex function.

Example 3.3.2. Let $f : \mathrm{dom}\, f \to \mathbb{R}$ be convex, with Lipschitz continuous third derivative ($L_3 < +\infty$).

Consider the following regularization of its Taylor approximation, for some $\tau > 1$:

$$g(y) \quad \equiv \quad \Omega_3(f, x; y) + \tfrac{\tau^2 L_3}{8}\|y - x\|^4.$$

Then, for the Euclidean norm, the function $g(\cdot)$ is strongly convex with respect to the following prox function (see Lemma 4 in [118]):

$$d(h) \quad \equiv \quad \tfrac{1}{2}\left(1 - \tfrac{1}{\tau}\right) D^2 f(x)[h]^2 + \tfrac{\tau(\tau-1)L_3}{8}\|h\|^4.$$

□

Let us summarize some basic properties of Bregman divergence, which follow directly from its definition. For any pair f_1, f_2 of convex functions and all $x, y \in \mathrm{dom}\,(f_1 + f_2)$ we have

$$\beta_{a_1 f_1 + a_2 f_2}(x;y) \quad = \quad a_1\beta_{f_1}(x;y) + a_2\beta_{f_2}(x;y), \qquad a_1, a_2 \geq 0. \tag{3.3.10}$$

For any linear function $\ell(x) = a + \langle g, x \rangle$ we have

$$\beta_\ell(x;y) \quad = \quad 0. \tag{3.3.11}$$

Therefore, from (3.3.10) and (3.3.11) we conclude, that

$$\beta_f(x;y) \quad = \quad \beta_d(x;y), \tag{3.3.12}$$

when $f(y) = \beta_d(z;y)$ for some fixed z. Now, consider the following simple but general construction, which we use in a core of our analysis. Let h be a regularized composite objective:

$$h(y) \quad = \quad g(y) + a\psi(y) + \mu\beta_d(z;y), \qquad a, \mu \geq 0,$$

where g and ψ are arbitrary closed convex functions, and ψ is strongly convex with respect to d for some constant $\sigma_d(\psi) \geq 0$. Then, for every $x, y \in \operatorname{dom} h$ and every $h'(x) \in \partial h(x)$, we have that

$$\begin{array}{rcl} \beta_h(x;y) & = & h(y) - h(x) - \langle h'(x), y - x \rangle \\ & \overset{(3.3.10),(3.3.12)}{=} & \beta_g(x;y) + a\beta_\psi(x;y) + \mu\beta_d(x;y) \\ & \geq & (a\sigma_d(\psi) + \mu)\beta_d(x;y). \end{array} \tag{3.3.13}$$

In particular, for the exact minimum $T = \operatorname*{argmin}_{y \in \mathbb{E}} h(y)$, we have

$$h(y) \quad \geq \quad h(T) + (a\sigma_d(\psi) + \mu)\beta_d(T;y). \tag{3.3.14}$$

3.3.2 Contracting Proximal Methods

In our general scheme for solving the composite optimization problem,

$$\min_x \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\},$$

we are going to maintain the following inequality, for every $x \in \operatorname{dom} \psi$ and $k \geq 0$:

$$\mu_0\beta_d(x_0;x) + A_k F(x) \quad \geq \quad \mu_k\beta_d(v_k;x) + A_k F(x_k) + C_k(x), \tag{3.3.15}$$

where $\{x_k\}_{k\geq 0}$ and $\{v_k\}_{k\geq 0}$ are sequences of points from $\mathrm{dom}\,\psi$, $\{A_k\}_{k\geq 0}$ is a sequence of increasing numbers:

$$a_{k+1} \overset{\text{def}}{=} A_{k+1} - A_k \;>\; 0, \qquad A_0 \;=\; 0,$$

and $\{\mu_k\}_{k\geq 0}$ is a sequences of nondecreasing proximal coefficients:

$$\mu_{k+1} \;\;\geq\;\; \mu_k, \qquad \mu_0 \;\;>\;\; 0.$$

We would prefer functions $C_k(x)$ be as big as possible. Thus, if it happens to be $C_k(x^*) \geq 0$ for all $k \geq 1$, then from (3.3.15) we have a convergence guarantee:

$$F(x_k) - F^* \;\;\leq\;\; \tfrac{\mu_0 \beta_d(x_0, x^*)}{A_k}, \qquad k \geq 1,$$

and the rate of convergence is determined by the growth of coefficients A_k towards infinity. However, in general $C_k(x)$ may have arbitrary sign.

Let us discus a simple possibility for propagating relation (3.3.15) to the next iteration.

$$\begin{array}{rl} & \mu_0\beta_d(x_0;x) + A_{k+1}F(x) \\ = & \mu_0\beta_d(x_0;x) + A_kF(x) + a_{k+1}F(x) \\ \overset{(3.3.15)}{\geq} & \mu_k\beta_d(v_k;x) + A_kF(x_k) + a_{k+1}F(x) + C_k(x) \\ \geq & \mu_k\beta_d(v_k;x) + A_{k+1}f\big(\tfrac{a_{k+1}x + A_kx_k}{A_{k+1}}\big) \\ & \qquad + a_{k+1}\psi(x) + A_k\psi(x_k) + C_k(x), \end{array} \tag{3.3.16}$$

where the last inequality is due to convexity of f. Let us consider a contracted objective with regularizer from the last step:

$$h_{k+1}(x) \;\overset{\text{def}}{=}\; A_{k+1}f\big(\tfrac{a_{k+1}x + A_kx_k}{A_{k+1}}\big) + a_{k+1}\psi(x) + \mu_k\beta_d(v_k;x). \tag{3.3.17}$$

This function is strongly convex with respect to $d(\cdot)$ with parameter

$$\sigma_d(h_{k+1}) \;\;\geq\;\; \mu_{k+1} \overset{\text{def}}{=} a_{k+1}\sigma_d(\psi) + \mu_k. \tag{3.3.18}$$

If we are able to compute the *exact minimum*

$$T \quad = \quad \operatorname*{argmin}_{x \in \mathbb{E}} h_{k+1}(x), \tag{3.3.19}$$

then by (3.3.14) we see that

$$\begin{aligned} & h_{k+1}(x) + A_k \psi(x_k) \\ & \qquad \geq \quad h_{k+1}(T) + \mu_{k+1}\beta_d(T;x) + A_k\psi(x_k) \\ & \qquad = \quad A_{k+1} f\big(\tfrac{a_{k+1}T + A_k x_k}{A_{k+1}}\big) + a_{k+1}\psi(T) + \mu_k \beta_d(v_k;T) \\ & \qquad\qquad\quad + \mu_{k+1}\beta_d(T;x) + A_k\psi(x_k) \\ & \qquad \geq \quad A_{k+1} F\big(\tfrac{a_{k+1}T + A_k x_k}{A_{k+1}}\big) + \mu_k\beta_d(v_k;T) + \mu_{k+1}\beta_d(T;x). \end{aligned}$$

And it is natural to set $v_{k+1} = T$ and

$$x_{k+1} \quad \overset{\text{def}}{=} \quad \tfrac{a_{k+1}v_{k+1} + A_k x_k}{A_{k+1}}. \tag{3.3.20}$$

Thus we would obtain guarantee (3.3.15) for the next step, with

$$C_{k+1}(x) \quad \equiv \quad C_k(x) + \mu_k\beta_d(v_k;v_{k+1}) \; \equiv \; \textstyle\sum_{i=1}^{k} \mu_i \beta_d(v_i;v_{i+1}) \geq 0.$$

Now, instead of computing the exact minimum (3.3.19), let us relax $v_{k+1} \in \operatorname{dom}\psi$ to be a point with a *small norm of subgradient*:

$$\|s\|_* \;\leq\; \delta_{k+1}, \quad \text{for some} \quad s \in \partial h_{k+1}(v_{k+1}). \tag{3.3.21}$$

Note that condition (3.3.21) can be easily verified algorithmically since in composite setting we are able to compute points with small subgradient of h_{k+1}.

Thus, we come to the following general scheme.

Contracting Proximal Method

```
Initialization.
Choose $x_0 \in \operatorname{dom} \psi$, $\mu_0 > 0$. Set $v_0 = x_0$, $A_0 = 0$.

Iteration $k \geq 0$.

1: Choose $a_{k+1} > 0$. Set $A_{k+1} = A_k + a_{k+1}$.

2: Denote contracted objective with regularizer:

$h_{k+1}(x) \;=\; A_{k+1} f\big(\frac{a_{k+1}x + A_k x_k}{A_{k+1}}\big) + a_{k+1}\psi(x) + \mu_k \beta_d(v_k; x)$.

3: Choose accuracy $\delta_{k+1} \geq 0$.

4: Find $v_{k+1} \in \operatorname{dom} \psi$ s.t. $\exists s \in \partial h_{k+1}(v_{k+1}) : \; \|s\|_* \leq \delta_{k+1}$.

5: Set $x_{k+1} = \frac{a_{k+1}v_{k+1} + A_k x_k}{A_{k+1}}$.

6: Set $\mu_{k+1} = \mu_k + a_{k+1}\sigma_d(\psi) = \mu_0 + A_{k+1}\sigma_d(\psi)$ .
```
(3.3.22)

At this moment, we need one additional assumption. It relates the dual norm $\| \cdot \|_*$ (used at Step 4) with the Bregman divergence $\beta_d(v; x)$.

Assumption 3.3.3. For some $p \geq 1$, prox-function $d(\cdot)$ is uniformly convex of degree $p+1$ with respect to the primal norm $\|\cdot\|$ with parameter $\sigma_{p+1}(d) > 0$ (see inequality (3.3.7)).

Let us write down the convergence guarantees of the method.

Theorem 3.3.4 (Convergence of Contracting Proximal Method)**.** *Let Assumption* (3.3.3) *hold. Then for algorithm* (3.3.22) *at all iterations $k \geq 0$ we have:*

$$\begin{aligned} & A_k \left(F(x_k) - F^*\right) + \mu_k \beta_d(v_k; x^*) + \sum_{i=1}^{k} \mu_i \beta_d(v_{i-1}; v_i) \\ \leq \;\; & R_k(p, \delta), \end{aligned} \tag{3.3.23}$$

where

$$R_k(p,\delta) \quad \overset{\text{def}}{=} \quad \left(\left(\mu_0 \beta_d(x_0;x^*)\right)^{\frac{p}{p+1}} + \left(\frac{p+1}{\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} \sum_{i=1}^{k} \frac{\delta_i}{\mu_i^{1/(p+1)}} \right)^{\frac{p+1}{p}}.$$

Proof. First, let us ensure by induction in $k \geq 0$ that the following inequality holds:

$$\begin{array}{rl} & A_k\left(F(x_k) - F(x)\right) + \mu_k \beta_d(v_k;x) + \sum\limits_{i=1}^{k} \mu_i \beta_d(v_{i-1};v_i) \\ \leq & \mu_0 \beta_d(x_0;x) + \sum\limits_{i=1}^{k} \langle s_i, v_i - x\rangle, \qquad x \in \operatorname{dom}\psi, \end{array} \tag{3.3.24}$$

where $s_i \in \partial h_i(v_i)$. It is obviously true for $k=0$. Suppose that it holds for some $k \geq 0$ and consider the case $k+1$. Note that (3.3.24) is exactly (3.3.15) with

$$C_k(x) \quad \equiv \quad \sum_{i=1}^{k} \Big[\mu_i \beta_d(v_{i-1};v_i) + \langle s_i, x - v_i\rangle \Big].$$

Therefore, we have

$$\begin{array}{rl} & \mu_0\beta_d(x_0;x) + A_{k+1}F(x) \\ \overset{(3.3.16)}{\geq} & h_{k+1}(x) + A_k\psi(x_k) + C_k(x) \\ \overset{(3.3.13)}{\geq} & h_{k+1}(v_{k+1}) + \langle s_{k+1}, x - v_{k+1}\rangle + \mu_{k+1}\beta_d(v_{k+1};x) \\ & \quad + A_k\psi(x_k) + C_k(x) \\ = & A_{k+1}f(x_{k+1}) + a_{k+1}\psi(v_{k+1}) + \mu_{k+1}\beta_d(v_{k+1};x) \\ & \quad + A_k\psi(x_k) + C_{k+1}(x) \\ \geq & A_{k+1}F(x_{k+1}) + \mu_{k+1}\beta_d(v_{k+1};x) + C_{k+1}(x). \end{array}$$

This is (3.3.24) for the next step.

Now, plugging $x \equiv x^*$ into (3.3.24) and taking into account nonnegativity of all terms in the left-hand side, we get

$$\mu_k \beta_d(v_k; x^*) \quad \leq \quad \mu_0 \beta_d(x_0; x^*) + \sum_{i=1}^{k} \langle s_i, v_i - x^* \rangle.$$

Now, we need to estimate the right-hand side from above. Using uniform convexity (3.3.7), we conclude that for every $k \geq 0$

$$\begin{array}{rl} \tfrac{\mu_k \sigma_{p+1}(d)}{p+1} \| v_k - x^* \|^{p+1} & \\ \leq & \mu_0 \beta_d(x_0; x^*) + \sum\limits_{i=1}^{k} \| s_i \|_* \cdot \| v_i - x^* \| \\ \overset{(3.3.21)}{\leq} & \mu_0 \beta_d(x_0; x^*) + \sum\limits_{i=1}^{k} \delta_i \| v_i - x^* \| \quad \equiv \quad \alpha_k. \end{array} \tag{3.3.25}$$

In order to finish the proof, it is enough to bound from above the value α_k, for which we have the following recurrence:

$$\alpha_k \quad = \quad \alpha_{k-1} + \delta_k \| v_k - x^* \| \quad \overset{(3.3.25)}{\leq} \quad \alpha_{k-1} + \delta_k \left(\tfrac{p+1}{\mu_k \sigma_{p+1}(d)} \right)^{\frac{1}{p+1}} \alpha_k^{\frac{1}{p+1}}.$$

Dividing both sides by $\alpha_k^{\frac{1}{p+1}}$ and using monotonicity of this sequence, we get

$$\alpha_k^{\frac{p}{p+1}} \quad \leq \quad \tfrac{\alpha_{k-1}}{\alpha_k^{1/(p+1)}} + \delta_k \left(\tfrac{p+1}{\mu_k \sigma_{p+1}(d)} \right)^{\frac{1}{p+1}} \quad \leq \quad \alpha_{k-1}^{\frac{p}{p+1}} + \delta_k \left(\tfrac{p+1}{\mu_k \sigma_{p+1}(d)} \right)^{\frac{1}{p+1}}.$$

Finally, from the last inequality we obtain

$$\alpha_k \quad \leq \quad \left(\alpha_0^{\frac{p}{p+1}} + \left(\tfrac{p+1}{\sigma_{p+1}(d)} \right)^{\frac{1}{p+1}} \sum_{i=1}^{k} \tfrac{\delta_i}{\mu_i^{1/(p+1)}} \right)^{\frac{p+1}{p}},$$

which is the right-hand side of (3.3.23). □

We see that accuracies δ_k for subgradients of the subproblems appears in $R_k(p, \delta)$ in an additive form, weighted by the coefficients $\mu_k^{-\frac{1}{p+1}}$. They should be chosen in a way making the right-hand side of (3.3.23) small enough. Let us consider the simplest case, when all δ_k are the same.

Corollary 3.3.5. *Let $\delta_k = \delta > 0$ for all $k \geq 1$. Assume that the coefficients*

A_k grow sublinearly:

$$A_k \quad \geq \quad c k^{p+1}, \quad k \geq 1, \tag{3.3.26}$$

with some constant $c > 0$. Then for every

$$\begin{aligned} k \quad &\geq \quad \left(\frac{\mu_0\beta_d(x_0;x_*)}{c\varepsilon}\right)^{\frac{1}{p+1}} 2^{\frac{1}{p}} \qquad \textit{and} \\ \delta \quad &\leq \quad \frac{(c\varepsilon)^{\frac{p}{p+1}}}{2}\left(\frac{\mu_0\sigma_{p+1}(d)}{p+1}\right)^{\frac{1}{p+1}} \end{aligned} \tag{3.3.27}$$

we have

$$R_k(p,\delta) \quad \leq \quad \varepsilon A_k. \tag{3.3.28}$$

Consequently, by (3.3.23) we have $F(x_k) - F^* \leq \varepsilon$.

Proof. Indeed,

$$\left(\frac{\mu_0\beta_d(x_0;x^*)}{A_k}\right)^{\frac{p}{p+1}} \quad \overset{(3.3.26)}{\leq} \quad \left(\frac{\mu_0\beta_d(x_0;x^*)}{c}\right)^{\frac{p}{p+1}} k^p \quad \overset{(3.3.27)}{\leq} \quad \frac{\varepsilon^{\frac{p}{p+1}}}{2},$$

and

$$\begin{aligned} \frac{\left(\frac{p+1}{\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}}}{A_k^{\frac{p}{p+1}}} \sum_{i=1}^{k} \frac{\delta_i}{\mu_i^{1/(p+1)}} \quad &\leq \quad \frac{\left(\frac{p+1}{\mu_0\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} k\delta}{A_k^{\frac{p}{p+1}}} \\ &\overset{(3.3.26)}{\leq} \quad \frac{\left(\frac{p+1}{\mu_0\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} \delta}{c^{\frac{p}{p+1}} k^{p+1}} \\ &\leq \quad \frac{\left(\frac{p+1}{\mu_0\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} \delta}{c^{\frac{p}{p+1}}} \quad \overset{(3.3.27)}{\leq} \quad \frac{\varepsilon^{\frac{p}{p+1}}}{2}. \end{aligned}$$

Summing up these two inequalities we obtain (3.3.28). □

Corollary 3.3.6. *Let $\delta_k = \delta > 0$ for all $k \geq 1$. Let the coefficients A_k grow linearly:*

$$A_k \quad \geq \quad A_1 \exp\bigl(\omega(k-1)\bigr), \quad k \geq 1, \tag{3.3.29}$$

with some constant $0 < \omega \leq 1$ and initial $A_1 > 0$. Then for every

$$k \quad \geq \quad 1 + \frac{1}{\omega}\log\left(\frac{\mu_0\beta_d(x_0;x^*)}{A_1\varepsilon} 2^{(p+1)/p}\right) \tag{3.3.30}$$

and

$$\delta \quad \leq \quad \frac{(A_1\varepsilon)^{\frac{p}{p+1}}\omega}{2} \cdot \frac{p}{p+1} \cdot \left(\frac{\mu_0\sigma_{p+1}(d)}{p+1}\right)^{\frac{1}{p+1}} \tag{3.3.31}$$

we have

$$R_k(p,\delta) \quad \leq \quad \varepsilon A_k. \tag{3.3.32}$$

Consequently, by (3.3.23) *we have* $F(x_k) - F^* \leq \varepsilon$.

Proof. Indeed,

$$\left(\frac{\mu_0\beta_d(x_0;x^*)}{A_k}\right)^{\frac{p}{p+1}} \overset{(3.3.29)}{\leq} \left(\frac{\mu_0\beta_d(x_0;x^*)}{A_1\exp\bigl(\omega(k-1)\bigr)}\right)^{\frac{p}{p+1}} \overset{(3.3.31)}{\leq} \frac{\varepsilon^{\frac{p}{p+1}}}{2}.$$

Now, note that the following inequality holds for all $x \geq 0$:

$$\exp(x) \quad \geq \quad 1 + x. \tag{3.3.33}$$

Therefore,

$$\begin{aligned} \frac{A_k^{\frac{p}{p+1}}}{k} \quad & \overset{(3.3.29)}{\geq} \quad \frac{A_1^{\frac{p}{p+1}}\exp\bigl(\frac{p}{p+1}\omega(k-1)\bigr)}{k} \\ & \overset{(3.3.33)}{\geq} \quad \frac{A_1^{\frac{p}{p+1}}\Bigl(1+\frac{p}{p+1}\omega(k-1)\Bigr)}{k} \quad > \quad \frac{p}{p+1}A_1^{\frac{p}{p+1}}\omega. \end{aligned} \tag{3.3.34}$$

And we obtain

$$\begin{aligned} \frac{\left(\frac{p+1}{\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}}}{A_k^{\frac{p}{p+1}}} \sum_{i=1}^{k} \frac{\delta_i}{\mu_i^{1/(p+1)}} \quad & \leq \quad \frac{\left(\frac{p+1}{\mu_0\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} k\delta}{A_k^{\frac{p}{p+1}}} \\ & \overset{(3.3.34)}{<} \quad \frac{\left(\frac{p+1}{\mu_0\sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} (p+1)\delta}{A_1^{\frac{p}{p+1}} p\omega} \\ & \overset{(3.3.31)}{\leq} \quad \frac{\varepsilon^{\frac{p}{p+1}}}{2}. \end{aligned}$$

□

Estimates (3.3.27) and (3.3.31) show that the bound for the inner accuracy δ has a reasonable dependency on ε, which is the absolute accuracy required for the initial problem. Thus, in both cases, on step 4 of the algorithm we need to find a point v_{k+1} with subgradient $s \in \partial h_{k+1}(v_{k+1})$:

$$\|s\|_* \;\leq\; \mathcal{O}\Big(\varepsilon^{\frac{p}{p+1}}\Big) \quad \Leftrightarrow \quad \|s\|_*^{\frac{p+1}{p}} \;\leq\; \mathcal{O}\big(\varepsilon\big).$$

This is a reachable goal, especially for methods minimizing $h_{k+1}(\cdot)$ with a linear rate of convergence.

In practice, it may be reasonable not to use very small inner accuracy on a first stage, but to decrease it over the iterations. Then, the following simple choice of $\{\delta_k\}_{k\geq 0}$ can work.

Corollary 3.3.7. *Let us define* $\delta_k \equiv \frac{c}{k^s}$ *with fixed absolute constants* $c > 0$ *and* $s > 1$*. Then,*

$$\sum\limits_{i=1}^{k} \delta_i \quad \overset{(1.3.9)}{\leq} \quad \tfrac{cs}{s-1}.$$

Therefore, we have

$$R_k(p,\delta) \quad \leq \quad \left(\big(\mu_0 \beta_d(x_0; x^*)\big)^{\frac{p}{p+1}} + \left(\tfrac{p+1}{\mu_0 \sigma_{p+1}(d)}\right)^{\frac{1}{p+1}} \tfrac{cs}{s-1} \right)^{\frac{p+1}{p}}.$$

3.3.3 Applications of Tensor Methods

Let us incorporate steps of the basic Tensor Method (1.5.1) into algorithm (3.3.22) for solving the corresponding inner subproblem (3.3.19). From now on, we restrict our attention to the Euclidean norm: $\|x\| \equiv \langle Bx, x\rangle^{1/2}$, $x \in \mathbb{E}$.

Assumption 3.3.8. For fixed $p \geq 1$, $f \in C^{p,p}(\mathrm{dom}\,\psi)$. So the p-th derivative of the smooth component of the objective is Lipschitz continuous with some constant $0 < L_p(f) < +\infty$.

For this setup, we use the following simple prox function:

$$d(x) \quad \equiv \quad \tfrac{1}{p+1}\|x - x_0\|^{p+1}. \tag{3.3.35}$$

Thus, the choice of prox function (3.3.35) is strictly related to the preferable degree $p \geq 1$ of smoothness of function f.

We recall that the Taylor approximation $\Omega_p(f,x;y)$ of function f around the point $x \in \mathrm{dom}\, f$ is defined as

$$\Omega_p(f,x;y) \quad \overset{\text{def}}{=} \quad f(x) + \sum\limits_{i=1}^{p} \tfrac{1}{i!} D^i f(x)[y-x]^i.$$

We have the following bounds (Lemma 1.3.7): for all $x, y \in \operatorname{dom} \psi$,

$$|f(y) - \Omega_p(f, x; y)| \quad \leq \quad \tfrac{L_p(f)}{(p+1)!} \|y - x\|^{p+1}, \tag{3.3.36}$$

$$\|\nabla f(y) - \nabla_y \, \Omega_p(f, x; y)\|_* \quad \leq \quad \tfrac{L_p(f)}{p!} \|y - x\|^p. \tag{3.3.37}$$

Let us look at our regularized objective $h_{k+1}(\cdot)$ which need to be minimized at every step $k \geq 0$:

$$\begin{aligned} & h_{k+1}(x) \\ & = \underbrace{A_{k+1} f\left(\frac{a_{k+1}x + A_k x_k}{A_{k+1}}\right)}_{\overset{\text{def}}{=} g_{k+1}(x)} + \underbrace{a_{k+1}\psi(x) + \mu_k \beta_d(v_k; x)}_{\overset{\text{def}}{=} \phi_{k+1}(x)}. \end{aligned} \tag{3.3.38}$$

This is a sum of two convex functions: smooth component g_{k+1}, and possibly nonsmooth but simple component ϕ_{k+1}, which is strongly convex with respect to d.

Let us drop unnecessary indices and consider the subproblem in a general form:

$$\min_x \Bigl\{ h(x) \equiv g(x) + \phi(x) \Bigr\}, \tag{3.3.39}$$

with g having bounded Lipschitz constant for some $p \geq 1$: $0 < L_p(g) < +\infty$. Since we assume the objective to be strongly convex with respect to d from (3.3.35) with parameter $\sigma_d(h) > 0$, for every $x, y \in \operatorname{dom} h$ and all $h'(x) \in \partial h(x)$ we have:

$$\begin{aligned} h(y) - h(x) - \langle h'(x), y - x \rangle \quad & \geq \quad \sigma_d(h) \beta_d(x; y) \\ & \overset{(3.3.8)}{\geq} \quad \tfrac{\sigma_d(h) 2^{1-p}}{p+1} \|y - x\|^{p+1}. \end{aligned} \tag{3.3.40}$$

Bound (3.3.36) motivates us to define the following point:

$$\begin{aligned} T_M(h; x) \quad \overset{\text{def}}{=} \quad & \operatorname*{argmin}_y \Bigl\{ \Omega_p(g, x; y) \\ & \qquad + \tfrac{M}{(p+1)!} \|y - x\|^{p+1} + \phi(y) \Bigr\}, \end{aligned} \tag{3.3.41}$$

and consider the following iteration process:

$$\boxed{z_{t+1} \quad = \quad T_M(h; z_t), \qquad t \geq 0} \tag{3.3.42}$$

Let us mention some properties of point $T \equiv T_M(h;x)$. Its characteristic condition is as follows,

$$\phi'(T) \quad \overset{\text{def}}{=} \quad -\nabla_y \Omega_p(g,x;T) - \tfrac{M}{p!}\|T-x\|^{p-1}B(T-x) \quad \in \quad \partial\phi(T).$$

This inclusion justifies notation $h'(T) \overset{\text{def}}{=} \nabla g(T) + \phi'(T) \quad \in \quad \partial h(T)$. In order to work with this object, we need to use Lemma 2.2.1. In terms of our current objective (3.3.39), we have, setting $M = pL_p(g)$:

$$\langle h'(T), x - T\rangle \quad \geq \quad \left(\tfrac{p!}{(p+1)L_p(g)}\right)^{\frac{1}{p}} \cdot \|h'(T)\|_*^{\frac{p+1}{p}}. \tag{3.3.43}$$

Next, by (2.2.19)we have the following description of the global behaviour of the method, for all $x, y \in \operatorname{dom} h$:

$$h(T_M(x)) \quad \leq \quad h(y) + \tfrac{(p+1)L_p(g)}{(p+1)!}\|y-x\|^{p+1}, \tag{3.3.44}$$

when $M = pL_p(g)$. Now, we are ready to prove a convergence result on the iteration process (3.3.42), for the norm of the subgradients.

Theorem 3.3.9. *Let $M = pL_p(g)$. Then, for every $t \geq 0$ and $y \in \operatorname{dom} h$ we have*

$$\begin{aligned} \|h'(z_{t+2})\|_*^{\frac{p+1}{p}} \quad & \leq \quad \exp\left(-t \cdot \min\left\{1, \left[\tfrac{p!\,\sigma_d(h)2^{1-p}}{(p+1)L_p(g)}\right]^{\frac{1}{p}}\right\} \cdot \tfrac{p}{p+1}\right) \\ & \qquad \cdot \left(\tfrac{(p+1)L_p(g)}{p!}\right)^{\frac{1}{p}} \\ & \qquad \cdot \left(h(y) - h^* + \tfrac{L_p(g)}{p!}\|y - z_0\|^{p+1}\right). \end{aligned} \tag{3.3.45}$$

Proof. Let us consider the point $z_{t+1} = T_M(z_t)$. By (3.3.44), we have

$$h(z_{t+1}) \quad \leq \quad h(y) + \tfrac{L_p(g)}{p!}\|y - z_t\|^{p+1}, \tag{3.3.46}$$

for any $y \in \operatorname{dom} h$. Denote $x_h^* \overset{\text{def}}{=} \operatorname{argmin}_y h(y)$, and consider $y = z_t +$

$\alpha(x_h^* - z_t)$ for $\alpha \in [0,1]$. Then we have

$$\begin{aligned} & h(z_{t+1}) - h^* \\ & \leq \quad h(z_t) - h^* - \alpha\,(h(z_t) - h^*) + \alpha^{p+1} \tfrac{L_p(g)}{p!} \|x_h^* - z_t\|^{p+1} \\ & \overset{(3.3.40)}{\leq} \left(1 - \alpha + \alpha^{p+1} \tfrac{(p+1)L_p(g)}{p!\,\sigma_d(h) 2^{1-p}}\right) \cdot (h(z_t) - h^*)\,. \end{aligned} \tag{3.3.47}$$

The minimum of the right-hand side is attained at

$$\alpha^* \quad = \quad \min\left\{1, \left[\tfrac{p!\,\sigma_d(h)2^{1-p}}{(p+1)L_p(g)}\right]^{\frac{1}{p}}\right\}.$$

Plugging it into (3.3.47) gives

$$\begin{aligned} h(z_{t+1}) - h^* \quad & \leq \quad \left(1 - \alpha^* \tfrac{p}{p+1}\right) \cdot (h(z_t) - h^*) \\ & \leq \quad \exp\left(-\alpha^* \tfrac{p}{p+1}\right) \cdot (h(z_t) - h^*). \end{aligned} \tag{3.3.48}$$

Therefore, for every $t \geq 0$ we have

$$\begin{aligned} h(z_{t+1}) - h^* \quad & \overset{(3.3.48)}{\leq} \quad \exp\left(-t\alpha^* \tfrac{p}{p+1}\right) \cdot (h(z_1) - h^*) \\ & \overset{(3.3.46)}{\leq} \quad \exp\left(-t\alpha^* \tfrac{p}{p+1}\right) \cdot \left(h(y) - h^* + \tfrac{L_p(g)}{p!} \|y - z_0\|^{p+1}\right), \end{aligned}$$

for every $y \in \operatorname{dom} h$. It remains to use (3.3.43) and finish the proof:

$$\begin{aligned} h(z_{t+1}) - h^* \quad & \geq \quad h(z_{t+1}) - h(z_{t+2}) \\ & \geq \quad \langle h'(z_{t+2}), z_{t+1} - z_{t+2} \rangle \\ & \overset{(3.3.43)}{\geq} \quad \left(\tfrac{p!}{(p+1)L_p(g)}\right)^{\frac{1}{p}} \cdot \|h'(z_{t+2})\|_*^{\frac{p+1}{p}}. \qquad \square \end{aligned}$$

Thus, we can see that applying the Tensor Method (3.3.42) of degree $p \geq 1$ on Step 4 of the general Contracting Proximal Method (algorithm (3.3.22)), we obtain fast linear convergence for the norms of subgradients. Hence, we can estimate the total number of inner steps t_k at iteration $k \geq 0$ as follows.

Corollary 3.3.10. *Let us minimize function $h_{k+1}(\cdot)$ by iterations:*

$$z_{t+1} = T_M(h_{k+1}; z_t), \quad t \geq 0,$$

using $M := pL_p(g_{k+1})$ and $z_0 := v_k$. Then we have

$$\| h'_{k+1}(z_{t_k}) \|_* \quad \leq \quad \delta_{k+1},$$

for

$$t_k \quad \geq \quad 2 + \max\left\{1, \tfrac{\ell_{k+1}}{\bar{\mu}_{k+1}}\right\} \cdot \tfrac{p+1}{p} \cdot \log\left(\tfrac{\ell_{k+1}\, D_{k+1}}{\delta_{k+1}^{\frac{p+1}{p}}} \right), \tag{3.3.49}$$

where

$$\ell_{k+1} \quad \overset{\text{def}}{=} \quad \left(\tfrac{(p+1)L_p(g_{k+1})}{p!} \right)^{\frac{1}{p}}, \qquad \bar{\mu}_{k+1} \quad \overset{\text{def}}{=} \quad \left(\mu_{k+1} 2^{1-p} \right)^{\frac{1}{p}}, \tag{3.3.50}$$

and

$$\begin{aligned} D_{k+1} \quad & \overset{\text{def}}{=} \quad A_k(F(x_k) - F^*) + \mu_k \beta_d(v_k; x^*) \\ & \qquad + \quad \left(\tfrac{\ell_{k+1}}{\bar{\mu}_{k+1}} \right)^p \beta_d(v_k; x^*) \\ & \overset{(3.3.23)}{\leq} \quad R_k(p, \delta) \cdot \left(1 + \tfrac{1}{\mu_0} \left(\tfrac{\ell_{k+1}}{\bar{\mu}_{k+1}} \right)^p \right). \end{aligned} \tag{3.3.51}$$

Proof. By definition, for all $x \in \operatorname{dom} \psi$, we have

$$\begin{aligned} & h_{k+1}(x) + A_k \psi(x_k) \\ = \quad & A_{k+1} f\left(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}} \right) + a_{k+1}\psi(x) + \mu_k \beta_d(v_k; x) + A_k \psi(x_k) \\ \geq \quad & A_{k+1} F\left(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}} \right) + \mu_k \beta_d(v_k; x) \quad \geq \quad A_{k+1} F^*. \end{aligned}$$

Therefore,

$$-h^*_{k+1} - A_k \psi(x_k) \quad \leq \quad -A_{k+1} F^*. \tag{3.3.52}$$

Then for $y \equiv x^* \overset{\mathrm{def}}{=} \operatorname{argmin}_y F(y)$ we obtain

$$\begin{array}{rl} & h_{k+1}(y) - h^*_{k+1} + \frac{L_p(g_{k+1})}{p!}\|y - z_0\|^{p+1} \\ = & h_{k+1}(x^*) - h^*_{k+1} + \frac{L_p(g_{k+1})}{p!}\|x^* - v_k\|^{p+1} \\ = & A_{k+1} f\big(\frac{a_{k+1}x^* + A_k x_k}{A_{k+1}}\big) + a_{k+1}\psi(x^*) - h^*_{k+1} + \mu_k \beta_d(v_k; x^*) \\ & \quad + \quad \frac{L_p(g_{k+1})}{p!}\|x^* - v_k\|^{p+1} \\ \leq & a_{k+1}F^* + A_k F(x_k) - h^*_{k+1} - A_k\psi(x_k) + \mu_k\beta_d(v_k; x^*) \\ & \quad + \quad \frac{L_p(g_{k+1})}{p!}\|x^* - v_k\|^{p+1} \\ \overset{(3.3.52)}{\leq} & A_k(F(x_k) - F^*) + \mu_k\beta_d(v_k; x^*) + \frac{L_p(g_{k+1})}{p!}\|x^* - v_k\|^{p+1} \\ \overset{(3.3.8)}{\leq} & D_{k+1}. \end{array}$$

It remains to use this bound together with (3.3.45) and the following estimation of strong convexity parameter: $\sigma_d(h_{k+1}) \overset{(3.3.18)}{\geq} \mu_{k+1}$. □

By representation (3.3.38), we have a simple relations between Lipschitz constants of the derivatives for function $g_{k+1}(\cdot)$ and $f(\cdot)$:

$$L_p(g_{k+1}) \;=\; \frac{a_{k+1}^{p+1}}{A_{k+1}^p} L_p(f), \quad p \geq 1. \tag{3.3.53}$$

Therefore, we can control the condition number of our objective. Indeed, by (3.3.49), the main complexity factor in minimization process for $h_{k+1}(\cdot)$ is the ratio

$$\frac{\ell_{k+1}}{\bar{\mu}_{k+1}} \;\equiv\; \Big(\frac{(p+1)L_p(g_{k+1})}{p!\,2^{1-p}\mu_{k+1}}\Big)^{\frac{1}{p}} \overset{(3.3.53),(3.3.18)}{=} \Big(\frac{(p+1)2^{p-1}a_{k+1}^{p+1}L_p(f)}{p!\,A_{k+1}^p(\mu_0 + A_{k+1}\sigma_d(\psi))}\Big)^{\frac{1}{p}}.$$

We are able to keep this ratio small by applying an appropriate growth strategy for coefficients A_k.

Let us consider two cases: $\sigma_d(\psi) = 0$ and $\sigma_d(\psi) > 0$.

1. $\sigma_d(\psi) = 0$. Let us choose $c \equiv \frac{p!\,\mu_0}{2^{p-1}(p+1)^{p+2}L_p(f)}$ and $a_k \equiv c(p+1)k^p$.

Then we have

$$A_k \;=\; c(p+1)\sum_{i=1}^{k} i^p \;\geq\; c(p+1)\int\limits_0^k x^p dx \;=\; ck^{p+1},$$

and we get

$$\frac{a_{k+1}^{p+1}}{A_{k+1}^{p}} \;\leq\; c(p+1)^{p+1} \;=\; \frac{p!\,\mu_0}{2^{p-1}(p+1)L_p(f)}. \tag{3.3.54}$$

Thus we obtain

$$\frac{\ell_{k+1}}{\bar{\mu}_{k+1}} \;=\; \left(\frac{a_{k+1}^{p+1}}{A_{k+1}^{p}} \cdot \frac{2^{p-1}(p+1)L_p(f)}{p!\,\mu_0}\right)^{\frac{1}{p}} \overset{(3.3.54)}{\leq} 1. \tag{3.3.55}$$

2. $\sigma_d(\psi) > 0$. For $k = 0$ we pick $a_1 \equiv c(p+1)$ as in the previous case. Now consider $k \geq 1$. Denote

$$\omega \;\overset{\text{def}}{=}\; \min\left\{\left(\frac{\sigma_d(\psi)p!}{L_p(f)(p+1)2^{p-1}}\right)^{\frac{1}{p+1}}, \frac{1}{2}\right\} \tag{3.3.56}$$

and choose a_{k+1} from the equation

$$\frac{a_{k+1}}{A_{k+1}} \;=\; \frac{a_{k+1}}{a_{k+1}+A_k} \;=\; \omega \quad\Leftrightarrow\quad a_{k+1} \;=\; \omega(1-\omega)^{-1}A_k.$$

Therefore

$$\begin{aligned}\frac{\ell_{k+1}}{\bar{\mu}_{k+1}} \;&\leq\; \left(\frac{a_{k+1}^{p+1}}{A_{k+1}^{p+1}} \cdot \frac{L_p(f)(p+1)2^{p-1}}{p!\,\sigma_d(\psi)}\right)^{\frac{1}{p}} \\ &=\; \omega \cdot \left(\frac{L_p(f)(p+1)2^{p-1}}{p!\,\sigma_d(\psi)}\right)^{\frac{1}{p+1}} \;\leq\; 1.\end{aligned} \tag{3.3.57}$$

Thus, in both cases, at every upper-level step we need to perform a logarithmic number of iterations of the inner method, multiplied by a small constant.

We are ready to specify the whole optimization procedure.

Contracting Proximal Tensor Method

```
Initialization.
Choose $x_0 \in \mathrm{dom}\,\psi$, $\mu_0 > 0$, $\delta > 0$. Set $v_0 = x_0$, $A_0 = 0$.
Fix $d(x) = \frac{1}{p+1}\|x - x_0\|^{p+1}$.
Set $c = \frac{p!\,\mu_0}{2^{p-1}(p+1)^{p+2}L_p(f)}$, $\omega = \min\{\big(\frac{\sigma_d(\psi)p!}{L_p(f)(p+1)2^{p-1}}\big)^{\frac{1}{p+1}}, \frac{1}{2}\}$.

Iteration $k \geq 0$.

1: If $k = 0$ or $\omega = 0$, then choose $a_{k+1} = c(p+1)(k+1)^p$.
   Else choose $a_{k+1} = \omega(1-\omega)^{-1}A_k$.

2: Set $A_{k+1} = A_k + a_{k+1}$.

3: Denote contracted objective with regularizer:
      $g_{k+1}(x) = A_{k+1}f\big(\frac{a_{k+1}x + A_k x_k}{A_{k+1}}\big)$,
      $\phi_{k+1}(x) = a_{k+1}\psi(x) + \mu_k\beta_d(v_k; x)$,
      $h_{k+1}(x) = g_{k+1}(x) + \phi_{k+1}(x)$.

4: Solve inner subproblem by Tensor Method.
4-a: Initialization. Set $z_0 = v_k, t_k = 0, M = pL_p(f)\frac{a_{k+1}^{p+1}}{A_{k+1}^p}$.
4-b: Compute $z_{t_k+1} = T_M(h_{k+1}, z_{t_k})$. Set $t_k = t_k + 1$.
4-c: If $\|h'_{k+1}(z_{t_k})\|_* \leq \delta$, then set $v_{k+1} = z_{t_k}$ and go to 5.
     Else go to 4-b.

5: Set $x_{k+1} = \frac{a_{k+1}v_{k+1} + A_k x_k}{A_{k+1}}$.

6: Set $\mu_{k+1} = \mu_k + a_{k+1}\sigma_d(\psi) = \mu_0 + A_{k+1}\sigma_d(\psi)$.
```

(3.3.58)

Let us present global complexity bounds for this method in convex and strongly convex cases.

Theorem 3.3.11 (Convex Case). *Let for a given $\varepsilon > 0$, we choose:*

$$\delta \;=\; \left(\frac{p!\,\varepsilon}{L_p(f)}\right)^{\frac{p}{p+1}} \frac{\mu_0}{2^p(p+1)^{p+1}}.$$

Then, in order to achieve $F(x_K) - F^* \leq \varepsilon$ it is enough to perform

$$K \;=\; \left\lfloor 1 + 2^{\frac{1}{p}} \left(\frac{2^{p-1}(p+1)^{p+2} L_p(f)\,\beta_d(x_0;x^*)}{\varepsilon\, p!}\right)^{\frac{1}{p+1}} \right\rfloor \tag{3.3.59}$$

iterations of algorithm (3.3.58). *The total number of oracle calls $N_K \overset{\text{def}}{=} \sum_{k=1}^{K} t_k$ is bounded as*

$$N_K \;\leq\; K \cdot \Big(3 + \tfrac{p+1}{p} \log\Big(4\big(1 + \tfrac{1}{\mu_0}\big)(p+1)^{\frac{1}{p}} K^p\Big)\Big). \tag{3.3.60}$$

Proof. Estimate (3.3.59) follows from (3.3.27), by substituting the value $c = \frac{p!\mu_0}{2^{p-1}(p+1)^{p+2}L_p(f)}$. Now, let us prove (3.3.60). By (3.3.49), we have

$$\begin{aligned} t_k \quad &\leq \quad 3 + \max\Big\{1, \tfrac{\ell_{k+1}}{\bar{\mu}_{k+1}}\Big\} \cdot \tfrac{p+1}{p} \cdot \log\Big(\tfrac{\ell_{k+1} D_{k+1}}{\delta^{\frac{p+1}{p}}}\Big) \\ &\overset{(3.3.55),(3.3.51)}{\leq} \quad 3 + \tfrac{p+1}{p} \cdot \log\Big(\tfrac{\mu_0^{1/p}(1+\mu_0^{-1})R_k(p,\delta)}{\delta^{\frac{p+1}{p}}}\Big). \end{aligned}$$

In order to finish the proof, we need to bound the value under the logarithm.

By the choice of a_k, we have an upper bound for A_k:

$$A_k \;=\; c(p+1)\sum_{i=1}^{k} i^p \;\leq\; c(p+1)\int\limits_0^{k+1} x^p dx \;=\; c(k+1)^{p+1}. \tag{3.3.61}$$

Therefore, for every $0 \leq k \leq K$:

$$\begin{aligned} \frac{R_k(p,\delta)}{\delta^{\frac{p+1}{p}}} \quad &= \quad \left(\frac{(\mu_0\beta_d(x_0;x^*))^{\frac{p}{p+1}}}{\delta} + \big(\tfrac{(p+1)2^{p-1}}{\mu_0}\big)^{\frac{1}{p+1}} k\right)^{\frac{p+1}{p}} \\ &\leq \quad \left(\frac{(\mu_0\beta_d(x_0;x^*))^{\frac{p}{p+1}}}{\delta} + \big(\tfrac{(p+1)2^{p-1}}{\mu_0}\big)^{\frac{1}{p+1}} K\right)^{\frac{p+1}{p}} \\ &= \quad \left(\Big(\frac{L_p(f)\beta_d(x_0;x^*)}{p!\,\varepsilon}\Big)^{\frac{p}{p+1}} \frac{2^p(p+1)^{p+1}}{\mu_0^{1/(p+1)}} + \big(\tfrac{(p+1)2^{p-1}}{\mu_0}\big)^{\frac{1}{p+1}} K\right)^{\frac{p+1}{p}} \\ &\overset{(3.3.59)}{\leq} \quad \left(\big(\tfrac{(p+1)2^{p-1}}{\mu_0}\big)^{\frac{1}{p+1}} (K^p + K)\right)^{\frac{p+1}{p}} \;\leq\; 4\big(\tfrac{p+1}{\mu_0}\big)^{\frac{1}{p}} K^p. \qquad \square \end{aligned}$$

Now, let us discuss the overall dependence of δ and K on p, given by the claim of Theorem 3.3.11. For simplicity, we fix $\frac{L_p(f)}{\varepsilon}$, $\beta_d(x_0;x^*)$, and μ_0. Thus, we observe the functions

$$\delta(p) \;\; := \;\; \frac{(p!)^{\frac{p}{p+1}}}{2^p(p+1)^{p+1}}, \qquad K(p) \;\; := \;\; 1 + 2^{\frac{1}{p}} \left(\frac{2^{p-1}(p+1)^{p+2}}{p!} \right)^{\frac{1}{p+1}}. \qquad (3.3.62)$$

One can see that $\log_2 \delta(p) \leq -p$. Therefore, increasing the order of the method by one, it requires at least to double the precision of solving the subproblem. At the same time, we have (using Stirling's formula):

$$\lim_{p \to +\infty} K(p) \;\; = \;\; 1 + 2\exp\Bigl(\lim_{p \to +\infty} \frac{(p+2)\log(p+1) - \log p!}{p+1} \Bigr) \;\; = \;\; 1 + 2\exp(1).$$

Hence, the value of $K(p)$ is bounded from above by an absolute constant. The graphs of the dependence (3.3.62) are shown in Figure 3.10. Note that in practice, we are interested rather in small values of p.

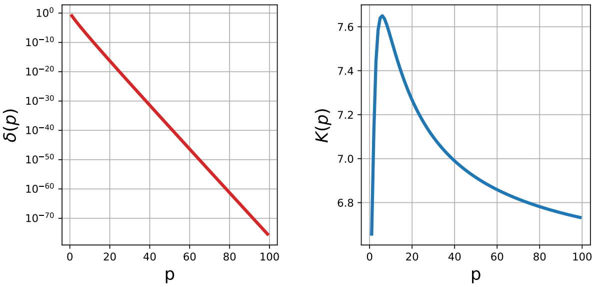


Figure 3.10: The dependence of δ and K on p, while $\frac{L_p(f)}{\varepsilon}$ and $\beta_d(x_0;x^*)$ are fixed.

Theorem 3.3.12 (Strongly Convex Case). *Let $\sigma_d(\psi) > 0$ and condition number ω be defined as in (3.3.56). Let for a given $\varepsilon > 0$, the inner accuracy δ be fixed as follows:*

$$\delta \;\; = \;\; \left(\frac{p!\,\varepsilon}{L_p(f)} \right)^{\frac{p}{p+1}} \frac{\mu_0 p\, \omega}{2^p(p+1)^{((p+1)^2+1)/(p+1)}}. \qquad (3.3.63)$$

Then, in order to achieve $F(x_K) - F^* \leq \varepsilon$, it is enough to perform

$$K \;\; = \;\; \left\lfloor 2 + \tfrac{1}{\omega}\mathcal{L} \right\rfloor \qquad (3.3.64)$$

iterations of algorithm (3.3.58)*, where*

$$\mathcal{L} \;\overset{\mathrm{def}}{=}\; \log\Big(\max\big\{ \tfrac{(p+1)^p}{\omega^{p+1}}, \tfrac{L_p(f)\beta_d(x_0;x^*)(p+1)^{p+1}2^{p+\frac{1}{p}}}{p!\,\varepsilon} \big\}\Big).$$

The total number of oracle calls N_K is bounded as follows:

$$\begin{array}{rcl} N_K & \leq & K \cdot \Big(3 + \big(1 + \tfrac{e}{(e-1)p}\big) \cdot \big(1 + \mathcal{L}\big) + \log\Big(\\ & & \max\{1, \Big(\tfrac{4\sigma_d(\psi)p!}{(p+1)L_p(f)}\Big)^{\frac{1}{p}}\} \cdot \Big(1 + \tfrac{1}{\mu_0}\Big) \cdot \tfrac{(p+1)^{\frac{p+2}{p}}}{p^{\frac{p+1}{p}}} \cdot 2^{\frac{2p^2+p+4}{p}} \Big)\Big). \end{array} \qquad (3.3.65)$$

Proof. At every iteration $k \geq 1$, we have $A_{k+1} = (1-\omega)^{-1}A_k \geq A_k \exp(\omega)$. At the same time, we know that

$$\omega \;\leq\; \tfrac{1}{2} \;\leq\; \tfrac{e-1}{e}, \qquad (3.3.66)$$

where $e = \exp(1)$. Since for all $\alpha \in [0,1]$ it holds

$$1 - \tfrac{e-1}{e}\alpha \;\;\geq\;\; \exp(-\alpha),$$

taking $\alpha = \omega\frac{e}{e-1} \overset{(3.3.66)}{\leq} 1$ we obtain $A_{k+1} \leq A_k \exp\big(\omega\frac{e}{e-1}\big)$. Therefore we have, for all $k \geq 0$:

$$A_1 \exp\big(k\omega\big) \;\;\leq\;\; A_{k+1} \;\leq\; A_1 \exp\Big(k\omega\tfrac{e}{e-1}\Big). \qquad (3.3.67)$$

Now, estimate (3.3.64) follows directly from (3.3.67) and (3.3.30) by using the value $A_1 = \frac{p!\,\mu_0}{2^{p-1}(p+1)^{p+1}L_p(f)}$.

By the choice of a_{k+1}, we have $\frac{\ell_{k+1}}{\bar{\mu}_{k+1}} \overset{(3.3.57)}{\leq} 1$, and we need only to estimate the value under the logarithm in (3.3.49). For every $0 \leq k \leq K$, we have:

$$\begin{aligned}
\frac{\ell_{k+1} D_{k+1}}{\delta^{\frac{p+1}{p}}} \quad & \overset{(3.3.57),(3.3.51)}{\leq} \quad \frac{\bar{\mu}_{k+1} R_k(p,\delta)\left(1+\frac{1}{\mu_0}\right)}{\delta^{\frac{p+1}{p}}} \\
& = \quad (\mu_0 + \sigma_d(\psi) A_{k+1})^{\frac{1}{p}} 2^{\frac{1}{p}-1}\left(1+\frac{1}{\mu_0}\right) \\
& \qquad \cdot \left(\frac{(\mu_0 \beta_d(x_0;x^*))^{\frac{p}{p+1}}}{\delta} + \left(\frac{(p+1)2^{p-1}}{\mu_0}\right)^{\frac{1}{p+1}} k \right)^{\frac{p+1}{p}} \\
& \leq \quad (\mu_0 + \sigma_d(\psi) A_{K+1})^{\frac{1}{p}} 2^{\frac{1}{p}-1}\left(1+\frac{1}{\mu_0}\right) \\
& \qquad \cdot \left(\frac{(\mu_0 \beta_d(x_0;x^*))^{\frac{p}{p+1}}}{\delta} + \left(\frac{(p+1)2^{p-1}}{\mu_0}\right)^{\frac{1}{p+1}} K \right)^{\frac{p+1}{p}} .
\end{aligned}$$

Let us estimate different terms in this expression separately.

1. By definition of ω, we have

$$\omega^{p+1} \quad \leq \quad \frac{(p+1)^p \sigma_d(\psi) A_1}{\mu_0}. \tag{3.3.68}$$

   Therefore,

$$\begin{aligned}
\mu_0 + \sigma_d(\psi) A_{K+1} \quad & \overset{(3.3.68),(3.3.67)}{\leq} \quad \sigma_d(\psi) A_1 \left(\frac{(p+1)^p}{\omega^{p+1}} + \exp\left(K\omega \frac{e}{e-1}\right)\right) \\
& \overset{(3.3.64)}{\leq} \quad 2\sigma_d(\psi) A_1 \exp\left(K\omega \frac{e}{e-1}\right).
\end{aligned}$$

2. Substituting the value for δ, we obtain

$$\begin{aligned}
\frac{(\mu_0 \beta_d(x_0;x^*))^{\frac{p}{p+1}}}{\delta} \quad & \overset{(3.3.63)}{=} \quad \left(\frac{L_p(f)\beta_d(x_0;x^*)}{p!\,\varepsilon}\right)^{\frac{p}{p+1}} \frac{2^p (p+1)^{((p+1)^2+1)/(p+1)}}{p\,\omega \mu_0^{\frac{1}{p+1}}} \\
& \overset{(3.3.64)}{\leq} \quad \frac{(p+1)^2 2^{(2p^2+p+1)/(p+1)}}{p\,\omega \mu_0^{\frac{1}{p+1}}} \exp\left(K\omega \frac{p}{p+1}\right).
\end{aligned}$$

3. Finally, using that $\exp(x) \geq x$ for all $x \geq 0$, we have

$$K \quad \leq \quad \frac{p+1}{p\,\omega} \exp\left(K\omega \frac{p}{p+1}\right).$$

Therefore,

$$
\begin{array}{rcl}
\frac{\ell_{k+1}D_{k+1}}{\delta^{\frac{p+1}{p}}} & \leq & \exp\Big(K\omega\frac{e}{(e-1)p}\Big)\cdot\Big(2^{2-p}\sigma_d(\psi)A_1\Big)^{\frac{1}{p}}\cdot\Big(1+\frac{1}{\mu_0}\Big)\\[2ex]
& & \cdot\left(\frac{\exp\big(K\omega\frac{p}{p+1}\big)}{p\,\omega\mu_0^{1/(p+1)}}\Big((p+1)^2 2^{\frac{2p^2+p+1}{p+1}}+(p+1)^{\frac{p+2}{p+1}}2^{\frac{p-1}{p+1}}\Big)\right)^{\frac{p+1}{p}}\\[2ex]
& < & \exp\Big(K\omega\Big(\frac{e}{(e-1)p}+1\Big)\Big)\cdot\Big(\frac{1}{p\,\omega}\Big)^{\frac{p+1}{p}}\cdot\Big(\frac{\sigma_d(\psi)A_1}{\mu_0}\Big)^{\frac{1}{p}}\\[2ex]
& & \cdot\Big(1+\frac{1}{\mu_0}\Big)\cdot(p+1)^{\frac{2(p+1)}{p}}2^{\frac{2p^2+p+4}{p}}\\[2ex]
& = & \exp\Big(K\omega\Big(\frac{e}{(e-1)p}+1\Big)\Big)\cdot\max\{1,\Big(\frac{4\sigma_d(\psi)p!}{(p+1)L_p(f)}\Big)^{\frac{1}{p}}\}\\[2ex]
& & \cdot\Big(1+\frac{1}{\mu_0}\Big)\cdot\frac{(p+1)^{\frac{p+2}{p}}}{p^{\frac{p+1}{p}}}\cdot 2^{\frac{2p^2+p+4}{p}},
\end{array}
$$

and we obtain (3.3.65). □

According to Theorem 3.3.11 and Theorem 3.3.12, the rate of convergence for the outer iterations of algorithm (3.3.58) is of the same order, than that one of accelerated Tensor Method from [118]. However, at each step it uses logarithmic number of steps of the basic method. It seems to be a reasonable price for the level of generality. Indeed, we are free to choose an arbitrary method as the basic one. The only requirement to it is the possibility of solving the inner subproblem (3.3.39) efficiently.

Note, that an additional feature of our methods is that the sequences of points $\{x_k\}_{k\geq 0}$ and $\{v_k\}_{k\geq 0}$ form *triangles* (see the rule (3.3.20)). A first-order accelerated method with this nice property was discovered in [55].

3.3.4 Experiments

Quadratic function. Let us compare numerical performance of the first-order Contracting Proximal Method and the classical Proximal-Point algorithm (3.3.1) for unconstrained minimization of a convex quadratic function:

$$
f(x) \quad = \quad \tfrac{1}{2}\langle Ax,x\rangle-\langle b,x\rangle, \qquad x\in\mathbb{R}^n,
$$

with $A = A^* \succeq 0$. We also run the Gradient Method and the Accelerated Gradient Method for this problem. A typical behaviour of the algorithms is shown on Figure 3.11. The Contracting Proximal Method has the same iteration rate as that of the Accelerated Gradient Method, but requires more gradient evaluations (matrix-vector products) per iteration.

To compute every step of the proximal algorithms, we use the Gradient Method with line search. We try different strategies for choosing inner accuracies δ_k, and end up with a simple rule $\delta_k = 1/k^2$, which provides a good balance in performance of outer proximal iterations and the inner method (usually, it requires to do about 4 inner steps per iteration).

We generate a random rotation from the uniform distribution, but the set of eigenvalues of the matrix was fixed according to the sigmoid function, for some given $\alpha > 0$

$$\lambda_i \quad = \quad \frac{1}{1+\exp\left(\frac{\alpha}{n-1}(n+1-2i)\right)}, \qquad 1 \leq i \leq n.$$

Therefore it holds: $\lambda_1 = 1/(1+\exp(\alpha))$ and $\lambda_n = 1/(1+\exp(-\alpha))$, so parameter α is related to the *condition number* of the problem.

In Table 3.1 we demonstrate the number of iterations and the total number of matrix-vector products, which are required for the methods to solve the problem up to $\varepsilon = 10^{-7}$ accuracy in functional residual.

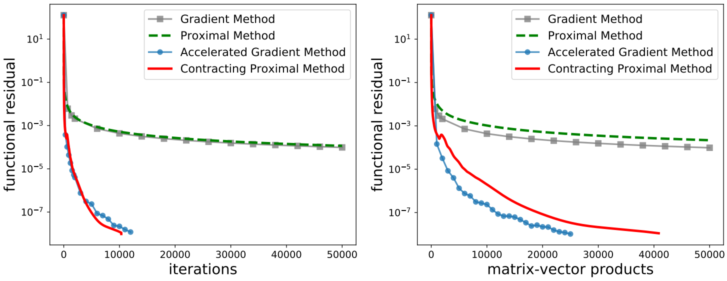


Figure 3.11: Convergence of first-order methods on quadratic function.

We see that the Contracting Proximal Method is *always better* than the usual Proximal algorithm. It requires about the same number of iteration as the Accelerated Gradient Methods, but it needs to spend more oracle calls per iteration, which confirms the theory.

		Gradient Method		**Proximal Method**		**Accelerated Gradient Method**		**Contracting Proximal Method**	
n	q	iter	mat-vec	iter	mat-vec	iter	mat-vec	iter	mat-vec
500	10^{-2}	339	339	361	1044	115	229	**74**	**137**
	10^{-4}	12158	12158	12842	36731	**350**	**699**	393	1104
	10^{-6}	96072	96072	99269	313795	**854**	**1707**	1081	3780
1000	10^{-2}	338	338	359	1035	110	219	**73**	**135**
	10^{-4}	11884	11884	11912	56996	**360**	**719**	361	1014
	10^{-6}	77675	77675	80758	239508	**755**	**1509**	1117	3957

Table 3.1: Minimization of quadratic function, $q = \lambda_{\min}(A)/\lambda_{\max}(A)$.

Log-Sum-Exp. In the next example we compare performance of second-order methods for unconstrained minimization of the following objective

$$f(x) \quad = \quad \mu \log\left(\sum_{i=1}^{m} \exp\left(\frac{\langle a_i, x\rangle - b_i}{\mu} \right) \right), \qquad x \in \mathbb{R}^n,$$

where $\mu > 0$ is a parameter, while coefficients of the vectors $\{a_i\}_{i=1}^m$ and b are randomly generated, and we set $m = 6n$. We get the more ill-conditioned problem for smaller values of parameter μ.

We compare the Cubic Newton Method (1.4.9) and its accelerated variant from [111] with *Contracting Proximal Cubic Newton* (algorithm (3.3.58) for $p = 2$), when minimizing the objective up to $\varepsilon = 10^{-8}$ accuracy in functional residual. In these algorithms we use the following Euclidean norm for the primal space: $\|x\| = \langle Bx, x\rangle^{1/2}$, with matrix $B = \sum_{i=1}^m a_i a_i^T$, and fix regularization parameter being equal 1. The results are shown in Table 3.2.

		Cubic Newton		**Accelerated Cubic Newton**		**Contracting Proximal Cubic Newton**	
n	μ	iter	oracle	iter	oracle	iter	oracle
50	1	389	389	177	**353**	**112**	491
	0.1	482	482	202	**403**	**141**	587
	0.05	886	886	343	**685**	**236**	1129
100	1	834	834	308	**615**	**189**	849
	0.1	1210	1210	377	**753**	**232**	1021
	0.05	2598	2598	641	**1281**	**397**	1740

Table 3.2: Comparison of second-order methods on Log-sum-exp.

We see that the Contracting Proximal Method outperforms the direct methods in the number of iterations, but usually requires additional oracle calls for solving the subproblem.

3.3.5 Discussion

We have proposed a general acceleration scheme, based on the Proximal iterations. There are two distinguishing features of our methods: employing the *contraction* of the smooth component of the objective (this provides the acceleration), and flexibility of *prox-function* (its choice should take into account both the geometry of the problem and the order of the smoothness).

One of the recent important applications of the accelerated Proximal-Point methods in machine learning is the universal framework *Catalyst*, applicable to the first-order methods [93, 94]. This is a powerful approach for accelerating many specific optimization methods in a common way. We believe that our results can help in advancing in this direction, resulting in the faster high-order methods for many practical applications.

In Section 4.1.3 of Chapter 4, we will study inexact Contracting Proximal Methods based on the small residual in the function value. This condition can be preferable in situations when minimization of the (sub)gradient norm is difficult or even impossible to manage, such as stochastic and fully composite [43] optimization problems.

Chapter 4

Inexact and Stochastic Algorithms

With the growth of computing power, high-order optimization methods are becoming more and more popular in machine learning, due to their ability in tackling ill-conditioning and improving the rate of convergence. We have already seen several modifications of Newton's method equipped with global complexity guarantees, which are better than those of the first-order gradient methods.

The main weakness, though, is that every step of the high-order methods is much more expensive. It requires to solve an auxiliary subproblem, which involves a minimization of a sum of a nontrivial smooth function (at least, quadratic function as in Newton's method) with a regularizer, and possibly with some additional nondifferentiable components.

At the same time, it is clear that often we do not need exact solutions to the subproblems, especially in the beginning of the optimization process. In this chapter, we study relaxed versions of the high-order methods. Our aim is to ensure the fast convergence of the initial algorithms under some suitable and practically implementable conditions of inexactness for the method's step and for the oracle information.

4.1 Inexact Tensor Methods with Dynamic Inner Accuracies

Now we have a family of the basic Tensor Methods (1.5.1) (starting from the methods of order one), for each iteration of which we may need to call some auxiliary subsolver. We suggest to describe the approximate solution to the subproblem in terms of the residual in the function value. We propose two strategies for the inner accuracies, which are *dynamic* (changing with iterations). Indeed, there is no need to have a very precise solution to the subproblem at the first iterations, but we reasonably ask for a higher precision closer to the end of the optimization process.

Global convergence of the first-order methods with inexact proximal-gradient steps was studied in [142]. The authors considered the errors in the residual in function value of the subproblem, and require them to decrease with iterations at an appropriate rate. This setting is the most similar to our approach.

In [21, 22], adaptive second-order methods with cubic regularization and inexact steps were proposed. High-order inexact tensor methods were considered in [13, 74, 63, 62, 26, 96]. In all of these works, the authors describe approximate solution of the subproblem in terms of the corresponding first-order optimality condition (using the gradients). This can be difficult to achieve by the current optimization schemes, since more often we have a better (or the only) guarantees for the decrease of the residual in function value. The latter one is used as a measure of inaccuracy in the recent work [119] on the inexact Basic Tensor Methods. However, only the constant choice of the accuracy level is considered there.

We propose new dynamic strategies for choosing the inner accuracy for the general Tensor Methods, and several inexact algorithms based on it, with proven complexity guarantees, summarized next. We denote by δ_k the required precision for the residual in function value of the auxiliary problem.

- The rule $\delta_k := 1/k^{p+1}$, where $p \geq 1$ is the order of the method, and k is the iteration counter.

  Using this strategy, we propose two optimization schemes: Monotone Inexact Tensor Method I (algorithm (4.1.4)) and Inexact Tensor Method with Averaging (algorithm (4.1.31)). Both of them have the global complexity estimates $\mathcal{O}(1/\varepsilon^{\frac{1}{p}})$ iterations for minimizing the convex function up to ε -accuracy (see Theorem 4.1.3 and Theorem 4.1.9). The latter method seems to be the first *primal* high-order scheme (aggregating the points from the primal space only), having the explicit distance between the starting point and the solution, in the complexity bound.

- The rule $\delta_k := c \cdot (F(x_{k-2}) - F(x_{k-1}))$, where $F(x_i)$ are the values of the target objective during the iterations, and $c \geq 0$ is a constant.

  We incorporate this strategy into our Monotone Inexact Tensor Method II (algorithm (4.1.14)). For this scheme, for minimizing convex functions up to ε-accuracy by the methods of order $p \geq 1$, we prove the global complexity proportional to $\mathcal{O}(1/\varepsilon^{\frac{1}{p}})$ (Theorem 4.1.5). The global rate becomes linear, if the objective is uniformly convex (Theorem 4.1.6).

  Assuming that $\delta_k := c \cdot (F(x_{k-2}) - F(x_{k-1}))^{\frac{p+1}{2}}$, for the methods of order $p \geq 2$ as applied to minimization of strongly convex objective, we also establish the local superlinear rate of convergence (see Theorem 4.1.7).

- Using the technique of Contracting Proximal iterations discovered in the previous chapter, we propose inexact Accelerated Scheme (algorithm (4.1.35)), in which at each iteration k, we solve the corresponding subproblem with the precision $\zeta_k := 1/k^{p+2}$ in the residual of the function value, by inexact Tensor Methods of order $p \geq 1$. The resulting complexity bound is $\tilde{\mathcal{O}}(1/\varepsilon^{\frac{1}{p+1}})$ inexact tensor steps for minimizing the convex function up to ε accuracy (Theorem 4.1.10).

- Numerical results with empirical study of the methods for different accuracy policies are provided.

In Section 4.1.1 we introduce an approximate minimum of the high-order model of the objective. We study monotone inexact methods, for which we guarantee the decrease of the objective function at every iteration. In Section 4.1.2 we study the methods with averaging. In Section 4.1.3 we present our accelerated scheme. Section 4.1.4 contains numerical results.

4.1.1 Monotone Inexact Methods

As before, we are interested in solving the convex optimization problem in the composite form:

$$\min_x \Big\{ F(x) \quad = \quad f(x) + \psi(x) \Big\}.$$

For a fixed $p \geq 1$, we assume $f \in C^{p,p}(\operatorname{dom}\psi)$. Let us denote by $M_H(x;y)$ the following regularized model of our objective, with H being the regularization constant:

$$M_H(x;y) \quad \overset{\text{def}}{=} \quad \Omega_p(f,x;y) + \tfrac{H\|y-x\|^{p+1}}{(p+1)!} + \psi(y). \tag{4.1.1}$$

This model is used in the basic Tensor Method (1.5.1). For $H \geq pL_p$, function $M_H(x;\cdot)$ is *always convex* (Theorem 1.5.2), and thus its minimum is well defined.

Let us assume that at every step of our method, we minimize the model (4.1.1) inexactly by an auxiliary subroutine up to some given accuracy $\delta \geq 0$. We use the following definition of *inexact* δ*-step.*

Definition 4.1.1. Denote by $T_{H,\delta}(x)$ a point $T = T_{H,\delta}(x) \in \operatorname{dom}\psi$, satisfying

$$M_H(x;T) - \min_y M_H(x;y) \quad \leq \quad \delta. \tag{4.1.2}$$

The main property of this point is given by the next lemma.

Lemma 4.1.2. *Let $H = \alpha L_p$ for some $\alpha \geq p$. Then, for every $y \in \operatorname{dom}\psi$*

$$F(T_{H,\delta}(x)) \quad \leq \quad F(y) + \tfrac{(\alpha+1)L_p\|y-x\|^{p+1}}{(p+1)!} + \delta. \tag{4.1.3}$$

Proof. Indeed, denoting $T = T_{H,\delta}(x)$, we have

$$\begin{aligned} F(T) \quad &\overset{(1.3.5)}{\leq} \quad M_H(x;T) \quad \overset{(4.1.2)}{\leq} \quad M_H(x;y) + \delta, \\ &\overset{(1.3.5)}{\leq} \quad F(y) + \tfrac{(\alpha+1)L_p\|y-x\|^{p+1}}{(p+1)!} + \delta. \end{aligned}$$

□

Now, if we plug $y = x$ (a current iterate) into (4.1.3), we obtain

$$F(T_{H,\delta}(x)) \quad \leq \quad F(x) + \delta.$$

So in the case $\delta = 0$ (exact tensor step), we would have nonincreasing sequence $\{F(x_k)\}_{k\geq 0}$ of test points of the method. However, this is not the case for $\delta > 0$ (inexact tensor step). Therefore we propose the following minimization scheme *with correction.*

Monotone Inexact Tensor Method, I

```
Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$. Fix $H = pL_p$.
Iteration $k \geq 0$.
1: Pick up $\delta_{k+1} \geq 0$.
2: Compute inexact tensor step $T_{k+1} = T_{H,\delta_{k+1}}(x_k)$.
3: If $F(T_{k+1}) \leq F(x_k)$, then set $x_{k+1} = T_{k+1}$.
   Else choose $x_{k+1} = x_k$.
```

(4.1.4)

If at some step $k \geq 0$ of this algorithm we get $x_{k+1} = x_k$, then we need to decrease inner accuracy for the next step. From the practical point of view, an efficient implementation of this algorithm should include a possibility of improving accuracy of the previously computed point.

Denote by D_0 the radius of the initial level set of the objective:

$$D_0 \overset{\mathrm{def}}{=} \sup_x \Big\{ \|x - x^*\| \ : \ F(x) \leq F(x_0) \Big\}. \tag{4.1.5}$$

For algorithm (4.1.4), we can prove the following convergence result, which uses a simple strategy for choosing δ_{k+1}.

Theorem 4.1.3. *Let $D_0 < +\infty$. Let the sequence of inner accuracies $\{\delta_k\}_{k \geq 1}$ be chosen according to the rule*

$$\boxed{\delta_k \ = \ \tfrac{c}{k^{p+1}}} \tag{4.1.6}$$

with some $c \geq 0$. Then for the sequence $\{x_k\}_{k \geq 1}$ produced by algorithm (4.1.4), we have

$$F(x_k) - F^* \ \leq \ \tfrac{(p+1)^{p+1} L_p D_0^{p+1}}{p!\, k^p} + \tfrac{c}{k^p}. \tag{4.1.7}$$

Proof. Indeed, by Lemma 4.1.2, for every $y \in \mathrm{dom}\,\psi$ and $k \geq 0$ we have

$$F(x_{k+1}) \ \leq \ F(T_{k+1}) \ \overset{(4.1.3)}{\leq} \ F(y) + \tfrac{L_p \|y - x_k\|^{p+1}}{p!} + \delta_{k+1}. \tag{4.1.8}$$

Let us introduce an *arbitrary* sequence of positive increasing coefficients $\{A_k\}_{k \geq 0}$, $A_0 \overset{\text{def}}{=} 0$. Denote $a_{k+1} \overset{\text{def}}{=} A_{k+1} - A_k$. Then, plugging $y = \frac{a_{k+1}x^* + A_k x_k}{A_{k+1}}$ into (4.1.8), we obtain

$$F(x_{k+1}) \quad \leq \quad \frac{a_{k+1}}{A_{k+1}} F^* + \frac{A_k}{A_{k+1}} F(x_k) + \frac{a_{k+1}^{p+1}}{A_{k+1}^{p+1}} \frac{L_p \|x_k - x^*\|^{p+1}}{p!} + \delta_{k+1},$$

or, equivalently

$$\begin{aligned} A_{k+1}(F(x_{k+1}) - F^*) \quad &\leq \quad A_k(F(x_k) - F^*) + \frac{a_{k+1}^{p+1}}{A_{k+1}^{p}} \frac{L_p \|x_k - x^*\|^{p+1}}{p!} \\ &\qquad + A_{k+1}\delta_{k+1}. \end{aligned}$$

Summing up these inequalities, we get, for every $k \geq 1$

$$\begin{aligned} A_k(F(x_k) - F^*) \quad &\leq \quad \sum_{i=1}^{k} A_i \delta_i + \frac{L_p}{p!} \sum_{i=1}^{k} \frac{a_i^{p+1}}{A_i^p} \|x_i - x^*\|^{p+1} \\ &\leq \quad \sum_{i=1}^{k} A_i \delta_i + \frac{L_p D_0^{p+1}}{p!} \sum_{i=1}^{k} \frac{a_i^{p+1}}{A_i^p}, \end{aligned} \tag{4.1.9}$$

where the last inequality holds due to monotonicity of the method. Finally, let us fix $A_k \equiv k^{p+1}$. Then, for some $\xi \in [k-1; k]$,

$$a_k \quad = \quad k^{p+1} - (k-1)^{p+1} \quad = \quad (p+1)\xi^p \quad \leq \quad (p+1)k^p.$$

Therefore,

$$\sum_{i=1}^{k} \frac{a_i^{p+1}}{A_i^p} \quad \leq \quad \sum_{i=1}^{k} \frac{(p+1)^{p+1} i^{p(p+1)}}{i^{(p+1)p}} \quad = \quad (p+1)^{p+1} k, \tag{4.1.10}$$

and

$$\sum_{i=1}^{k} A_i \delta_i \quad = \quad \sum_{i=1}^{k} \frac{c i^{p+1}}{i^{p+1}} \quad = \quad ck. \tag{4.1.11}$$

Plugging these bounds into (4.1.9) completes the proof. □

We see that the global convergence rate of the inexact Tensor Method remains on the same level, as of the exact one. Namely, in order to achieve $F(x_K) - F^* \leq \varepsilon$, we need to perform $K = \mathcal{O}(1/\varepsilon^{\frac{1}{p}})$ iterations of the algorithm. According to these bounds, at the last iteration K, the rule (4.1.6) requires to solve the subproblem up to the absolute accuracy $\delta_K = \mathcal{O}(c\varepsilon^{\frac{p+1}{p}})$.

This is intriguing, since for bigger p (order of the method) we need less accurate solutions. Note that this estimate for δ_K coincides with the constant choice of inner accuracy in [119]. However, the dynamic strategy (4.1.6) provides a significant decrease of the computational time on the first iterations of the method, which is also confirmed by our numerical results (see Section 4.1.4).

Now, looking at algorithm (4.1.4), one may think that we are forgetting the points T_{k+1} such that $F(T_{k+1}) \geq F(x_k)$, and thus we are loosing some computations. However, this is not true: even if point T_{k+1} has not been taken as x_{k+1}, we shall use it internally as a starting point for computing the next T_{k+2}. To support this concept, we introduce the *inexact δ-step* with an additional condition of *monotonicity.* Specifically,

Definition 4.1.4. Denote by $S_{H,\delta}(x)$ a point $S = S_{H,\delta}(x) \in \operatorname{dom}\psi$, satisfying the following two conditions:

$$M_H(x;S) - \min_y M_H(x;y) \;\leq\; \delta, \tag{4.1.12}$$

$$F(S) \;<\; F(x). \tag{4.1.13}$$

It is clear, that point S from Definition 4.1.4 satisfies Definition 4.1.1 as well (while the opposite is not always the case). Therefore, we can also use Lemma 4.1.2 for the *monotone* inexact tensor step.

Using this definition, we simplify algorithm (4.1.4) and present the following scheme.

Monotone Inexact Tensor Method, II

```
Initialization. Choose $x_0 \in \operatorname{dom}\psi$. Fix $H = pL_p$.
Iteration $k \geq 0$.
1: Pick up $\delta_{k+1} \geq 0$.
2: Compute inexact monotone tensor step
   $x_{k+1} = S_{H,\delta_{k+1}}(x_k)$.
```

(4.1.14)

When our method is strictly monotone, we guarantee that $F(x_{k+1}) < F(x_k)$ for all $k \geq 0$, and we propose to use the following *adaptive strategy* to define the inner accuracies.

Theorem 4.1.5. *Let $D_0 < +\infty$. Let sequence of inner accuracies $\{\delta_k\}_{k \geq 1}$ be chosen in accordance to the rule*

$$\boxed{\delta_{k+1} \;\; = \;\; c \cdot \bigl(F(x_{k-1}) - F(x_k)\bigr), \quad k \geq 1} \tag{4.1.15}$$

for some fixed $0 \leq c < \frac{1}{(p+2)3^{p+1}-1}$ and $\delta_1 \geq 0$. Then for the sequence $\{x_k\}_{k \geq 1}$ produced by algorithm (4.1.14), we have

$$F(x_k) - F^* \;\; \leq \;\; \tfrac{\gamma L_p D_0^{p+1}}{p!\, k^p} + \tfrac{\beta}{k^{p+2}}, \tag{4.1.16}$$

where γ and β are the constants:

$$\gamma \;\; \overset{\text{def}}{=} \;\; \tfrac{(p+2)^{p+1}}{1-c((p+2)3^{p+1}-1)}, \;\; \beta \;\overset{\text{def}}{=}\; \tfrac{\delta_1 + c2^{p+2}(F(x_0)-F^*)}{1-c((p+2)^2/(p+1)-1)}. \tag{4.1.17}$$

Proof. First, by the same reasoning as in Theorem 4.1.3, we obtain the following bound, for every $k \geq 1$:

$$A_k(F(x_k) - F^*) \;\; \leq \;\; \sum_{i=1}^{k} A_i \delta_i + \tfrac{L_p D_0^{p+1}}{p!} \sum_{i=1}^{k} \tfrac{a_i^{p+1}}{A_i^p}, \tag{4.1.18}$$

where $\{A_k\}_{k \geq 0}$ is a sequence of increasing coefficients, with $A_0 \overset{\text{def}}{=} 0$, and $a_k \overset{\text{def}}{=} A_k - A_{k-1}$. Substituting into (4.1.18) the expression for δ_i, we get

$$\begin{aligned} A_k(F(x_k) - F^*) \;\; &\leq \;\; A_1\delta_1 + c\sum_{i=2}^{k} A_i(F(x_{i-2}) - F(x_{i-1})) \\ &\qquad + \tfrac{L_p D_0^{p+1}}{p!} \sum_{i=1}^{k} \tfrac{a_i^{p+1}}{A_i^p}, \qquad k \geq 1, \end{aligned} \tag{4.1.19}$$

or, rearranging the terms, it holds for every $k \geq 2$:

$$\begin{aligned} &(c+1)A_k(F(x_k) - F^*) \\ &\quad \leq \quad A_k(F(x_k) - F^*) + cA_k(F(x_{k-1}) - F^*) \\ &\quad \overset{(4.1.19)}{\leq} \quad \tfrac{L_p D_0^{p+1}}{p!} \sum_{i=1}^{k} \tfrac{a_i^{p+1}}{A_i^p} + c\sum_{i=1}^{k-2} (A_{i+2} - A_{i+1})(F(x_i) - F^*) \\ &\qquad\quad + A_1\delta_1 + cA_2(F(x_0) - F^*) \\ &\quad = \quad \tfrac{L_p D_0^{p+1}}{p!} \sum_{i=1}^{k} \tfrac{a_i^{p+1}}{A_i^p} + c\sum_{i=1}^{k-2} a_{i+2}(F(x_i) - F^*) \\ &\qquad\quad + A_1\delta_1 + cA_2(F(x_0) - F^*), \end{aligned} \tag{4.1.20}$$

and for $k = 1$ we have

$$\begin{array}{rcl} A_1(F(x_1) - F^*) & \overset{(4.1.19)}{\leq} & A_1\delta_1 + \frac{L_p D_0^{p+1}}{p!} \frac{a_1^{p+1}}{A_1^p} \\ & = & A_1\big(\delta_1 + \frac{L_p D_0^{p+1}}{p!}\big). \end{array} \tag{4.1.21}$$

Now, let us pick $A_k \equiv k^{p+2}$. Then,

$$a_k \;\; \equiv \;\; k^{p+2} - (k-1)^{p+2} \;\; \leq \;\; (p+2)k^{p+1},$$

and

$$\sum_{i=1}^k \frac{a_i^{p+1}}{A_i^p} \;\; \leq \;\; (p+2)^{p+1} \sum_{i=1}^k \frac{i^{(p+1)^2}}{i^{(p+2)p}} \;\; = \;\; (p+2)^{p+1} \sum_{i=1}^k i \;\; \leq \;\; (p+2)^{p+1}k^2.$$

Therefore, (4.1.20) leads to

$$\begin{array}{l} (c+1)k^{p+2}(F(x_k) - F^*) \\ \leq \;\; \frac{(p+2)^{p+1}k^2 L_p D_0^{p+1}}{p!} + c(p+2)\sum\limits_{i=1}^{k-2}(i+2)^{p+1}(F(x_i) - F^*) \\ \quad + \delta_1 + c2^{p+2}(F(x_0) - F^*), \qquad k \geq 2. \end{array} \tag{4.1.22}$$

And the statement to be proved is

$$F(x_k) - F^* \;\; \leq \;\; \frac{\beta}{k^{p+2}} + \frac{\gamma L_p D_0^{p+1}}{p!\, k^p}, \qquad k \geq 1, \tag{4.1.23}$$

where β and γ are from (4.1.17). Note that from our assumptions c is small enough: $c \leq \frac{1}{(p+2)3^{p+1}-1}$. Hence, the constants are correctly defined.

Let us prove (4.1.23) by induction. It holds for $k = 1$ by (4.1.21). Assuming that it holds for all $1 \leq k \leq K-2$, we have

$$\begin{array}{rl} F(x_K) - F^* & \overset{(4.1.22),(4.1.23)}{\leq} \;\; \frac{(p+2)^{p+1} L_p D_0^{p+1}}{(c+1)\, p!\, K^p} \\ & + \;\; \frac{c(p+2)}{(c+1)K^{p+2}} \sum\limits_{i=1}^{K-2} (i+2)^{p+1}\big(\frac{\gamma L_p D_0^{p+1}}{p!\, i^p} + \frac{\beta}{i^{p+2}}\big) + \frac{\delta_1 + c2^{p+2}(F(x_0) - F^*)}{(c+1)K^{p+2}} \\ = & \big(\frac{(p+2)^{p+1}}{c+1} + \frac{\gamma c(p+2)}{(c+1)K^2} \sum\limits_{i=1}^{K-2} \frac{(i+2)^{p+1}}{i^p}\big) \cdot \frac{L_p D_0^{p+1}}{p!\, K^p} \\ & + \;\; \big(\frac{\beta c(p+2)}{(c+1)} \sum\limits_{i=1}^{K-2} \frac{1}{i^{p+2}} + \frac{\delta_1 + c2^{p+2}(F(x_0) - F^*)}{c+1}\big) \cdot \frac{1}{K^{p+2}}. \end{array}$$

Using in the last expression the following two simple bounds:

$$\sum_{i=1}^{K-2} \frac{(i+2)^{p+1}}{i^p} \quad \leq \quad 3^{p+1} \sum_{i=1}^{K-2} \frac{i^{p+1}}{i^p} \;\leq\; 3^{p+1} K^2,$$

$$\sum_{i=1}^{K-2} \frac{1}{i^{p+2}} \quad \overset{(1.3.9)}{\leq} \quad \frac{p+2}{p+1},$$

we obtain

$$\begin{array}{rcl} F(x_K) - F^* & \leq & \frac{(p+2)^{p+1} + \gamma c (p+2) 3^{p+1}}{c+1} \cdot \frac{L_p D_0^{p+1}}{p!\, K^p} \\ & & \quad + \left(\frac{\beta c (p+2)^2}{(c+1)(p+1)} + \frac{\delta_1 + c 2^{p+2}(F(x_0) - F^*)}{c+1} \right) \frac{1}{K^{p+2}}, \end{array}$$

Therefore, to finish the proof, its enough to verify two equations:

$$\frac{\beta c (p+2)^2}{(c+1)(p+1)} + \frac{\delta_1 + c 2^{p+2}(F(x_0) - F^*)}{c+1} \;=\; \beta, \quad \text{and} \quad \frac{(p+2)^{p+1} + \gamma c (p+2) 3^{p+1}}{c+1} \;=\; \gamma.$$

which hold by definition (4.1.17). □

The rule (4.1.15) is surprisingly simple and natural: while the method is approaching the optimum, it becomes more and more difficult to optimize the function. Consequently, the progress in the function value at every step is decreasing. Therefore, we need to solve the auxiliary problem more accurately, and this is exactly what we are doing in accordance to this rule.

It is also notable, that the rule (4.1.15) is *universal*, in a sense that it remains the same (up to a constant factor) for the methods of *any order*, starting from $p = 1$.

This strategy also works for the nondegenerate case. Let us assume that our objective is *uniformly convex* of degree $p+1$ with constant σ_{p+1} (see Chapter 2). Thus, for all $x, y \in \mathrm{dom}\, \psi$ and $F'(x) \in \partial F(x)$ it holds

$$F(y) - F(x) + \langle F'(x), y - x \rangle \quad \geq \quad \frac{\sigma_{p+1}}{p+1} \|y - x\|^{p+1}. \tag{4.1.24}$$

For $p = 1$ this definition corresponds to the standard class of *strongly convex* functions.

Denote by $\bar{\omega}_p$ the *condition number* of degree p:

$$\bar{\omega}_p \quad \overset{\text{def}}{=} \quad \max\{ \frac{(p+1)^2 L_p}{p!\, \sigma_{p+1}}, 1 \}. \tag{4.1.25}$$

The next theorem shows, that $\bar{\omega}_p$ serves as the main factor in the complexity of solving the uniformly convex problems by inexact Tensor Methods.

Theorem 4.1.6. *Let $\sigma_{p+1} > 0$. Let sequence of inner accuracies $\{\delta_k\}_{k \geq 1}$ be chosen in accordance to the rule*

$$\boxed{\delta_k \quad = \quad c \cdot \bigl(F(x_{k-2}) - F(x_{k-1})\bigr), \quad k \geq 2} \tag{4.1.26}$$

for some fixed $0 \leq c < \frac{p}{p+1}\bar{\omega}_p^{-1/p}$ and $\delta_1 \geq 0$. Then for the sequence $\{x_k\}_{k \geq 1}$ produced by algorithm (4.1.14), we have the following <u>*linear*</u> *rate of convergence:*

$$F(x_{k+1}) - F^* \quad \leq \quad \bigl(1 - \tfrac{p}{p+1}\bar{\omega}_p^{-1/p} + c\bigr)(F(x_{k-1}) - F^*). \tag{4.1.27}$$

Proof. Let us substitute $x := x_k$ and $y := \lambda x^* + (1 - \lambda)x_k$ into (4.1.3), where $\lambda \equiv \bar{\omega}_p^{-1/p} \in (0, 1]$. This gives

$$\begin{aligned} F(x_{k+1}) \quad &\leq \quad \lambda F^* + (1-\lambda)F(x_k) + \tfrac{\lambda^{p+1} L_p \|x_k - x^*\|^{p+1}}{p!} + \delta_{k+1} \\ &\leq \quad \lambda F^* + (1-\lambda)F(x_k) + \tfrac{\lambda^{p+1}(p+1)L_p}{\sigma_{p+1} p!}(F(x_k) - F^*) + \delta_{k+1}, \end{aligned}$$

where we used uniform convexity. Therefore, for every $k \geq 1$:

$$\begin{aligned} F(x_{k+1}) - F^* \quad &\leq \quad \bigl(1 - \omega_p^{-1/p} + \tfrac{\bar{\omega}_p^{-1/p}}{p+1}\bigr)(F(x_k) - F^*) + \delta_{k+1} \\ &= \quad \bigl(1 - \tfrac{p}{p+1}\bar{\omega}_p^{-1/p}\bigr)(F(x_k) - F^*) + c(F(x_{k-1}) - F(x_k)) \\ &\leq \quad \bigl(1 - \tfrac{p}{p+1}\bar{\omega}_p^{-1/p} + c\bigr)(F(x_{k-1}) - F^*), \end{aligned}$$

the last inequality uses monotonicity of the method: $F(x_k) \leq F(x_{k-1})$ and the bound: $F^* \leq F(x_k)$. □

Let us pick $c = \frac{p}{2(p+1)}\bar{\omega}_p^{-1/p}$. Then, according to (4.1.27), in order solve the problem up to ε-accuracy: $F(x_K) - F^* \leq \varepsilon$, we need to perform

$$K \quad = \quad \mathcal{O}\Bigl(\bar{\omega}_p^{1/p} \log \tfrac{F(x_0) - F^*}{\varepsilon}\Bigr) \tag{4.1.28}$$

iterations of the algorithm.

Finally, we study the local behaviour of the method for strongly convex objective.

Theorem 4.1.7. *Let $\sigma_2 > 0$. Let sequence of inner accuracies $\{\delta_k\}_{k\geq 1}$ be chosen in accordance to the rule*

$$\boxed{\delta_k \quad = \quad c \cdot \bigl(F(x_{k-2}) - F(x_{k-1})\bigr)^{\frac{p+1}{2}}, \; k \geq 2} \tag{4.1.29}$$

with some fixed $c \geq 0$ and $\delta_1 \geq 0$. Then for $p \geq 2$ the sequence $\{x_k\}_{k\geq 1}$ produced by algorithm (4.1.14) has the <u>*local superlinear rate of convergence:*</u>

$$F(x_{k+1}) - F^* \quad \leq \quad \Bigl(\tfrac{L_p}{p!}\bigl(\tfrac{2}{\sigma_2}\bigr)^{\frac{p+1}{2}} + c\Bigr)(F(x_{k-1}) - F^*)^{\frac{p+1}{2}}. \tag{4.1.30}$$

Proof. Let us plug $y = x^*$ into (4.1.3). Thus, we obtain, for every $k \geq 1$:

$$\begin{aligned} F(x_{k+1}) \quad & \leq \quad F^* + \tfrac{L_p\|x_k - x^*\|^{p+1}}{p!} + \delta_{k+1} \\ & \leq \quad F^* + \tfrac{L_p}{p!}\bigl(\tfrac{2}{\sigma_2}\bigr)^{\frac{p+1}{2}}(F(x_k) - F^*)^{\frac{p+1}{2}} + \delta_{k+1} \\ & = \quad F^* + \tfrac{L_p}{p!}\bigl(\tfrac{2}{\sigma_2}\bigr)^{\frac{p+1}{2}}(F(x_k) - F^*)^{\frac{p+1}{2}} + c(F(x_{k-1}) - F(x_k))^{\frac{p+1}{2}} \\ & \leq \quad F^* + \Bigl(\tfrac{L_p}{p!}\bigl(\tfrac{2}{\sigma_2}\bigr)^{\frac{p+1}{2}} + c\Bigr)(F(x_{k-1}) - F^*)^{\frac{p+1}{2}}, \end{aligned}$$

where monotonicity of the method: $F(x_k) \leq F(x_{k-1})$ and the bound: $F^* \leq F(x_k)$ are used in the last inequality. □

Let us assume for simplicity, that the constant c is chosen to be small enough: $c \leq \frac{L_p}{p!}\bigl(\frac{2}{\sigma_2}\bigr)^{(p+1)/2}$. Then, we are able to describe the region of superlinear convergence as

$$\mathcal{Q} = \Biggl\{ x \in \operatorname{dom}\psi : F(x) - F^* \leq \Bigl(\tfrac{\sigma_2^{p+1}}{2^{p+3}}\bigl(\tfrac{p!}{L_p}\bigr)^2\Bigr)^{\frac{1}{(p-1)}} \Biggr\}.$$

After reaching it, the method becomes very fast: we need to perform no more than $O(\log\log\frac{1}{\varepsilon})$ additional iterations to solve the problem.

Note, that estimate (4.1.30) of the local convergence is slightly weaker than the corresponding one for *exact* Tensor Methods (see Section 2.2). For example, for $p = 2$ (Cubic regularization of Newton Method) we obtain the convergence of order $\frac{3}{2}$, not the quadratic, which affects only a constant factor in the complexity estimate. The region $\mathcal{Q}$ of the superlinear convergence is remaining the same.

4.1.2 Inexact Methods with Averaging

Methods from the previous section were developed by forcing the monotonicity of the sequence of function values $\{F(x_k)\}_{k \geq 0}$ into the scheme. As a byproduct, we get the radius of the initial level set D_0 (see definition (4.1.5)) in the right-hand side of our complexity estimates (4.1.7) and (4.1.16). Note, that D_0 may be significantly bigger than the distance $\|x_0 - x^*\|$ from the initial point to the solution.

Example 4.1.8. Consider the following function, for $x \in \mathbb{R}^n$:

$$f(x) \quad = \quad |x^{(1)}|^{p+1} + \sum_{i=2}^{n} |x^{(i)} - 2x^{(i-1)}|^{p+1},$$

where $x^{(i)}$ indicates ith coordinate of x. Clearly, the minimum of f is at the origin: $x^* = (0, \ldots, 0)^T$. Let us take two points: $x_0 = (1, \ldots, 1)^T$ and x_1, such that $x_1^{(i)} = 2^i - 1$. It holds, $f(x_0) = f(x_1) = n$, so they belong to the same level set. However, we have (for the standard Euclidean norm): $\|x_0 - x^*\| = \sqrt{n}$, while $D_0 \geq \|x_1 - x^*\| \geq 2^{n-1}$. □

Here we present an alternative approach, Tensor Methods with *Averaging*. In this scheme, we perform a step not from the previous point x_k, but from a point y_k, which is a convex combination of the previous point and the starting point:

$$y_k \quad = \quad \lambda_k x_k + (1 - \lambda_k) x_0,$$

where $\lambda_k \equiv \left(\frac{k}{k+1}\right)^{p+1}$. The whole optimization scheme remains very simple.

```
Inexact Tensor Method with Averaging

Initialization. Choose $x_0 \in \mathrm{dom}\, \psi$. Fix $H = pL_p$.
Iteration $k \geq 0$.
1: Set $\lambda_k = \left(\frac{k}{k+1}\right)^{p+1}$, $y_k = \lambda_k x_k + (1 - \lambda_k) x_0$.
2: Pick up $\delta_{k+1} \geq 0$.
3: Compute inexact tensor step $x_{k+1} := T_{H, \delta_{k+1}}(y_k)$.
```

(4.1.31)

For this method, we are able to prove a similar convergence result as that of algorithm (4.1.4). However, now we have the explicit distance $\|x_0 - x^*\|$ in the right hand side of our bound for the convergence rate (compare with Theorem 4.1.3).

Theorem 4.1.9. *Let sequence of inner accuracies $\{\delta_k\}_{k\geq 1}$ be chosen according the rule*

$$\boxed{\delta_k \quad = \quad \tfrac{c}{k^{p+1}}} \tag{4.1.32}$$

for some $c \geq 0$. Then for the sequence $\{x_k\}_{k\geq 1}$ produced by algorithm (4.1.31), we have

$$F(x_k) - F^* \quad \leq \quad \tfrac{(p+1)^{p+1} L_p \|x_0 - x^*\|^{p+1}}{p!\, k^p} + \tfrac{c}{k^p}. \tag{4.1.33}$$

Proof. The proof is similar to that one of Theorem 4.1.3. By Lemma 4.1.2, for every $y \in \mathrm{dom}\, \psi$, we have

$$F(x_{k+1}) \quad \overset{(4.1.3)}{\leq} \quad F(y) + \tfrac{L_p \|y - y_k\|^{p+1}}{p!} + \delta_{k+1}, \qquad k \geq 0.$$

Let us substitute $y = \lambda_k x_k + (1 - \lambda_k)x^*$, with λ_k defined in the algorithm:

$$\lambda_k \quad \equiv \quad \left(\tfrac{k}{k+1}\right)^{p+1}.$$

Thus we obtain

$$F(x_{k+1}) \quad \leq \quad (1 - \lambda_k)F^* + \lambda_k F(x_k) + (1 - \lambda_k)^{p+1} \tfrac{L_p \|x_0 - x^*\|^{p+1}}{p!} + \delta_{k+1},$$

or, equivalently

$$A_{k+1}(F(x_{k+1}) - F^*) \quad \leq \quad A_k(F(x_k) - F^*) + \tfrac{a_{k+1}^{p+1}}{A_{k+1}^p} \tfrac{L_p \|x_0 - x^*\|^{p+1}}{p!}$$

$$+ A_{k+1}\delta_{k+1},$$

where $A_k \equiv k^{p+1}$ and $a_k \equiv A_k - A_{k-1}$ (so it holds: $\lambda_k \equiv A_k / A_{k+1}$ and $1 - \lambda_k \equiv a_{k+1}/A_{k+1}$). Telescoping these inequalities and using the bounds (4.1.10) and (4.1.11) complete the proof. □

Thus, algorithm (4.1.31) seems to be the first *Primal* Tensor method (aggregating only the points from the primal space $\mathbb{E}$), which admits the explicit initial distance in the global convergence estimate (4.1.33). Table 4.1 contains a short overview of the inexact Tensor methods from this section.

Algorithm	**The rule for δ_k**	**Global rate**	**Local superlinear rate**
Tensor Method (algorithm (1.5.1))	0	$\mathcal{O}\big(\frac{L_p D_0^{p+1}}{k^p}\big)$	Yes
Monotone Inexact Tensor Method, I (algorithm (4.1.4))	$1/k^{p+1}$	$\mathcal{O}\big(\frac{L_p D_0^{p+1}}{k^p}\big)$	No
Monotone Inexact Tensor Method, II (algorithm (4.1.14))	$(F(x_{k-1}) - F(x_k))^{\alpha}$	$\mathcal{O}\big(\frac{L_p D_0^{p+1}}{k^p}\big)$, $\alpha = 1$	Yes, $\alpha = \frac{p+1}{2}$
Inexact Tensor Method with Averaging (algorithm (4.1.31))	$1/k^{p+1}$	$\mathcal{O}\big(\frac{L_p \Vert x_0 - x^* \Vert^{p+1}}{k^p}\big)$	No

Table 4.1: Comparison of the inexact basic Tensor methods.

4.1.3 Acceleration

After the Fast Gradient Method had been discovered in [107], there were made huge efforts to develop accelerated second-order [111, 101, 61] and high-order [6, 118, 54, 63, 145] optimization algorithms. Most of these schemes use the notion of Estimating Sequences (see [117]), that is based on accumulating the gradients. In our inexact methods we guarantee only the progress for the objective function. Thus, we study an alternative approach to accelerate our inexact tensor methods, using the technique of Contracting Proximal iterations developed in Section 3.3.

In the accelerated scheme, two sequences of points are used: the main sequence $\{x_k\}_{k\geq 0}$, for which we are able to guarantee the convergence in function residuals, and auxiliary sequence $\{v_k\}_{k\geq 0}$ of prox-centers, starting from the same initial point: $v_0 = x_0$. Also, we use the sequence $\{A_k\}_{k\geq 0}$ of scaling coefficients. Denote $a_k \overset{\text{def}}{=} A_k - A_{k-1}, k \geq 1$.

Then, at every iteration, we apply Monotone Inexact Tensor Method, II (algorithm (4.1.14)) to minimize the following *contracted* objective with regularization:

$$\begin{aligned} h_{k+1}(x) \quad &\overset{\text{def}}{=} \quad A_{k+1} f\big(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\big) + a_{k+1}\psi(x) \\ &\qquad + \beta_d(v_k; x). \end{aligned}$$

Here $\beta_d(v_k; x) \overset{\text{def}}{=} d(x) - d(v_k) - \langle \nabla d(v_k), x - v_k \rangle$ is *Bregman divergence* centered at v_k, for the following choice of prox-function:

$$d(x) \quad \equiv \quad \tfrac{1}{p+1}\|x - x_0\|^{p+1},$$

which is uniformly convex of degree $p+1$ (Lemma 2.1.4). Therefore, inexact Tensor Method achieves fast linear rate of convergence (Theorem 4.1.6). By an appropriate choice of scaling coefficients $\{A_k\}_{k\geq 1}$, we are able to make the condition number of the subproblem being an absolute constant. This means that only $\tilde{\mathcal{O}}(1)$ steps of algorithm (4.1.14) are needed to find an approximate minimizer of $h_{k+1}(\cdot)$:

$$h_{k+1}(v_{k+1}) - h^*_{k+1} \quad \leq \quad \zeta_{k+1}. \tag{4.1.34}$$

Note that inexact condition (4.1.34) differs from the that one from Section 3.3, where a bound for the (sub)gradient norm was used. The bound for the residual in function value is easier to ensure by our methods. The price that we pay is a more difficult analysis. Also, it can be not easy to choose a stopping condition for the inner method.

```
Accelerated Scheme

Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$. Set $v_0 = x_0$, $A_0 = 0$.
Iteration $k \geq 0$.
1: Set $A_{k+1} = \frac{(k+1)^{p+1}}{L_p}$.
2: Pick up $\zeta_{k+1} \geq 0$.
3: Find $v_{k+1}$ such that (4.1.34) holds.
4: Set $x_{k+1} = \frac{a_{k+1}v_{k+1} + A_k x_k}{A_{k+1}}$.
```

(4.1.35)

Therefore, for accelerating inexact Tensor Methods, we propose a multilevel approach. On the upper level, we run algorithm (4.1.35). At each iteration of this method, we call algorithm (4.1.14) to find v_{k+1}.

Theorem 4.1.10. *Let sequence $\{\zeta_k\}_{k\geq 1}$ be chosen according to the rule*

$$\boxed{\zeta_k \quad = \quad \tfrac{c}{k^{p+2}}} \tag{4.1.36}$$

with some $c \geq 0$. Then for the iterations $\{x_k\}_{k \geq 1}$ produced by algorithm (4.1.35), it holds:

$$F(x_k) - F^* \quad \leq \quad \mathcal{O}\Bigl(\tfrac{L_p(\|x_0 - x^*\|^{p+1} + c)}{k^{p+1}} \Bigr). \tag{4.1.37}$$

For every $k \geq 0$, in order to find v_{k+1} by algorithm (4.1.14) (for minimizing $h_{k+1}(\cdot)$, starting from v_k), it is enough to perform no more than

$$\mathcal{O}\Bigl(\log \tfrac{(k+1)(\|x_0 - x^*\|^{p+1} + c)}{c} \Bigr). \tag{4.1.38}$$

inexact monotone tensor steps.

Proof. The proof is similar to that one of Theorem 3.3.4, where convergence rate of the Contracting Proximal Method is established. Additional technical difficulties, which are arising here, are caused by using inexact solution of the subproblem, equipped with the stopping condition (4.1.34).

We denote the optimal point of $h_{k+1}(\cdot)$ by $z_{k+1} \overset{\mathrm{def}}{=} \operatorname{argmin}_y h_{k+1}(y)$. Since the next prox-center v_{k+1} is defined as an approximate minimizer, we have

$$h_{k+1}(v_{k+1}) - h_{k+1}(z_{k+1}) \quad \leq \quad \zeta_{k+1}. \tag{4.1.39}$$

Function $h_{k+1}(\cdot)$ is strongly convex with respect to $d(\cdot)$, thus we have

$$\begin{array}{rcl} \zeta_{k+1} & \overset{(4.1.39)}{\geq} & h_{k+1}(v_{k+1}) - h_{k+1}(z_{k+1}) \;\; \geq \;\; \beta_d(z_{k+1}; v_{k+1}) \\ & \geq & \frac{1}{2^{p-1}(p+1)} \|v_{k+1} - z_{k+1}\|^{p+1}. \end{array} \tag{4.1.40}$$

Therefore,

$$\|v_{k+1} - z_{k+1}\| \quad \overset{(4.1.40)}{\leq} \quad \xi_{k+1} \quad \overset{\mathrm{def}}{=} \quad 2^{\frac{p-1}{p+1}} (p+1)^{\frac{1}{p+1}} \zeta_{k+1}^{\frac{1}{p+1}}. \tag{4.1.41}$$

Let us prove by induction the following inequality, for every $k \geq 0$ and all $x \in \operatorname{dom} \psi$:

$$\beta_d(x_0; x) + A_k F(x) \quad \geq \quad \beta_d(v_k; x) + A_k F(x_k) + C_k(x), \tag{4.1.42}$$

where $C_k(x) \overset{\mathrm{def}}{=} -\sum\limits_{i=1}^{k} \bigl(\tau_i \|x - v_i\| + \zeta_i \bigr)$, and $\tau_i \overset{\mathrm{def}}{=} p 2^{p-2} \|z_i - x_0\|^{p-1} \xi_i + 2^{p-2} \xi_i^p$.

It obviously holds for $k = 0$. Assume that it holds for the current iterate, and consider the next step $k+1$:

$$\begin{array}{rl} & \beta_d(x_0;x) + A_{k+1}F(x) \\ = & \beta_d(x_0;x) + A_kF(x) + a_{k+1}F(x) \\ \overset{(4.1.42)}{\geq} & \beta_d(v_k;x) + A_kF(x_k) + a_{k+1}F(x) + C_k(x) \\ \geq & \beta_d(v_k;x) + A_{k+1}f\big(\frac{a_{k+1}x+A_kx_k}{A_{k+1}}\big) \\ & + a_{k+1}\psi(x) + A_k\psi(x_k) + C_k(x) \\ = & h_{k+1}(x) + A_k\psi(x_k) + C_k(x), \end{array} \tag{4.1.43}$$

where the last inequality holds by convexity of f.

Using the strong convexity of $h_{k+1}(\cdot)$ with respect to $d(\cdot)$, we obtain

$$\begin{array}{rcl} h_{k+1}(x) & \geq & h_{k+1}(z_{k+1}) + \beta_d(z_{k+1};x) \\ & \overset{(4.1.39)}{\geq} & h_{k+1}(v_{k+1}) + \beta_d(z_{k+1};x) - \zeta_{k+1} \\ & = & h_{k+1}(v_{k+1}) + \beta_d(v_{k+1};x) + \beta_d(z_{k+1};v_{k+1}) \\ & & + \langle \nabla d(v_{k+1}) - \nabla d(z_{k+1}), x - v_{k+1}\rangle - \zeta_{k+1} \\ & \geq & h_{k+1}(v_{k+1}) + \beta_d(v_{k+1};x) - \zeta_{k+1} \\ & & - \|\nabla d(v_{k+1}) - \nabla d(z_{k+1})\|_* \cdot \|x - v_{k+1}\|. \end{array} \tag{4.1.44}$$

Now, computing the second derivative of $d(x) = \frac{1}{p+1}\|x-x_0\|^{p+1}$, we get

$$\begin{array}{rcl} \nabla^2 d(x) & = & (p-1)\|x-x_0\|^{p-3}B(x-x_0)(x-x_0)^*B \\ & & + \|x-x_0\|^{p-1}B \quad \preceq \quad p\|x-x_0\|^{p-1}B. \end{array} \tag{4.1.45}$$

Note that for every $a, b \geq 0$ and integer $p \geq 1$ it holds that

$$(a+b)^{p-1} \quad \leq \quad 2^{p-2}a^{p-1} + 2^{p-2}b^{p-1}. \tag{4.1.46}$$

For $p = 1$ this is trivial. For $p \geq 2$ it holds by convexity of the one-dimensional function $y(x) = x^{p-1}$, $x \geq 0$.

Therefore,

$$\begin{array}{ll} \|\nabla d(v_{k+1}) - \nabla d(z_{k+1})\|_* & \\ \quad = & \| \int\limits_0^1 \nabla^2 d(z_{k+1} + \tau(v_{k+1} - z_{k+1}))d\tau(v_{k+1} - z_{k+1})\|_* \\ \quad \overset{(4.1.41)}{\leq} & \xi_{k+1} \int\limits_0^1 \|\nabla^2 d(z_{k+1} + \tau(v_{k+1} - z_{k+1}))\| d\tau \\ \quad \overset{(4.1.45)}{\leq} & p\xi_{k+1} \int\limits_0^1 \|z_{k+1} - x_0 + \tau(v_{k+1} - z_{k+1})\|^{p-1} d\tau \\ \quad \overset{(4.1.46),(4.1.41)}{\leq} & p\xi_{k+1} \int\limits_0^1 \left(2^{p-2}\|z_{k+1} - x_0\|^{p-1} + 2^{p-2}\tau^{p-1}\xi_{k+1}^{p-1}\right) d\tau \\ \quad = & p2^{p-2}\|z_{k+1} - x_0\|^{p-1}\xi_{k+1} + 2^{p-2}\xi_{k+1}^p \overset{\text{def}}{=} \tau_{k+1}. \end{array} \tag{4.1.47}$$

Combining the obtained bounds together, we conclude

$$\begin{array}{rcl} \beta_d(x_0; x) + A_{k+1}F(x) & \overset{(4.1.43)}{\geq} & h_{k+1}(x) + A_k\psi(x_k) + C_k(x) \\ & \overset{(4.1.44),(4.1.47)}{\geq} & h_{k+1}(v_{k+1}) + \beta_d(v_{k+1}; x) - \tau_{k+1}\|x - v_{k+1}\| \\ & & \quad - \zeta_{k+1} + A_k\psi(x_k) + C_k(x) \\ & = & h_{k+1}(v_{k+1}) + \beta_d(v_{k+1}; x) + A_k\psi(x_k) + C_{k+1}(x) \\ & = & A_{k+1}f(x_{k+1}) + a_{k+1}\psi(v_{k+1}) + \beta_d(v_k; v_{k+1}) \\ & & \quad + \beta_d(v_{k+1}; x) + A_k\psi(x_k) + C_{k+1}(x) \\ & \geq & A_{k+1}F(x_{k+1}) + \beta_d(v_{k+1}; x) + C_{k+1}(x). \end{array}$$

Thus, (4.1.42) is proven. Let us plug $x := x^*$ into (4.1.42). We obtain

$$\begin{array}{ll} \beta_d(v_k; x^*) + A_k(F(x_k) - F^*) \overset{(4.1.42)}{\leq} & \beta_d(x_0; x^*) - C_k(x^*) \\ \quad = & \beta_d(x_0; x^*) + \sum\limits_{i=1}^k \zeta_i + \sum\limits_{i=1}^k \tau_i\|v_i - x^*\| \\ \quad \overset{(1.3.9)}{\leq} & \beta_d(x_0; x^*) + \frac{c(p+2)}{p+1} + \sum\limits_{i=1}^k \tau_i\|v_i - x^*\| \overset{\text{def}}{=} \alpha_k, \end{array} \tag{4.1.48}$$

and to finish the proof, we need to estimate α_k from above. By uniform convexity of $d(\cdot)$, we have

$$\tfrac{1}{2^{p-1}(p+1)}\|v_k - x^*\|^{p+1} \quad \leq \quad \beta_d(v_k; x^*) \quad \overset{(4.1.48)}{\leq} \quad \alpha_k. \tag{4.1.49}$$

At the same time,

$$\alpha_k \quad = \quad \alpha_{k-1} + \tau_k \|v_k - x^*\| \quad \overset{(4.1.49)}{\leq} \quad \alpha_{k-1} + 2^{\frac{p-1}{p+1}}(p+1)^{\frac{1}{p+1}}\tau_k \alpha_k^{\frac{1}{p+1}}.$$

By dividing both sides of the last inequality by $\alpha_k^{\frac{1}{p+1}}$, and by using monotonicity of $\{\alpha_k\}_{k\geq 0}$, we get

$$\alpha_k^{\frac{p}{p+1}} \quad \leq \quad \alpha_{k-1}^{\frac{p}{p+1}} + 2^{\frac{p-1}{p+1}}(p+1)^{\frac{1}{p+1}}\tau_k.$$

Therefore,

$$\alpha_k \quad \leq \quad \left(\alpha_0^{\frac{p}{p+1}} + 2^{\frac{p-1}{p+1}}(p+1)^{\frac{1}{p+1}} \sum_{i=1}^{k} \tau_i\right)^{\frac{p+1}{p}}. \tag{4.1.50}$$

To finish, it remains to bound the sum of τ_i, which is

$$\begin{array}{rcl}
\sum\limits_{i=1}^{k} \tau_i & = & \sum\limits_{i=1}^{k} (p 2^{p-2}\|z_i - x_0\|^{p-1}\xi_i + 2^{p-2}\xi_i^p) \\
& \overset{(4.1.46)}{\leq} & \sum\limits_{i=1}^{k} 4^{p-2}(p\|z_i - x^*\|^{p-1}\xi_i + p\|x_0 - x^*\|^{p-1}\xi_i + \frac{\xi_i^p}{2^{p-2}}) \\
& \overset{(4.1.41)}{=} & p4^{p-2}\sum\limits_{i=1}^{k}\|z_i - x^*\|^{p-1}\xi_i \\
& & \quad + \quad p(p+1)^{\frac{1}{p+1}}2^{\frac{p-1}{p+1}}4^{p-2}\|x_0 - x^*\|^{p-1}\sum\limits_{i=1}^{k}\zeta_i^{\frac{1}{p+1}} \\
& & \quad + \quad 2^{p-2}2^{\frac{(p-1)p}{p+1}}(p+1)^{\frac{p}{p+1}}\sum\limits_{i=1}^{k}\zeta_i^{\frac{p}{p+1}} \\
& \overset{(1.3.9)}{\leq} & p4^{p-2}\sum\limits_{i=1}^{k}\|z_i - x^*\|^{p-1}\xi_i + \Delta_1,
\end{array} \tag{4.1.51}$$

where

$$\begin{array}{rcl}
\Delta_1 & \overset{\text{def}}{=} & p(p+1)^{\frac{1}{p+1}}2^{\frac{p-1}{p+1}}4^{p-2}\|x_0 - x^*\|^{p-1}c^{\frac{1}{p+1}}(p+2) \\
& & \qquad + \; 2^{p-2}2^{\frac{(p-1)p}{p+1}}(p+1)^{\frac{p}{p+1}}c^{\frac{p}{p+1}}\frac{(p+2)p}{(p+2)p-p-1},
\end{array}$$

and we need to bound $\|z_i - x^*\|$ from above.

By substituting $x := x^*$ into (4.1.43), we have

$$\begin{array}{rl} & \beta_d(x_0; x^*) + A_{k+1} F^* \\ \overset{(4.1.43)}{\geq} & h_{k+1}(x^*) + A_k \psi(x_k) + C_k(x^*) \\ \geq & h_{k+1}(z_{k+1}) + \beta_d(z_{k+1}; x^*) + A_k \psi(x_k) + C_k(x^*) \\ \geq & A_{k+1} F\left(\frac{a_{k+1} z_{k+1} + A_k x_k}{A_{k+1}}\right) + \beta_d(z_{k+1}; x^*) + C_k(x^*). \end{array} \tag{4.1.52}$$

Hence,

$$\begin{array}{rcl} \frac{1}{2^{p-1}(p+1)} \|z_{k+1} - x^*\|^{p+1} & \leq & \beta_d(z_{k+1}; x^*) \\ & \overset{(4.1.52)}{\leq} & \beta_d(x_0; x^*) - C_k(x^*) \quad \leq \quad \alpha_k \\ & \overset{(4.1.50),(4.1.51)}{\leq} & \left(\Delta_2 + \Delta_3 \sum\limits_{i=1}^{k} \|z_i - x^*\|^{p-1} \xi_i\right)^{\frac{p+1}{p}}, \end{array} \tag{4.1.53}$$

with

$$\Delta_2 \overset{\text{def}}{=} \alpha_0^{\frac{p}{p+1}} + 2^{\frac{p-1}{p+1}} (p+1)^{\frac{1}{p+1}} \Delta_1, \text{ and } \Delta_3 \overset{\text{def}}{=} 2^{\frac{p-1}{p+1}} (p+1)^{\frac{1}{p+1}} p 4^{p-2}.$$

For the monotone sequence $\gamma_k \overset{\text{def}}{=} \Delta_2 + \Delta_3 \sum\limits_{i=1}^{k} \|z_i - x^*\|^{p-1} \xi_i$, it holds

$$\begin{array}{rcl} \gamma_{k+1} & = & \gamma_k + \Delta_3 \|z_{k+1} - x^*\|^{p-1} \xi_{k+1} \\ & \overset{(4.1.53)}{\leq} & \gamma_k + \Delta_3 2^{\frac{(p-1)^2}{p+1}} (p+1)^{\frac{p-1}{p+1}} \gamma_k^{\frac{p-1}{p}} \xi_{k+1} \\ & \leq & \gamma_k + \Delta_3 2^{\frac{(p-1)^2}{p+1}} (p+1)^{\frac{p-1}{p+1}} \gamma_{k+1}^{\frac{p-1}{p}} \xi_{k+1}. \end{array}$$

By dividing both sides by $\gamma_{k+1}^{\frac{p-1}{p}}$, and by using monotonicity again, we obtain

$$\gamma_{k+1}^{\frac{1}{p}} \leq \gamma_k^{\frac{1}{p}} + \Delta_3 2^{\frac{(p-1)^2}{p+1}} (p+1)^{\frac{p-1}{p+1}} \xi_{k+1}, \tag{4.1.54}$$

Telescoping which, gives

$$\begin{aligned}
\alpha_k^{\frac{1}{p+1}} \quad &\overset{(4.1.53)}{\leq} \quad \gamma_k^{\frac{1}{p}} \quad \overset{(4.1.54)}{\leq} \quad \gamma_0^{\frac{1}{p}} + \Delta_3 2^{\frac{(p-1)^2}{p+1}} (p+1)^{\frac{p-1}{p+1}} \sum_{i=1}^{k} \xi_i \\
&= \quad \Delta_2^{\frac{1}{p}} + \Delta_3 2^{\frac{(p-1)p}{p+1}} (p+1)^{\frac{p}{p+1}} \sum_{i=1}^{k} \zeta_i^{\frac{1}{p+1}} \qquad (4.1.55) \\
&\overset{(1.3.9)}{\leq} \quad \Delta_2^{\frac{1}{p}} + \Delta_3 2^{\frac{(p-1)p}{p+1}} (p+1)^{\frac{p}{p+1}} c^{\frac{1}{p+1}} (p+2).
\end{aligned}$$

Note that

$$\Delta_1 \quad \leq \quad \mathcal{O}\Big(\|x_0 - x^*\|^{p-1} c^{\frac{1}{p+1}} + c^{\frac{p}{p+1}} \Big) \quad \leq \quad \mathcal{O}\Big(\|x_0 - x^*\|^{p} + c^{\frac{p}{p+1}} \Big),$$

where we used Young's inequality for products, and

$$\Delta_2 \quad \leq \quad \mathcal{O}\Big(\alpha_0^{\frac{p}{p+1}} + \Delta_1 \Big) \quad \leq \quad \mathcal{O}\Big(\|x_0 - x^*\|^{p} + c^{\frac{p}{p+1}} \Big),$$

while

$$\Delta_3 \quad \leq \quad \mathcal{O}(1).$$

Hence, from (4.1.55) we conclude that

$$\alpha_k \quad \leq \quad \mathcal{O}\Big(\|x_0 - x^*\|^{p+1} + c \Big). \qquad (4.1.56)$$

Finally,

$$F(x_k) - F^* \quad \overset{(4.1.48)}{\leq} \quad \tfrac{\alpha_k}{A_k} \quad \overset{(4.1.56)}{\leq} \quad \mathcal{O}\big(\tfrac{L_p (\|x_0 - x^*\|^{p+1} + c)}{k^{p+1}} \big).$$

Lastly, let us prove bound (4.1.38) for the number of tensor steps, needed to find v_{k+1}. We minimize $h_{k+1}(\cdot)$, starting from the previous prox-point v_k. We denote the first component of $h_{k+1}(\cdot)$, by:

$$g_{k+1}(x) \quad \stackrel{\mathrm{def}}{=} \quad A_{k+1} f\left(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\right),$$

which is a *contracted* version of the smooth part of our objective $F(x)$. Direct computation gives the following relation between Lipschitz constants for the derivatives of g_{k+1} and f:

$$\begin{array}{rcl} L_p(g_{k+1}) & = & \frac{a_{k+1}^{p+1}}{A_{k+1}^{p}} L_p(f) \quad = \quad \frac{((k+1)^{p+1} - k^{p+1})^{p+1}}{(k+1)^{p(p+1)}} \\ & \leq & \frac{((p+1)(k+1)^p)^{p+1}}{(k+1)^{p(p+1)}} \quad = \quad (p+1)^{p+1}. \end{array} \tag{4.1.57}$$

Therefore, condition number $\bar{\omega}_p$ (4.1.25) for h_{k+1} is bounded by an *absolute constant*, and we need to estimate only the value under the logarithm in (4.1.28). Due to Lemma 4.1.2, one monotone inexact tensor step $M := M_{H,\delta}(v_k)$ for function $h_{k+1}(\cdot)$ with constant $H := pL_p(g_{k+1})$ gives, for all $y \in \mathrm{dom}\, \psi$:

$$h_{k+1}(M) \quad \stackrel{(4.1.3),(4.1.57)}{\leq} \quad h_{k+1}(y) + \tfrac{(p+1)^{p+1}\|y - v_k\|^{p+1}}{p!} + \delta. \tag{4.1.58}$$

If substitute $y := x^*$ (minimizer of F) into (4.1.58), then we obtain

$$\begin{aligned}
&h_{k+1}(M) - h_{k+1}^* \\
&\overset{(4.1.58)}{\leq} \quad h_{k+1}(x^*) - h_{k+1}^* + \tfrac{(p+1)^{p+1}\|v_k - x^*\|^{p+1}}{p!} + \delta \\
&\overset{(4.1.43)}{\leq} \quad A_{k+1}F^* - A_k\psi(x_k) + \beta_d(x_0; x^*) - C_k(x^*) - h_{k+1}^* \\
&\qquad + \tfrac{(p+1)^{p+1}\|v_k - x^*\|^{p+1}}{p!} + \delta \\
&\overset{(4.1.48),(4.1.49)}{\leq} \quad A_{k+1}F^* - A_k\psi(x_k) - h_{k+1}^* + \big(1 + \tfrac{(p+1)^{p+2}2^{p-1}}{p!}\big)\alpha_k + \delta \\
&= \quad A_{k+1}F^* - \min_y\big\{h_{k+1}(y) + A_k\psi(x_k)\big\} \\
&\qquad + \big(1 + \tfrac{(p+1)^{p+2}2^{p-1}}{p!}\big)\alpha_k + \delta \\
&\leq \quad A_{k+1}F^* - \min_y\Big\{A_{k+1}F\big(\tfrac{a_{k+1}y + A_k x_k}{A_{k+1}}\big) + \beta_d(v_k; y)\Big\} \\
&\qquad + \big(1 + \tfrac{(p+1)^{p+2}2^{p-1}}{p!}\big)\alpha_k + \delta \\
&\leq \quad A_{k+1}F^* - \min_y\Big\{A_{k+1}F\big(\tfrac{a_{k+1}y + A_k x_k}{A_{k+1}}\big)\Big\} \\
&\qquad + \big(1 + \tfrac{(p+1)^{p+2}2^{p-1}}{p!}\big)\alpha_k + \delta \\
&= \quad \big(1 + \tfrac{(p+1)^{p+2}2^{p-1}}{p!}\big)\alpha_k + \delta \\
&\overset{(4.1.55)}{\leq} \quad \mathcal{O}\big(\|x_0 - x^*\|^{p+1} + c + \delta\big).
\end{aligned}$$

So, if we set $\delta := c$ and perform just one step of the monotone inexact tensor method for $h_{k+1}(\cdot)$, the remaining amount of steps N_k needed to find v_{k+1}, such that (4.1.34) holds, is bounded as:

$$N_k \overset{(4.1.28)}{\leq} \mathcal{O}\Big(\log \tfrac{h_{k+1}(M) - h_{k+1}^*}{\zeta_{k+1}}\Big) \leq \mathcal{O}\Big(\log \tfrac{k(\|x_0 - x^*\|^{p+1} + c)}{c}\Big). \qquad \square$$

Therefore, the total number of the inexact tensor steps for finding an ε-solution of the problem is bounded by $\widetilde{\mathcal{O}}\big(1/\varepsilon^{\frac{1}{p+1}}\big)$. One theoretical question remains open: is it possible to construct in the framework of inexact tensor steps, the *optimal methods* with the complexity $\mathcal{O}\big(1/\varepsilon^{\frac{2}{3p+1}}\big)$ having no hidden logarithmic factors. This would match the existing lower bound [4, 118].

4.1.4 Experiments

We now show computational results with empirical study of different accuracy policies. We consider inexact methods of order $p = 2$ (Cubic regularization of Newton method), and to solve the corresponding subproblem we use the flexible version of the Fast Gradient Method with restarts from [119]. To estimate the residual in function value of the subproblem, we use a simple stopping criterion, given by uniform convexity of the model $g(y) = \Omega_H(x; y)$:

$$g(y) - \min_y g(y) \quad \leq \quad \tfrac{4}{3}\left(\tfrac{1}{H}\right)^{1/2} \|\nabla g(y)\|_*^{3/2}. \tag{4.1.59}$$

An alternative approach would be to bound the functional residual by the duality gap[1].

We compare the *adaptive* rule for inner accuracies (4.1.15) with dynamic strategies in the form $\delta_k = 1/k^\alpha$, for different α (left graphs), and with the constant choices (right).

Logistic Regression. First, let us consider the problem of training ℓ_2-regularized logistic regression model for classification task with two classes, on several real datasets[2]: *mushrooms* ($m = 8124, n = 112$), *w8a* ($m = 49749, n = 300$), and *a8a* ($m = 22696, n = 123$)[3].

We use the standard Euclidean norm for this problem, and simple line search at every iteration, to fit the regularization parameter H. The results are shown in Figure 4.1.

Log-Sum-Exp. In the next set of experiments, we consider unconstrained minimization of the following objective:

$$f_\mu(x) \quad = \quad \mu \log\left(\sum_{i=1}^m \exp\left(\frac{\langle a_i, x\rangle - b_i}{\mu}\right)\right), \quad x \in \mathbb{R}^n,$$

where $\mu > 0$ is a *smoothing* parameter. To generate the data, we sample coefficients $\{\tilde{a}_i\}_{i=1}^m$ and b randomly from the uniform distribution on $[-1, 1]$. Then, we shift the parameters in a way to have the solution x^* in the

[1] Note that the left hand side in (4.1.59) can be bounded from below by the distance from y to the optimum of the model, using uniform convexity. Therefore, we have a computable bound for the distance to the solution of the subproblem.

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

[3] m is the number of training examples and n is the dimension of the problem (the number of features).

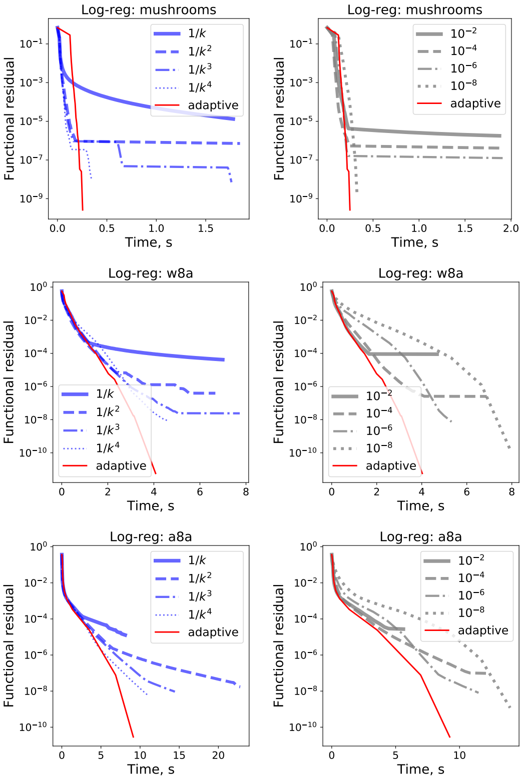


Figure 4.1: Comparison of different accuracy policies for the inexact Cubic Newton, training logistic regression.

origin. Namely, using $\{\tilde{a}_i\}_{i=1}^m$ we form a preliminary function $\tilde{f}_\mu(x)$, and set $a_i := \tilde{a}_i - \nabla \tilde{f}_\mu(0)$. Thus we essentially obtain $\nabla f_\mu(0) = 0$.

We set $m = 6n$, and $n = 100$. In the method, we use the following Euclidean norm for the primal space: $\|x\| = \langle Bx, x\rangle^{1/2}$, with the matrix $B = \sum_{i=1}^m a_i a_i^T$, and fix regularization parameter H being equal 1. The results are shown in Figure 4.2.

We see that the adaptive rule demonstrates reasonably good performance (in terms of the total computational time[4]) in all the scenarios.

[4] CPU time was evaluated on a machine with Intel Core i5 CPU, 1.6GHz; 8 GB RAM. All methods have been implemented in Python 3.7.1. Operation system: macOS 10.15. The source code can be found at `https://github.com/doikov/dynamic-accuracies/`

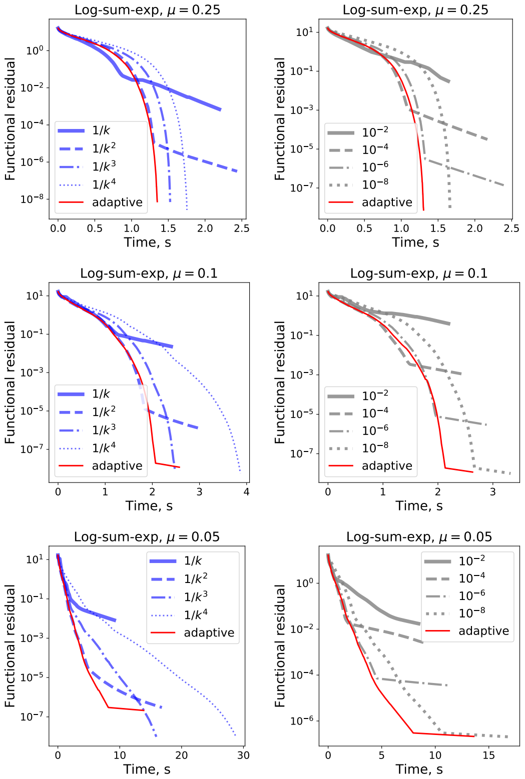


Figure 4.2: Comparison of different accuracy policies for the inexact Cubic Newton, minimizing Log-sum-exp function.

Exact Stopping Criterion. In the following set of experiments with Cubic Newton method, we compute the *exact* minimizer of the model (4.1.1), at every iteration. Then, we use this value to ensure the required precision in function value of the subproblem for the inexact step (in the previous settings we used the upper bound (4.1.59) for this purpose). The results for Log-Sum-Exp function are shown in Figures 4.3 – 4.6. The results for Logistic regression are shown in Figures 4.7 – 4.11.

We compare the iteration rate and the corresponding number of Hessian-vector products used, for the constant choice of inner accuracy (top left graphs), dynamic strategies in the form $\delta_k = 1/k^{\alpha}$ (top right), and adaptive strategies $\delta_k = (F(x_{k-1}) - F(x_k))^{\alpha}$ (center graphs). We use the names "adaptive", "adaptive 1.5" and "adaptive 2" for $\alpha = 1$, $\alpha = 3/2$, and $\alpha = 2$, respectively.

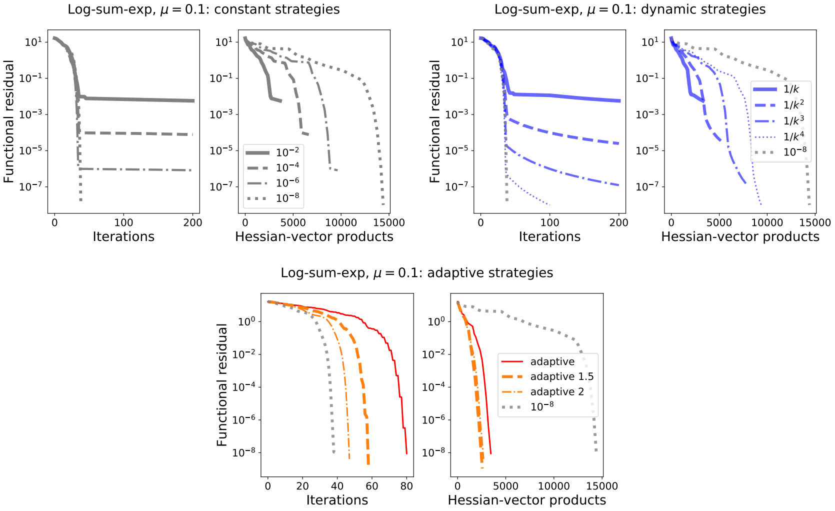


Figure 4.3: Exact stopping criterion, Log-sum-exp, $n = 100$, $\mu = 0.1$.

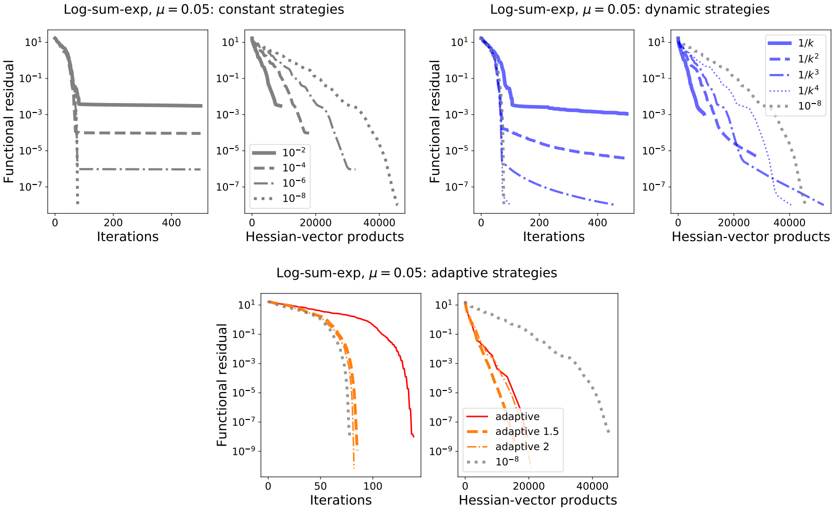


Figure 4.4: Exact stopping criterion, Log-sum-exp, $n = 100$, $\mu = 0.05$.

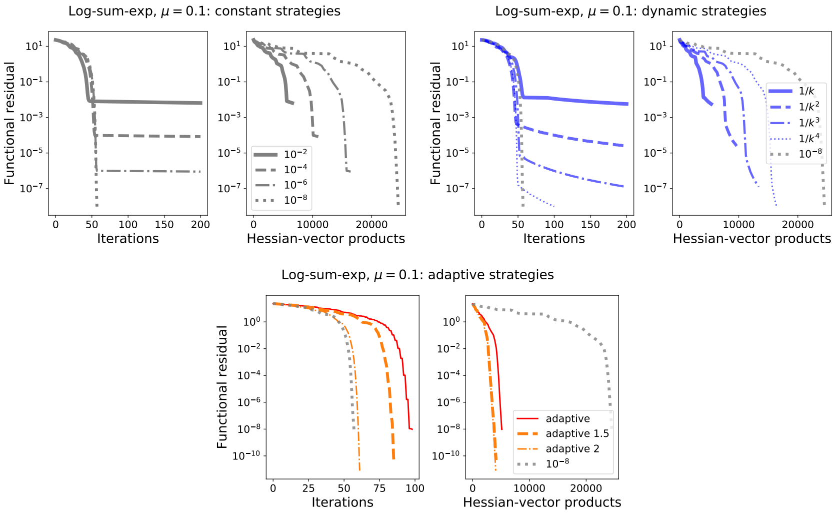


Figure 4.5: Exact stopping criterion, Log-sum-exp, $n = 200$, $\mu = 0.1$.

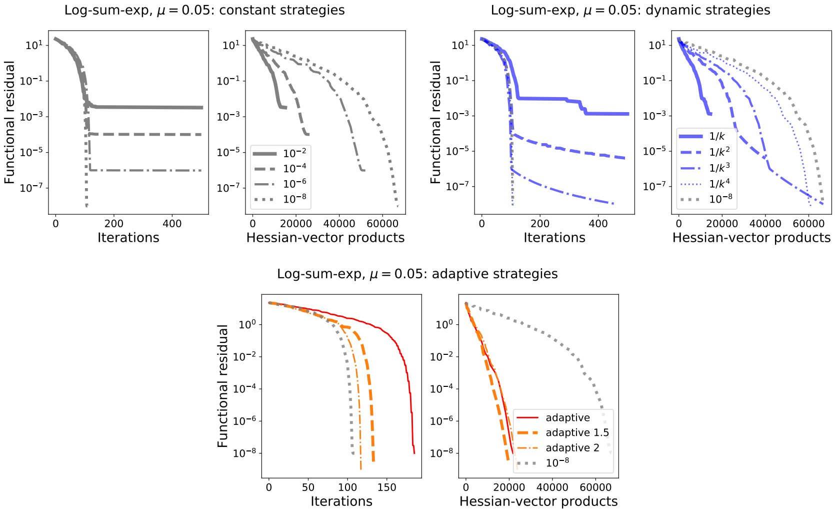


Figure 4.6: Exact stopping criterion, Log-sum-exp, $n = 200$, $\mu = 0.05$.

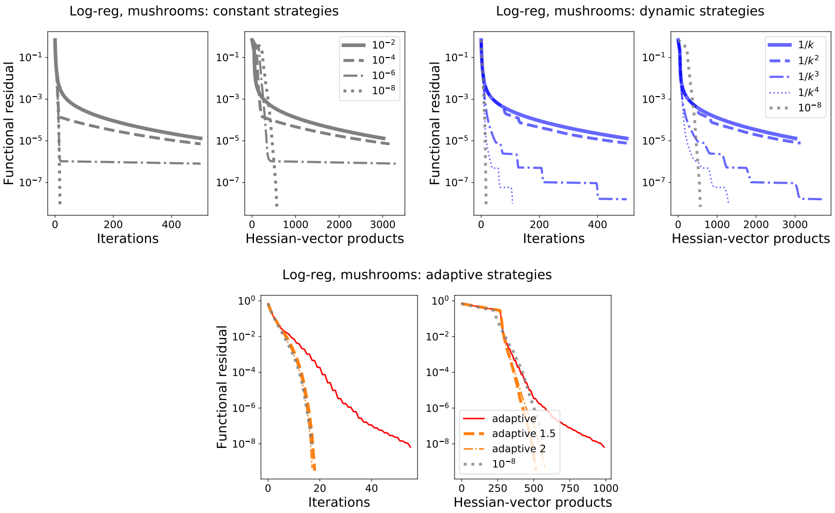


Figure 4.7: Exact stopping criterion, logistic regression, *mushrooms*.

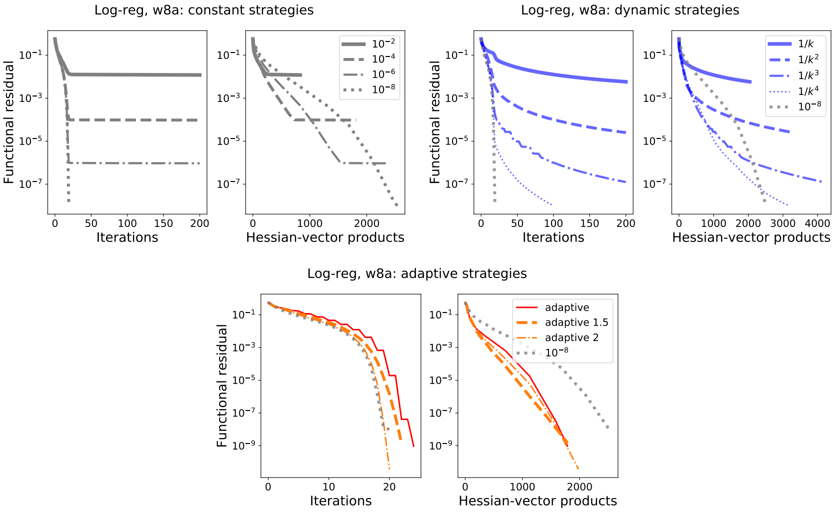


Figure 4.8: Exact stopping criterion, logistic regression, *w8a*.

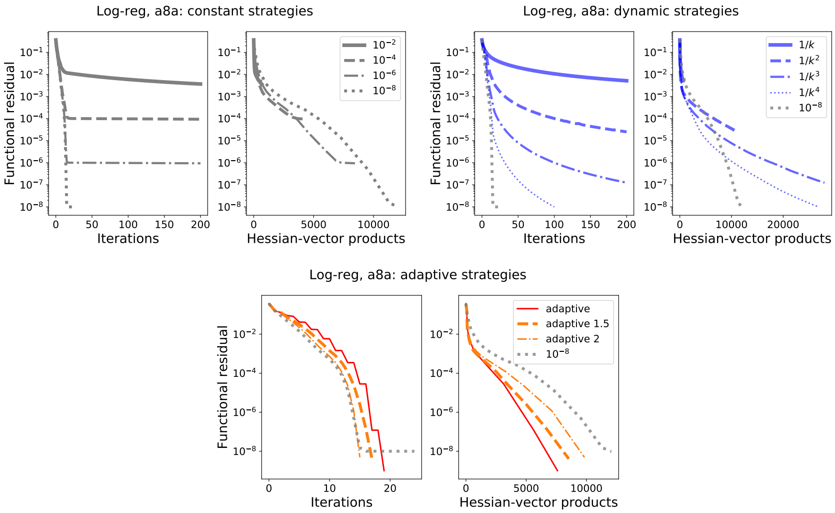


Figure 4.9: Exact stopping criterion, logistic regression, *a8a*.

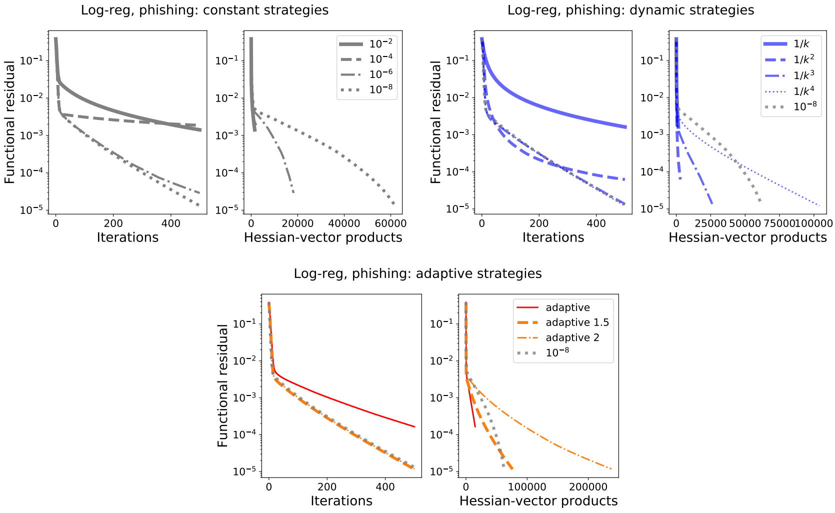


Figure 4.10: Exact stopping criterion, logistic regression, *phishing*.

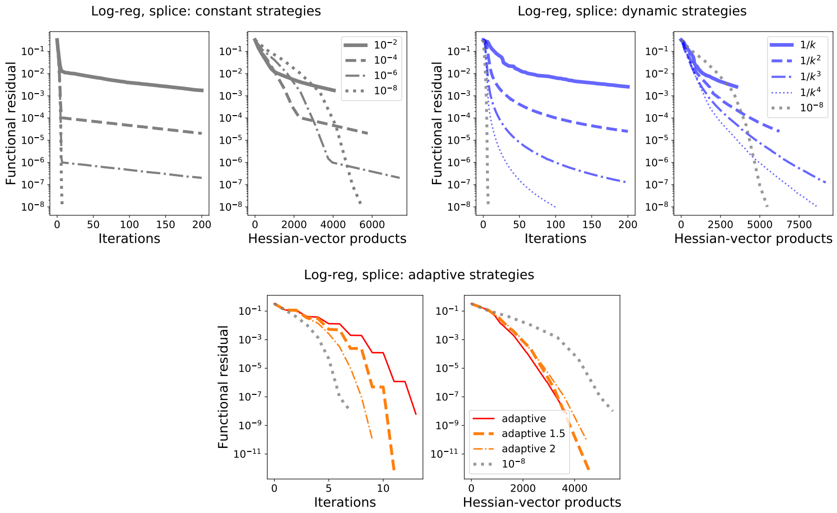


Figure 4.11: Exact stopping criterion, logistic regression, *splice*.

We see that the constant choice of inner accuracy reasonably depends on the desired precision for solving the initial problem. At the same time, the dynamic strategies are adjusting with the iterations. The best performance is achieved by the use of the adaptive policies. It is also important that in some cases we need to use "adaptive 1.5" or "adaptive 2" strategy, to have the local superlinear convergence. This confirms our theory.

Averaging and Acceleration. In this experiment, we consider unconstrained minimization of the following objective ($x^{(i)}$ indicates ith coordinate of x)

$$f(x) \quad = \quad |x^{(1)}|^3 + \sum_{i=2}^{n} |x^{(i)} - x^{(i-1)}|^3, \qquad x \in \mathbb{R}^n, \tag{4.1.60}$$

by different inexact Newton methods. Note, that the structure of (4.1.60) is similar to that one of the worst function for the second-order methods (see Chapter 4.3.1 in [117]). It is also similar to the function from Example 4.1.8.

We compare iteration rates of the following algorithms: Cubic Newton (CN) with dynamic rule $\delta_k = 1/k^3$, Cubic Newton with adaptive rule (4.1.15), the method with Averaging (algorithm (4.1.31)) with $\delta_k = 1/k^3$, and the accelerated method with Contracting proximal iterations (algorithm (4.1.35)). For the latter one we use $\zeta_k = 1/k$ and $\delta_k = 1/k$, as the rules for choosing the accuracy of inexact (outer) proximal steps, and inexact (inner) Newton steps, respectively.[5]

For the first three algorithms, we also compare the constant choice for the regularization parameter: $H = 1$ (on the top graphs), and a simple line search[6] for choosing H at every iteration (bottom). The results are shown in Figure 4.12.

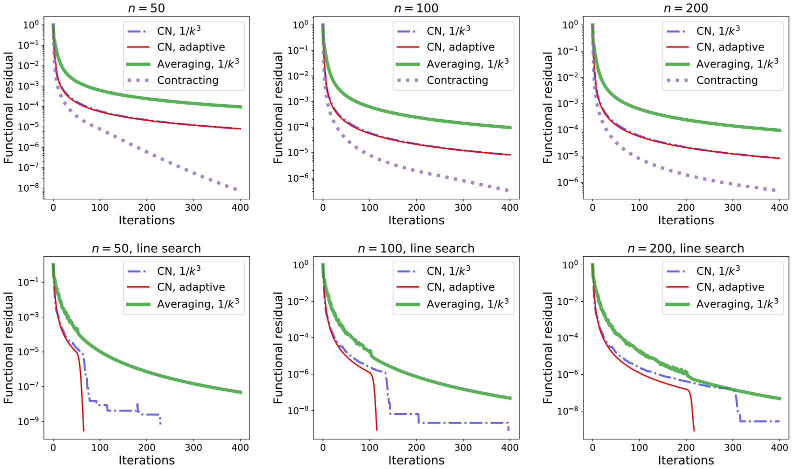


Figure 4.12: Methods with averaging and acceleration.

We see that all the methods have a *sublinear* rate of convergence, until the iteration counter is smaller than the dimension of the problem. The use of the line search significantly helps for improving the rate. Thus, it seems to be an important open question (which we keep for the further research) — to equip the contracting proximal scheme (algorithm (4.1.35)) with a variant of line search as well.

[5] In our experiments, there is no need of high precision for the inexact contracting proximal steps. A faster decrease of δ_k did not improve the rate of convergence.

[6] Namely, we multiply H by the factor of two, until condition $F(T_{H,\delta}(x^k)) \leq M_H(x^k; T_{H,\delta}(x^k))$ is satisfied. At the next iteration, we start the line search from the previous estimate of H, divided by two. See also algorithm (2.1.22) in Chapter 2.

4.2 Inexact Contracting Newton Method

The Contracting Newton Method (3.2.10) was developed for solving the composite convex minimization problem,

$$\min_{x \in \operatorname{dom} \psi} \Bigl\{ F(x) \quad = \quad f(x) + \psi(x) \Bigr\}$$

with *bounded* domain of ψ. Therefore, we assume that $\operatorname{dom} \psi$ is a compact convex set. At each iteration of the method, we need to find a solution to the following auxiliary subproblem:

$$\min_{y} \Bigl\{ \langle \nabla f(x_k), y - x_k \rangle + \tfrac{\gamma_k}{2} \langle \nabla^2 f(x_k)(y - x_k), y - x_k \rangle + \psi(y) \Bigr\},$$

where $\gamma_k \in (0, 1]$ is a contraction parameter and x_k is the current iterate. This is minimization of the quadratic function over the composite part, and it can be nontrivial to solve for a general ψ.

The subproblem with linear objective is usually called the *linear minimization oracle* for ψ, that is for a certain $s \in \mathbb{E}^*$:

$$\min_{y} \bigl\{ \langle s, y \rangle + \psi(y) \bigr\}. \tag{4.2.1}$$

Clearly, this is much easier to solve than the previous one. In this section, we investigate the idea of performing inexact Contracting Newton steps by using only the operations of type (4.2.1).

Let us recall the affine-invariant smoothness characteristics of the objective, introduced in Section 3.1.2.

$$\begin{aligned} \Delta^{(2,1)}_{\operatorname{dom} \psi}(f) \quad &\overset{\text{def}}{=} \quad \sup_{\substack{x, v \in \operatorname{dom} \psi, \\ t \in (0,1]}} \tfrac{1}{t^3} \Bigl| f(x + t(v - x)) - f(x) \\ &\qquad - t \langle \nabla f(x), v - x \rangle - \tfrac{t^2}{2} \langle \nabla^2 f(x)(v - x), v - x \rangle \Bigr| \\ &\overset{(3.1.15)}{\leq} \quad \tfrac{1}{6} \mathcal{V}^{(3)}_{\operatorname{dom} \psi}(f) \quad \overset{\text{def}}{=} \quad \sup_{x, y, v \in \operatorname{dom} \psi} \tfrac{1}{6} |D^3 f(y)[v - x]^3|, \end{aligned}$$

and

$$\mathcal{V}^{(2)}_{\operatorname{dom} \psi}(f) \quad \overset{\text{def}}{=} \quad \sup_{x, y, v \in \operatorname{dom} \psi} |D^2 f(y)[v - x]^2|.$$

In Section 4.2.1 we present an implementation of the Contracting Newton, when at each step we solve the subproblem inexactly by a variant of first-order Conditional Gradient Method. In Section 4.2.2 we address effective implementation of our algorithm for the standard Simplex. Section 4.2.3 contains numerical experiments.

4.2.1 Two-Level Scheme

The entire algorithm looks as follows.

```
Inexact Contracting-Point Newton Method

Initialization. Choose $x_0 \in \mathrm{dom}\,\psi$, $c > 0$.

Iteration $k \geq 0$.

1: Choose $\gamma_k \in (0, 1]$.

2: Denote the subproblem objective:

   $g_k(v) = \langle \nabla f(x_k), v - x_k \rangle + \frac{\gamma_k}{2} \langle \nabla^2 f(x_k)(v - x_k), v - x_k \rangle$.

3: Initialize inner method $t = 0$, $z_0 = x_k$, $\phi_0(w) \equiv 0$.
4-a: Set $\alpha_t = \frac{2}{t+2}$.
4-b: Set $\phi_{t+1}(w) = \alpha_t \big[ g_k(z_t) + \langle \nabla g_k(z_t), w - z_t \rangle + \psi(w) \big]$
                 $+ (1 - \alpha_t)\phi_t(w)$.
4-c: Compute $w_{t+1} \in \mathop{\mathrm{Argmin}}_{w} \phi_{t+1}(w)$.
4-d: Set $z_{t+1} = \alpha_t w_{t+1} + (1 - \alpha_t) z_t$.
4-e: If $g_k(z_{t+1}) + \psi(z_{t+1}) - \phi_{t+1}(w_{t+1}) > c\gamma_k^2$, then
          Set $t = t + 1$ and go to 4-a, else go to 5.

5: Set $\bar{x}_{k+1} = \gamma_k z_{t+1} + (1 - \gamma_k) x_k$.

6: If $F(\bar{x}_{k+1}) \leq F(x_k)$, then set $x_{k+1} = \bar{x}_{k+1}$.
   Else choose $x_{k+1} = x_k$.
```

(4.2.2)

We provide an analysis of the total number of *oracle calls* for f (step 2) and the total number of *linear minimization oracle calls* for the composite component ψ (step 4-c), required to solve the initial problem up to the given accuracy level.

Theorem 4.2.1. *Let $\gamma_k = \frac{3}{k+3}$. Then, for iterations $\{x_k\}_{k \geq 1}$ generated by method (4.2.2), we have*

$$F(x_k) - F^* \quad \leq \quad 27 \cdot \bigl(c + 2\Delta_{\mathrm{dom}\,\psi}^{(2,1)}(f)\bigr) \cdot k^{-2}. \tag{4.2.3}$$

Therefore, for any $\varepsilon > 0$, it is enough to perform

$$K \quad = \quad \left\lceil \sqrt{\tfrac{27(c+2\Delta_{\mathrm{dom}\,\psi}^{(2,1)}(f))}{\varepsilon}} \, \right\rceil \tag{4.2.4}$$

iterations of the method, in order to get $F(x_K) - F^* \leq \varepsilon$. And the total number N_K of linear minimization oracle calls during these iterations is bounded as

$$N_K \quad \leq \quad 2 \cdot \Bigl(1 + \tfrac{2\mathcal{V}_{\mathrm{dom}\,\psi}^{(2)}(f)}{c}\Bigr) \cdot \Bigl(1 + \tfrac{27(c+2\Delta_{\mathrm{dom}\,\psi}^{(2,1)}(f))}{\varepsilon}\Bigr). \tag{4.2.5}$$

Proof. Let us fix arbitrary iteration $k \geq 0$ of our method and consider the following objective:

$$\begin{aligned} m_k(v) \quad &= \quad g_k(v) + \psi(v) \\ &= \quad \langle \nabla f(x_k), v - x_k \rangle + \tfrac{\gamma_k}{2} \langle \nabla^2 f(x_k)(v - x_k), v - x_k \rangle + \psi(v). \end{aligned}$$

We need to find a point $\bar{v}_{k+1}$ such that

$$m_k(\bar{v}_{k+1}) - m_k^* \quad \leq \quad c\gamma_k^2. \tag{4.2.6}$$

Note that if we set $\bar{x}_{k+1} := \gamma_k \bar{v}_{k+1} + (1 - \gamma_k)x_k$, then from (4.2.6) we have bound (3.1.21) satisfied with $\xi_{k+1} = c\gamma_k^3$. Thus we get one step of algorithm (3.1.22) for $p = 2$, and Theorem 3.1.9 gives the required rate of convergence (4.2.3). We are about to show that steps 4-a – 4-e of our algorithm aim to find such a point $\bar{v}_{k+1}$.

Let us introduce the auxiliary sequences $A_t \overset{\mathrm{def}}{=} t \cdot (t+1)$ and $a_{t+1} \overset{\mathrm{def}}{=} A_{t+1} - A_t$ for $t \geq 0$. Then, $\alpha_t \equiv \frac{a_{t+1}}{A_{t+1}}$, and we have the following representation of the *Estimating Functions*, for every $t \geq 0$

$$\phi_{t+1}(w) \quad = \quad \tfrac{1}{A_{t+1}} \sum_{i=0}^{t} a_{i+1} \Bigl[g_k(z_i) + \langle \nabla g_k(z_i), w - z_i \rangle + \psi(w) \Bigr].$$

By convexity of $g_k(\cdot)$, we have

$$m_k(w) \quad \geq \quad \phi_{t+1}(w), \qquad w \in \mathrm{dom}\,\psi.$$

Therefore, we obtain the following upper bound for the residual (4.2.6), for any $v \in \mathrm{dom}\,\psi$

$$m_k(v) - m_k^* \quad \leq \quad m_k(v) - \phi_{t+1}^*, \tag{4.2.7}$$

where $\phi_{t+1}^* = \min_w \phi_{t+1}(w) = \phi_{t+1}(w_{t+1})$.

Now, let us show by induction, that

$$A_t \phi_t^* \quad \geq \quad A_t m_k(z_t) \; - \; B_t, \qquad t \geq 0, \tag{4.2.8}$$

for $B_t := \frac{\gamma_k \mathcal{V}^{(2)}_{\mathrm{dom}\,\psi} h(f)}{2} \sum_{i=0}^{t} \frac{a_{i+1}^2}{A_{i+1}}$. It obviously holds for $t = 0$. Assume that it holds for some $t \geq 0$. Then,

$$\begin{array}{rcl}
A_{t+1}\phi_{t+1}^* & = & A_{t+1}\phi_{t+1}(w_{t+1}) \\[1ex]
& = & A_t \phi_t(w_{t+1}) + a_{t+1}\bigl[g_k(z_t) + \langle \nabla g_k(z_t), w_{t+1} - z_t \rangle + \psi(w_{t+1}) \bigr] \\[1ex]
& \overset{(4.2.8)}{\geq} & A_t m_k(z_t) + a_{t+1}\bigl[g_k(z_t) + \langle \nabla g_k(z_t), w_{t+1} - z_t \rangle + \psi(w_{t+1}) \bigr] - B_t \\[1ex]
& = & A_{t+1}\Bigl[g_k(z_t) + \alpha_t \langle \nabla g_k(z_t), w_{t+1} - z_t \rangle + \alpha_t \psi(w_{t+1}) \\[1ex]
& & \qquad + \, (1 - \alpha_t)\psi(z_t) \Bigr] - B_t \\[1ex]
& \geq & A_{t+1}\Bigl[g_k(z_t) + \alpha_t \langle \nabla g_k(z_t), w_{t+1} - z_t \rangle + \psi(z_{t+1}) \Bigr] - B_t.
\end{array}$$

Note that

$$\begin{array}{rcl}
g_k(z_{t+1}) & = & g_k(z_t + \alpha_t(w_{t+1} - z_t)) \\[1ex]
& = & g_k(z_t) + \alpha_t \langle \nabla g_k(z_t), w_{t+1} - z_t \rangle \\[1ex]
& & \qquad + \, \frac{\alpha_t^2 \gamma_k}{2} \langle \nabla^2 f(x_k)(w_{t+1} - z_t), w_{t+1} - z_t \rangle.
\end{array}$$

Therefore, we obtain

$$A_{t+1}\phi^*_{t+1} \quad \geq \quad A_{t+1}m_k(z_{t+1}) - B_t - \tfrac{a_{t+1}^2}{A_{t+1}} \cdot \tfrac{\gamma_k \mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{2},$$

and this is (4.2.8) for the next step. Therefore, we have (4.2.8) established for all $t \geq 0$.

Combining (4.2.7) with (4.2.8), we get the following guarantee for the inner steps 4-a – 4-e:

$$\begin{array}{rcl} m_k(z_{t+1}) - m_k^* & \leq & m_k(z_{t+1}) - \phi^*_{t+1} \quad \leq \quad \tfrac{\gamma_k \mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{2A_{t+1}} \sum\limits_{i=0}^{t} \tfrac{a_{i+1}^2}{A_{i+1}} \\ & \leq & \tfrac{2\gamma_k \mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{t+1}. \end{array}$$

Therefore, all iterations of our method are well-defined. We exit from the inner loop on step 4-e after

$$t \quad \geq \quad \tfrac{2\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{c\gamma_k} - 1 \quad = \quad \tfrac{2(k+3)\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{3c} - 1, \tag{4.2.9}$$

and the point $\bar{v}_{k+1} \equiv z_{t+1}$ satisfies (4.2.6).

Hence, we obtain (4.2.3) and (4.2.4). The total number of linear minimization oracle calls can be estimated as follows

$$\begin{array}{rcl} N_K & \overset{(4.2.9)}{\leq} & \sum\limits_{k=0}^{K-1}\Big(1 + \tfrac{2(k+3)\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{3c}\Big) \quad = \quad K\Big(1 + \tfrac{\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{3c}\big(K+5\big)\Big) \\ & \leq & K^2\Big(1 + \tfrac{2\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{c}\Big) \\ & \leq & 2 \cdot \Big(1 + \tfrac{2\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)}{c}\Big) \cdot \Big(1 + \tfrac{27(c + 2\Delta^{(2,1)}_{\mathrm{dom}\,\psi}(f))}{\varepsilon}\Big). \qquad \Box \end{array}$$

According to the result of Theorem 4.2.1, in order to solve the initial problem up to $\varepsilon > 0$ accuracy, we need to perform $\mathcal{O}(\frac{1}{\varepsilon})$ total computations of step 4-c of the method (estimate (4.2.5)). This is the same amount of linear minimization oracle calls, as required in the classical Frank-Wolfe algorithm (the case $p = 1$ of the Contracting-Point Tensor Method (3.1.22)). However, this estimate can be over-pessimistic for our two-level scheme. Indeed, it comes as the product of the worst-case complexity bounds for the outer and the inner optimization processes. It seems to be very rare to meet with the worst-case instance at the both levels simultaneously. Thus, the practical performance of our method can be much better.

At the same time, the total number of gradient and Hessian computations is only $\mathcal{O}(\frac{1}{\varepsilon^{1/2}})$ (estimate (4.2.4)). This can lead to a significant acceleration over first-order Frank-Wolfe algorithm, when the gradient computation is a bottleneck (see our experimental comparison in the next section).

The only parameter which remains to choose in method (4.2.2), is the tolerance constant $c > 0$. Note that the right hand side of (4.2.5) is convex in c. Hence, its approximate minimization provides us with the following choice

$$c \;=\; 2\sqrt{\mathcal{V}^{(2)}_{\mathrm{dom}\,\psi}(f)\,\Delta^{(2,1)}_{\mathrm{dom}\,\psi}(f)}.$$

In practical applications, we may not know some of these constants. However, in many cases they are small. Therefore, an appropriate choice of c is a small constant.

Note that the only use of the Hessian in our method is step 4-b, that computes a Hessian-vector product. This operation can be implemented by using the first-order oracle (see Section 1.6), and thus method (4.2.2) can be viewed as a *first-order scheme*. A proper implementation of the method must take into account the structure of the problem, such as sparsity of the Hessian and geometry of the domain.

4.2.2 Minimization over the Simplex

Let us discuss efficient implementation of our method, when the composite part is $\{0, +\infty\}$-indicator of the standard simplex:

$$\mathrm{dom}\,\psi \;=\; \mathbb{S}_n \;\overset{\mathrm{def}}{=}\; \Big\{ x \in \mathbb{R}^n_+ \;:\; \sum\limits_{i=1}^n x^{(i)} = 1 \Big\}. \tag{4.2.10}$$

This is an example of the set with finite number of *atoms*, which are the standard coordinate vectors in this case:

$$\mathbb{S}_n \;=\; \mathrm{Conv}\,\{e_1, \ldots, e_n\}.$$

See [73] for more examples of atomic sets in the context of the Frank-Wolfe algorithm. The maximization of a convex function over such sets can be implemented very efficiently, since the maximum is always at the corner (one of the atoms).

At iteration $k \geq 0$ of method (4.2.2), we need to minimize over $\mathbb{S}_n$ the quadratic function

$$g_k(v) \quad = \quad \langle \nabla f(x_k), v - x_k \rangle + \tfrac{\gamma_k}{2} \langle \nabla^2 f(x_k)(v - x_k), v - x_k \rangle,$$

whose gradient is

$$\nabla g_k(v) = \nabla f(x_k) + \gamma_k \nabla^2 f(x_k)(v - x_k).$$

Assume that we keep the vector $\nabla g_k(z_t) \in \mathbb{R}^n$ for the current point z_t, $t \geq 0$ of the inner process, as well as its aggregation

$$h_t \quad \overset{\text{def}}{=} \quad \alpha_t \nabla g_k(z_t) + (1 - \alpha_t) h_{t-1}, \qquad h_{-1} \quad \overset{\text{def}}{=} \quad 0 \in \mathbb{R}^n.$$

Then, at step 4-c we need to compute a vector

$$w_{t+1} \quad \in \quad \mathop{\mathrm{Argmin}}_{w \in \mathbb{S}_n} \langle h_t, w \rangle \quad = \quad \mathrm{Conv} \Bigl\{ e_j \; : \; j \in \mathop{\mathrm{Argmin}}_{1 \leq j \leq n} h_t^{(j)} \Bigr\}.$$

It is enough to find an index j of a minimal element of h_t and to set $w_{t+1} := e_j$. The new gradient is equal to

$$\begin{array}{rcl} \nabla g_k(z_{t+1}) & \overset{\text{Step 4-d}}{=} & \nabla g_k(\alpha_t w_{t+1} + (1 - \alpha_t) z_t) \\ & = & \alpha_t \Bigl(\nabla f(x_k) + \gamma_k \nabla^2 f(x_k)(e_j - x_k) \Bigr) + (1 - \alpha_t) \nabla g_k(z_t), \end{array}$$

and the function value can be expressed using the gradient as follows

$$g_k(z_{t+1}) \quad = \quad \tfrac{1}{2} \langle \nabla f(x_k) + \nabla g_k(z_{t+1}), z_{t+1} - x_k \rangle.$$

The product $\nabla^2 f(x_k) e_j$ is just j-th column of the matrix. Hence, preparing in advance the following objects: $\nabla f(x_k) \in \mathbb{R}^n$, $\nabla^2 f(x_k) \in \mathbb{R}^{n \times n}$ and the Hessian-vector product $\nabla^2 f(x_k) x_k \in \mathbb{R}^n$, we are able to perform iteration of the inner loop (steps 4-a – 4-e) very efficiently in $\mathcal{O}(n)$ arithmetical operations.

4.2.3 Experiments

Let us consider the problem of minimizing the log-sum-exp function (Soft-Max)

$$f_\mu(x) \quad = \quad \mu \log\biggl(\sum_{i=1}^m \exp\Bigl(\tfrac{\langle a_i, x\rangle - b_i}{\mu} \Bigr) \biggr), \qquad x \in \mathbb{R}^n,$$

over the standard simplex $\mathbb{S}_n$ (4.2.10). Coefficients $\{a_i\}_{i=1}^m$ and b are generated randomly from the uniform distribution on $[-1, 1]$. We compare the performance of the Inexact Contracting-Point Newton Method (4.2.2) with that one of the classical Frank-Wolfe algorithm, for different values of the parameters.

The results are shown in Figures 4.13 – 4.15.

For fair comparison, we use the Frank-Wolfe algorithm with a predefined sequence of step sizes. It is known that the use of a line search can improve the empirical rate of convergence. It seems to be an important task to equip the Contracting Newton with efficient line search procedure, which we keep for further investigation.

We see that the new method works significantly better in terms of the outer iterations (oracle calls). This confirms our theory.

At the same time, for many values of the parameters, it shows better performance in terms of the total computational time as well[7].

[7] CPU time was evaluated on a machine with Intel Core i5 CPU, 1.6GHz; 8 GB RAM. The methods have been implemented in Python 3.7.1. Operation system: macOS 10.15. The source code can be found at `https://github.com/doikov/logsumexp-simplex/`

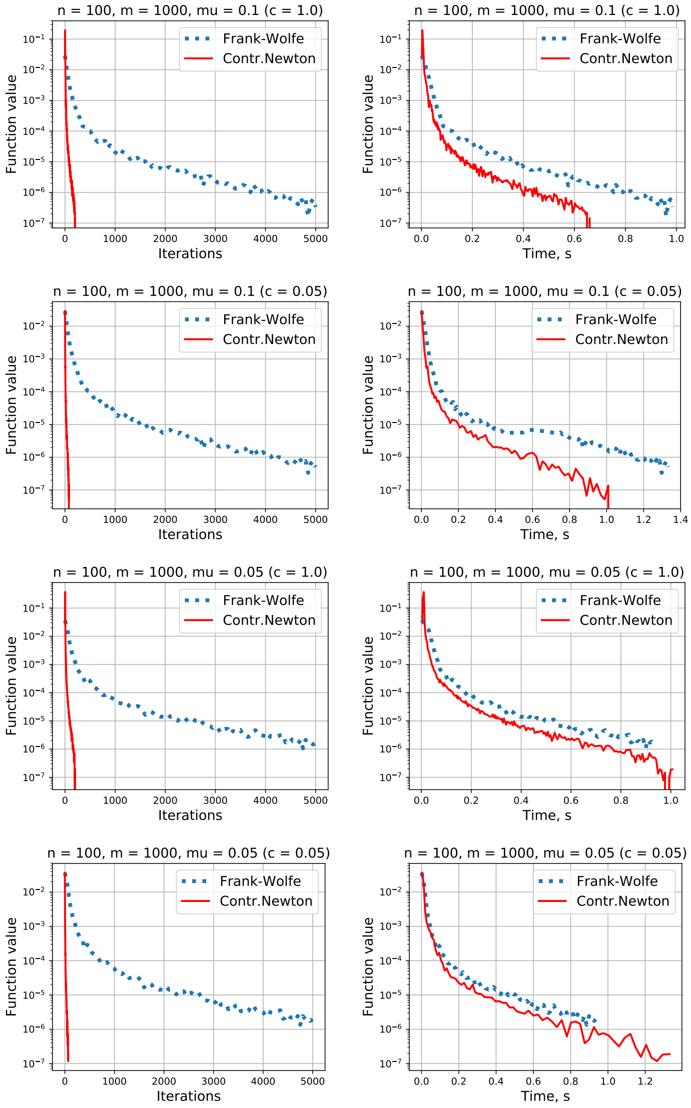


Figure 4.13: $n = 100,\ m = 1000.$

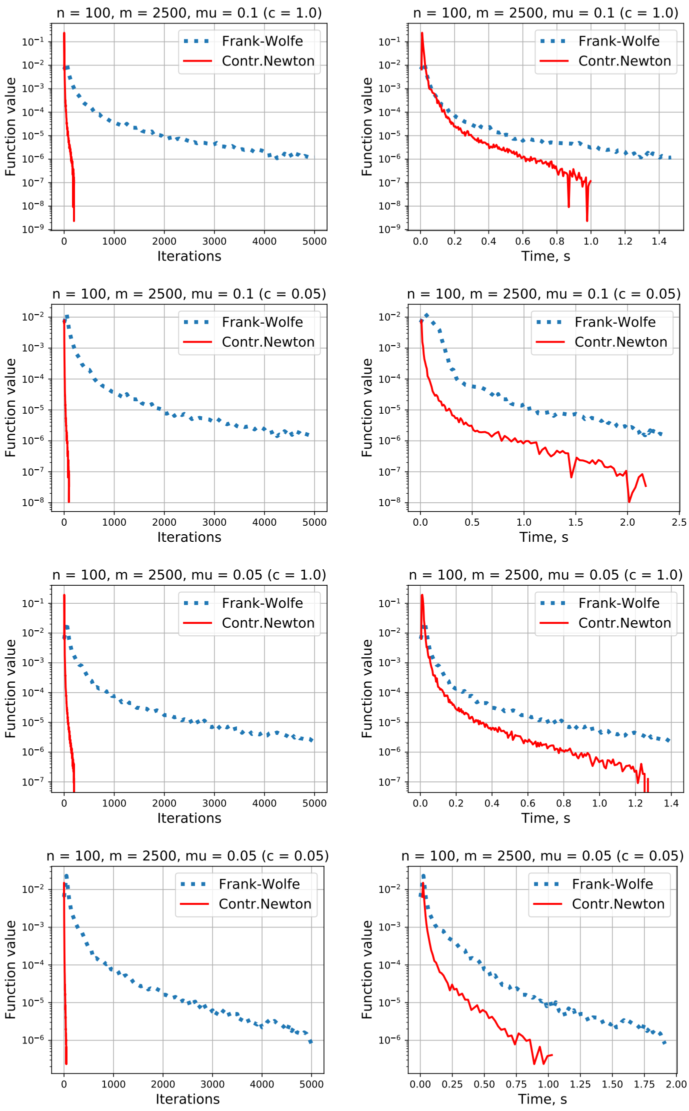
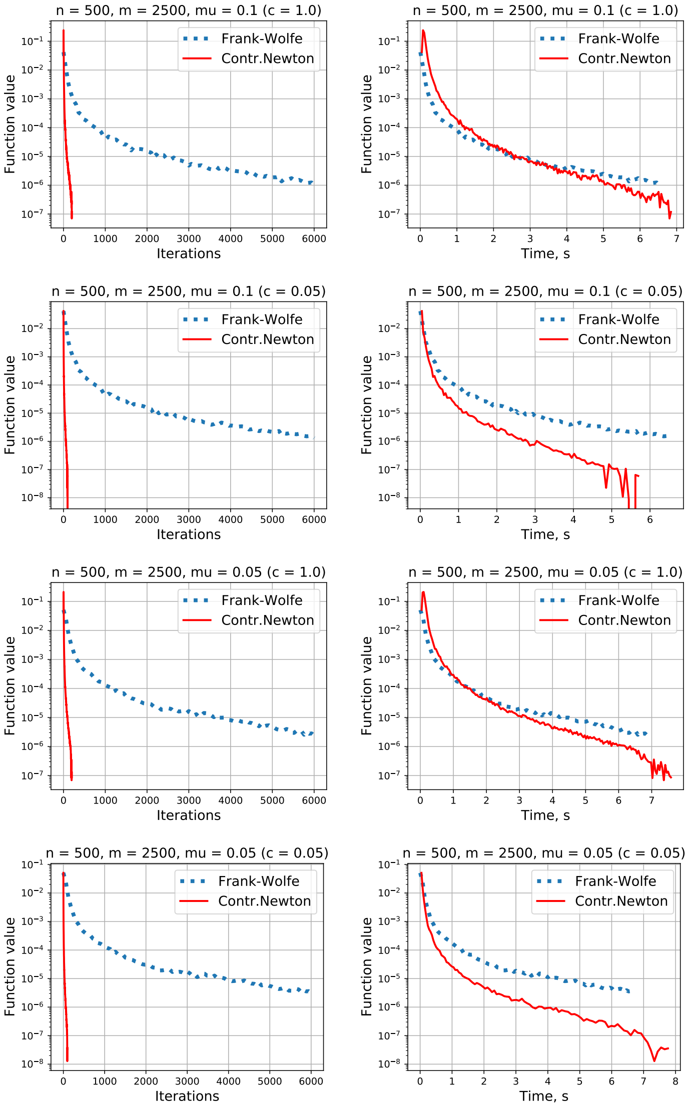


Figure 4.14: $n = 100,\ m = 2500.$


n = 500, m = 2500, mu = 0.1 (c = 1.0)

n = 500, m = 2500, mu = 0.1 (c = 1.0)

n = 500, m = 2500, mu = 0.1 (c = 0.05)

n = 500, m = 2500, mu = 0.1 (c = 0.05)

n = 500, m = 2500, mu = 0.05 (c = 1.0)

n = 500, m = 2500, mu = 0.05 (c = 1.0)

n = 500, m = 2500, mu = 0.05 (c = 0.05)

n = 500, m = 2500, mu = 0.05 (c = 0.05)

Figure 4.15: $n = 500,\ m = 2500.$

4.3 Stochastic Contracting Newton Method

Now, let us study the case when the smooth part f of the objective is represented as a sum of M convex two times continuously-differentiable components,

$$f(x) \quad := \quad \tfrac{1}{M}\sum\limits_{i=1}^{M} f_i(x). \tag{4.3.1}$$

This setting appears in many Machine Learning applications, such as *empirical risk minimization.* Often, the number M is very big. Thus, it becomes expensive to evaluate the whole gradient or the Hessian at every iteration (see the discussion on arithmetical complexity of oracles in Section 1.6). Hence, *stochastic* or *incremental* methods are the methods of choice in this situation. See [12] for a survey of the first-order incremental methods. The Newton-type Incremental Method with superlinear local convergence was proposed in [134]. Local linear rates of stochastic Newton methods were studied in [85]. Global convergence of sub-sampled Newton schemes, based on the Damped iterations, and on the Cubic regularization, was established in [139, 83, 150, 153, 154].

In this section, we develop stochastic variants of the Contracting Newton Method (3.2.10) suitable for solving the finite-sum minimization problems, when M is big.

Let us assume for simplicity that $\mathbb{E} = \mathbb{R}^n$ and the norm $\|\cdot\|$ is the standard Euclidean:

$$\|x\| \quad = \quad \sqrt{\sum\limits_{i=1}^{n}(x^{(i)})^2}.$$

We denote by $\mathscr{D}$ the diameter of the compact convex set $\mathrm{dom}\,\psi$:

$$\mathscr{D} \quad = \quad \max_{x,y \in \mathrm{dom}\,\psi} \|x-y\| \quad < \quad +\infty.$$

Let us denote by L_0 the Lipschitz constant for our objective:

$$|f(x) - f(y)| \quad \leq \quad L_0\|x-y\|, \qquad x, y \in \mathrm{dom}\,\psi,$$

and by L_1 and L_2 the Lipschitz constants on $\mathrm{dom}\,\psi$ for the gradient and for the Hessian, respectively (see definition (1.3.3)). Note that all these constants are well-defined since, by our assumption, the domain is bounded. For a random element ξ, we denote its expectation by $\boldsymbol{E}\bigl[\xi\bigr]$.

In Section 4.3.1, we study a basic stochastic version of the Contracting Newton. In Section 4.3.2, we incorporate the variance-reduction for the stochastic gradients into our scheme, Section 4.3.3 contains numerical experiments.

4.3.1 Basic Stochastic Scheme

The basic idea of stochastic algorithms is to substitute the true gradients and Hessians by some random unbiased estimators g_k, and H_k, respectively, with $\boldsymbol{E}\big[g_k\big] = \nabla f(x_k)$ and $\boldsymbol{E}\big[H_k\big] = \nabla^2 f(x_k)$.

Let us consider the following general iterations, for solving the composite optimization problem:

$$\boxed{\begin{array}{rcl} x_{k+1} & \in & \operatorname*{Argmin}\limits_{y} \Big\{ \langle g_k, y - x_k \rangle + \tfrac{1}{2}\langle H_k(y - x_k), y - x_k \rangle \\ & & \qquad + \gamma_k \psi(x_k + \tfrac{1}{\gamma_k}(y - x_k)) \Big\}, \qquad k \geq 0 \end{array}} \tag{4.3.2}$$

where $\gamma_k \in (0, 1]$ is a parameter. This is algorithm (3.2.10) with substituted vector g_k and matrix H_k instead of the true gradient and the Hessian.

First, we need to study the convergence of this process, assuming that g_k and H_k are *arbitrary*. As before, we use a sequence of positive numbers $\{a_k\}_{k \geq 1}$, and set

$$\gamma_k \quad \overset{\text{def}}{=} \quad \tfrac{a_{k+1}}{A_{k+1}}, \qquad A_k \quad \overset{\text{def}}{=} \quad \sum\limits_{i=1}^{k} a_i.$$

Lemma 4.3.1. *For iterations* (4.3.2)*, we have for all* $k \geq 1$

$$F(x_k) - F^* \quad \leq \quad \tfrac{B_k}{A_k}, \tag{4.3.3}$$

with

$$\begin{array}{rcl} B_k & \overset{\text{def}}{=} & \tfrac{L_2 \mathscr{D}^3}{2} \sum\limits_{i=0}^{k-1} \tfrac{a_{i+1}^3}{A_{i+1}^2} + \mathscr{D} \sum\limits_{i=0}^{k-1} a_{i+1} \|\nabla f(x_i) - g_i\| \\ & & + \mathscr{D}^2 \sum\limits_{i=0}^{k-1} \tfrac{a_{i+1}^2}{A_{i+1}} \|\nabla^2 f(x_i) - H_i\|. \end{array}$$

Proof. Let us prove by induction the following inequality

$$A_k F(x) \quad \geq \quad A_k F(x_k) - B_k, \qquad x \in \operatorname{dom} \psi. \tag{4.3.4}$$

It obviously holds for $k = 0$, and for $k \geq 1$ it is equivalent to (4.3.3).

Assume that (4.3.4) holds for some $k \geq 0$, and consider the next step:

$$\begin{aligned} A_{k+1}F(x) &= a_{k+1}F(x) + A_k F(x) \\ &\overset{(4.3.4)}{\geq} \quad a_{k+1}F(x) + A_k F(x_k) - B_k \\ &\overset{(*)}{\geq} \quad A_{k+1} f\bigl(\tfrac{a_{k+1}x + A_k x_k}{A_{k+1}}\bigr) + a_{k+1}\psi(x) + A_k \psi(x_k) - B_k \qquad (4.3.5) \\ &\overset{(*)}{\geq} \quad A_{k+1}\bigl[f(x_{k+1}) + \langle \nabla f(x_{k+1}), \tfrac{a_{k+1}x + A_k x_k}{A_{k+1}} - x_{k+1} \rangle \bigr] \\ &\qquad\qquad + a_{k+1}\psi(x) + A_k\psi(x_k) - B_k, \end{aligned}$$

where $(*)$ stands for convexity of f. Now, let us denote the point

$$v_{k+1} \quad \overset{\text{def}}{=} \quad x_k + \tfrac{1}{\gamma_k}(x_{k+1} - x_k) \quad \in \quad \operatorname{dom}\psi.$$

Then, the stationary condition for the method step (4.3.2) can be written as

$$\langle g_k + H_k(x_{k+1} - x_k), x - v_{k+1} \rangle + \psi(x) \quad \geq \quad \psi(v_{k+1}), \qquad (4.3.6)$$

for all $x \in \operatorname{dom}\psi$. Therefore,

$$\begin{aligned} & A_{k+1} \langle \nabla f(x_{k+1}), \tfrac{a_{k+1}x + A_k x_k}{A_{k+1}} - x_{k+1} \rangle + a_{k+1}\psi(x) \\ &= \quad a_{k+1}\bigl[\langle \nabla f(x_{k+1}), x - v_{k+1} \rangle + \psi(x) \bigr] \\ &= \quad a_{k+1}\bigl[\langle g_k + H_k(x_{k+1} - x_k), x - v_{k+1} \rangle + \psi(x) \\ &\qquad\qquad + \langle \nabla f(x_k) - g_k, x - v_{k+1} \rangle \\ &\qquad\qquad + \langle (\nabla^2 f(x_k) - H_k)(x_{k+1} - x_k), x - v_{k+1} \rangle \\ &\qquad\qquad + \langle \nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k)(x_{k+1} - x_k), x - v_{k+1} \rangle \bigr] \end{aligned}$$

By Lipschitz continuity of the Hessian, it holds

$$\begin{array}{rcl} & & \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k)(x_{k+1} - x_k)\| \\ & \leq & \frac{L_2 \|x_{k+1} - x_k\|^2}{2} \;\; = \;\; \frac{L_2 \gamma_k^2 \|v_{k+1} - x_k\|^2}{2}. \end{array} \tag{4.3.7}$$

Hence,

$$\begin{array}{rl} & A_{k+1}\langle \nabla f(x_{k+1}), \frac{a_{k+1}x + A_k x_k}{A_{k+1}} - x_{k+1} \rangle + a_{k+1}\psi(x) \\ \overset{(4.3.6),(4.3.7)}{\geq} & a_{k+1}\big[\psi(v_{k+1}) - \|\nabla f(x_k) - g_k\| \cdot \|x - v_{k+1}\| \\ & - \gamma_k \|\nabla^2 f(x_k) - H_k\| \cdot \|v_{k+1} - x_k\| \cdot \|x - v_{k+1}\| \\ & - \frac{L_2 \gamma_k^2 \|v_{k+1} - x_k\|^2 \cdot \|x - v_{k+1}\|}{2} \big] \\ \geq & a_{k+1}\psi(v_{k+1}) - a_{k+1}\mathscr{D}\|\nabla f(x_k) - g_k\|_* \\ & - \frac{a_{k+1}^2 \mathscr{D}^2 \|\nabla^2 f(x_k) - H_k\|}{A_{k+1}} - \frac{a_{k+1}^3 L_2 \mathscr{D}^3}{A_{k+1}^2}. \end{array} \tag{4.3.8}$$

Thus, combining all together, and using convexity of ψ, we obtain

$$\begin{array}{rl} A_{k+1}F(x) \overset{(4.3.5),(4.3.8)}{\geq} & A_{k+1}f(x_{k+1}) + a_{k+1}\psi(v_{k+1}) + A_k\psi(x_k) - B_k \\ & - a_{k+1}\mathscr{D}\|\nabla f(x_k) - g_k\| - \frac{a_{k+1}^2 \mathscr{D}^2 \|\nabla^2 f(x_k) - H_k\|}{A_{k+1}} - \frac{a_{k+1}^3 L_2 \mathscr{D}^3}{A_{k+1}^2} \\ \geq & A_{k+1}F(x_{k+1}) - B_{k+1}. \end{array}$$

So, we have (4.3.4) justified for all $k \geq 0$. □

Now, let us consider the simplest estimation strategy. At iteration k, we sample uniformly and independently two subsets of indices $S_k^g, S_k^H \subseteq \{1, \ldots, M\}$. Their sizes are $m_k^g \overset{\text{def}}{=} |S_k^g|$ and $m_k^H \overset{\text{def}}{=} |S_k^H|$, which are possibly different. Then, in algorithm (4.3.2), we can use the following random estimators:

$$g_k \;=\; \frac{1}{m_k^g} \sum_{i \in S_k^g} \nabla f_i(x_k), \qquad H_k \;=\; \frac{1}{m_k^H} \sum_{i \in S_k^H} \nabla^2 f_i(x_k). \tag{4.3.9}$$

Let us present for this process a result on its global convergence.

Theorem 4.3.2. *Let* $\gamma_k := 1 - \bigl(\frac{k}{k+1}\bigr)^3 = \mathcal{O}\bigl(\frac{1}{k}\bigr)$. *Set*

$$m_k^g \;\; := \;\; 1/\gamma_k^4, \qquad m_k^H \;\; := \;\; 1/\gamma_k^2. \tag{4.3.10}$$

Then, for the iterations $\{x_k\}_{k \geq 1}$ *of algorithm* (4.3.2)*, based on estimators* (4.3.9)*, it holds*

$$\boldsymbol{E}\bigl[F(x_k)\bigr] - F^* \;\; \leq \;\; \mathcal{O}\Bigl(\tfrac{L_2\mathscr{D}^3 + L_1\mathscr{D}^2(1+\log(n)) + L_0\mathscr{D}}{k^2}\Bigr). \tag{4.3.11}$$

Proof. Let us fix iteration $k \geq 0$. For one uniform random sample $i \in \{1, \ldots, M\}$, we have

$$\boldsymbol{E}\bigl[\|\nabla f(x_k) - \nabla f_i(x_k)\|^2\bigr] \;\; = \;\; \boldsymbol{E}\bigl[\|\nabla f_i(x_k)\|^2\bigr] - \|\nabla f(x_k)\|^2 \;\; \leq \;\; L_0^2.$$

Therefore, for the random batch of size m_k^g, we obtain

$$\begin{array}{rl} & \boldsymbol{E}\bigl[\|\nabla f(x_k) - g_k\|\bigr] \\[2ex] \leq & \sqrt{\boldsymbol{E}\bigl[\|\nabla f(x_k) - g_k\|^2\bigr]} \\[2ex] = & \sqrt{\frac{1}{(m_k^g)^2}\boldsymbol{E}\bigl[\|\sum_{i \in S_k^g}(\nabla f(x_k) - \nabla f_i(x_k))\|^2\bigr]} \\[2ex] = & \sqrt{\frac{1}{(m_k^g)^2}\sum_{i \in S_k^g}\boldsymbol{E}\bigl[\|\nabla f(x_k) - \nabla f_i(x_k)\|^2\bigr]} \\[2ex] \leq & \frac{L_0}{\sqrt{m_k^g}}. \end{array} \tag{4.3.12}$$

More advanced reasoning for matrices (Matrix Bernstein Inequality; see Chapter 6 in [151]) gives

$$\begin{array}{rcl} \boldsymbol{E}\bigl[\|\nabla^2 f(x_k) - H_k\|\bigr] & \leq & L_1\Bigl(\sqrt{\frac{2\log(2n)}{m_k^H}} + \frac{2\log(2n)}{3m_k^H}\Bigr) \\[2ex] & \leq & \frac{L_1(3\sqrt{2\log(2n)} + 2\log(2n))}{3\sqrt{m_k^H}} \\[2ex] & \leq & \frac{L_1(6+7\log(2n))}{6\sqrt{m_k^H}}. \end{array} \tag{4.3.13}$$

So, using these estimates together, we have, for every $k \geq 1$

$$\begin{aligned} \boldsymbol{E}\bigl[F(x_k)\bigr] - F^* & \\ \overset{(4.3.3)}{\leq} \quad & \tfrac{1}{A_k}\Bigl(\tfrac{L_2 D^3}{2}\sum_{i=0}^{k-1}\tfrac{a_{i+1}^3}{A_{i+1}^2} + D\sum_{i=0}^{k-1} a_{i+1}\boldsymbol{E}\bigl[\|\nabla f(x_i) - g_i\|\bigr] \\ & \qquad + D^2\sum_{i=0}^{k-1}\tfrac{a_{i+1}^2}{A_{i+1}}\boldsymbol{E}\bigl[\|\nabla^2 f(x_i) - H_i\|\bigr]\Bigr) \\ \overset{(4.3.12),(4.3.13)}{\leq} \quad & \tfrac{1}{A_k}\Bigl(\tfrac{L_2 D^3}{2}\sum_{i=0}^{k-1}\tfrac{a_{i+1}^3}{A_{i+1}^2} + L_0 D\sum_{i=0}^{k-1}\tfrac{a_{i+1}}{\sqrt{m_i^g}} \\ & \qquad + \tfrac{L_1 D^2(6+7\log(2n))}{6}\sum_{i=0}^{k-1}\tfrac{a_{i+1}^2}{A_{i+1}\sqrt{m_i^H}}\Bigr) \\ \overset{(4.3.10)}{=} \quad & \tfrac{1}{A_k}\Bigl(\tfrac{L_2 D^3}{2} + L_0 D + \tfrac{L_1 D^2(6+7\log(2n))}{6}\Bigr)\sum_{i=0}^{k-1}\tfrac{a_{i+1}^3}{A_{i+1}^2}. \end{aligned}$$

Thus, for the choice $A_k := k^3$, we get

$$\boldsymbol{E}\bigl[F(x_k)\bigr] - F^* \quad \leq \quad \mathcal{O}\Bigl(\tfrac{L_2 D^3 + L_1 D^2(1+\log(n)) + L_0 D}{k^2}\Bigr). \qquad \square$$

Therefore, in order to solve our problem with ε-accuracy in expectation, $\boldsymbol{E}\bigl[F(x_K)\bigr] - F^* \leq \varepsilon$, we need to perform $K = \mathcal{O}\bigl(\frac{1}{\varepsilon^{1/2}}\bigr)$ iterations of the method. In this case, the total number of gradient and Hessian samples are $\mathcal{O}\bigl(\frac{1}{\varepsilon^{5/2}}\bigr)$ and $\mathcal{O}\bigl(\frac{1}{\varepsilon^{3/2}}\bigr)$, respectively. It is interesting that we need higher accuracy for estimating the gradients, which results in a bigger batch size.

4.3.2 Stochastic Variance-Reduced Scheme

To improve this result, we incorporate a simple *variance reduction* strategy for the gradients. This is a popular technique in stochastic convex optimization (see [141, 75, 33, 71, 2, 126, 57] and references therein). At some iterations, we recompute the full gradient. However, during the whole optimization process this happens logarithmic number of times in total.

Let us denote by $\pi(k)$ the maximal *power of two* which is less than or equal to k:

$$\pi(k) \quad \overset{\text{def}}{=} \quad 2^{\lfloor \log_2 k \rfloor}, \quad k > 0, \quad \pi(0) \overset{\text{def}}{=} 0.$$

The entire scheme looks as follows.

```
Stochastic Variance-Reduced Contracting Newton

Initialization. Choose $x_0 \in \operatorname{dom} \psi$.

Iteration $k \geq 0$.

1: Set anchor point $z_k = x_{\pi(k)}$.

2: Sample random batch $S_k \subseteq \{1, \ldots, M\}$ of size $m_k$.

3: Compute variance-reduced stochastic gradient                    (4.3.14)
       $g_k \;=\; \frac{1}{m_k} \sum_{i \in S_k} \bigl( \nabla f_i(x_k) - \nabla f_i(z_k) + \nabla f(z_k) \bigr)$.

4: Compute stochastic Hessian
       $H_k \;=\; \frac{1}{m_k} \sum_{i \in S_k} \nabla^2 f_i(x_k)$.

5: Pick up $\gamma_k \in (0, 1]$.

6: Perform the main step
       $x_{k+1} \;\in\; \mathop{\mathrm{Argmin}}_{y} \Bigl\{ \langle g_k, y - x_k \rangle + \frac{1}{2} \langle H_k(y - x_k), y - x_k \rangle$
                 $+ \gamma_k \psi(x_k + \frac{1}{\gamma_k}(y - x_k)) \Bigr\}$.
```

Note that this is just algorithm (4.3.2) with a specific choice of the random estimators g_k and H_k. The following result holds.

Theorem 4.3.3. *Let* $\gamma_k = 1 - \bigl(\frac{k}{k+1}\bigr)^3 = \mathcal{O}(\frac{1}{k})$. *Set batch size*

$$m_k \;=\; 1/\gamma_k^2. \tag{4.3.15}$$

Then, for all iterations $\{x_k\}_{k \geq 1}$ *of algorithm* (4.3.14), *we have*

$$\begin{aligned} &\boldsymbol{E}\bigl[F(x_k)\bigr] - F^* \\ &\quad\leq\; \mathcal{O}\Bigl(\tfrac{L_2 \mathscr{D}^3 + L_1 \mathscr{D}^2 (1 + \log(n)) + L_1^{1/2} \mathscr{D}(F(x_0) - F^*)}{k^2} \Bigr). \end{aligned} \tag{4.3.16}$$

Proof. Let us consider the following stochastic estimate

$$g_k^i \overset{\mathrm{def}}{=} \nabla f_i(x_k) - \nabla f_i(z_k) + \nabla f(z_k),$$

for a uniform random sample $i \in \{1, \dots, M\}$, and a current iterate $k \geq 0$. We denote by x^* the solution of our problem: $F^* = F(x^*)$, stationary condition for which is

$$\langle \nabla f(x^*), x - x^* \rangle + \psi(x) \quad \geq \quad \psi(x^*), \qquad x \in \operatorname{dom} \psi. \tag{4.3.17}$$

Then, it holds

$$\begin{aligned}
&\boldsymbol{E}\big[\|\nabla f(x_k) - g_k^i\|^2\big] \\
&\quad = \quad \boldsymbol{E}\big[\|(\nabla f(x_k) - \nabla f(x^*)) \\
&\qquad\quad + \quad (\nabla f_i(z_k) - \nabla f_i(x^*) - \nabla f(z_k) + \nabla f(x^*)) \\
&\qquad\quad + \quad (\nabla f_i(x^*) - \nabla f_i(x_k))\|^2\big] \\
&\quad \leq \quad 3\boldsymbol{E}\big[\|\nabla f(x_k) - \nabla f(x^*)\|^2\big] \\
&\qquad\quad + \quad 3\boldsymbol{E}\big[\|(\nabla f_i(z_k) - \nabla f_i(x^*)) - (\nabla f(z_k) - \nabla f(x^*))\|^2\big] \\
&\qquad\quad + \quad 3\boldsymbol{E}\big[\|\nabla f_i(x_k) - \nabla f_i(x^*)\|^2\big] \\
&\quad \leq \quad 3\Big(\boldsymbol{E}\big[\|\nabla f(x_k) - \nabla f(x^*)\|^2\big] + \boldsymbol{E}\big[\|\nabla f_i(z_k) - \nabla f_i(x^*)\|^2\big] \\
&\qquad\quad + \quad \boldsymbol{E}\big[\|\nabla f_i(x_k) - \nabla f_i(x^*)\|^2\big]\Big),
\end{aligned}$$

where we used the following simple bounds:

$$\begin{aligned}
\|a + b + c\|^2 \quad &\leq \quad 3\|a\|^2 + 3\|b\|^2 + 3\|c\|^2, \\
\boldsymbol{E}\big[\|\xi - \boldsymbol{E}[\xi]\|^2\big] \quad &\leq \quad \boldsymbol{E}\big[\|\xi\|^2\big],
\end{aligned}$$

which are valid for any $a, b, c \in \mathbb{R}^n$ and arbitrary random vector $\xi \in \mathbb{R}^n$.

Now, by Lipschitz continuity of the gradients, we have (see Theorem 2.1.5

in [117]):

$$\begin{aligned} \|\nabla f(x_k) - \nabla f(x^*)\|^2 \quad & \leq \quad 2L_1\big(f(x_k) - f(x^*) - \langle \nabla f(x^*), x_k - x^* \rangle\big) \\ & \overset{(4.3.17)}{\leq} \quad 2L_1\big(F(x_k) - F^*\big). \end{aligned}$$

The same holds for the random sample i, for arbitrary fixed $x \in \operatorname{dom} \psi$

$$\begin{aligned} & \boldsymbol{E}_i\big[\|\nabla f_i(x) - \nabla f_i(x^*)\|^2\big] \\ & \quad \leq \quad 2L_1 \boldsymbol{E}_i\big[f_i(x) - f_i(x^*) - \langle \nabla f_i(x^*), x - x^* \rangle\big] \\ & \quad = \quad 2L_1\big(f(x) - f(x^*) - \langle \nabla f(x^*), x - x^* \rangle\big) \\ & \quad \overset{(4.3.17)}{\leq} \quad 2L_1\big(F(x) - F^*\big). \end{aligned}$$

Thus, we obtain

$$\begin{aligned} & \boldsymbol{E}\big[\|\nabla f(x_k) - g_k^i\|^2\big] \\ & \leq \quad 12L_1 \boldsymbol{E}\big[F(x_k) - F^*\big] + 6L_1 \boldsymbol{E}\big[F(z_k) - F^*\big]. \end{aligned} \tag{4.3.18}$$

Consequently, for the random batch

$$g_k \quad \overset{\text{def}}{=} \quad \tfrac{1}{m_k} \textstyle\sum_{i \in S_k} g_k^i,$$

we have (compare with (4.3.12))

$$\begin{aligned} & \boldsymbol{E}\big[\|\nabla f(x_k) - g_k\|\big] \\ & \leq \quad \sqrt{\tfrac{1}{(m_k)^2} \textstyle\sum_{i \in S_k} \boldsymbol{E}\big[\|\nabla f(x_k) - g_k^i\|^2\big]} \\ & \overset{(4.3.18)}{\leq} \quad \sqrt{\tfrac{6L_1}{m_k}\big(2\boldsymbol{E}\big[F(x_k) - F^*\big] + \boldsymbol{E}\big[F(z_k) - F^*\big]\big)} \\ & \leq \quad \sqrt{\tfrac{12L_1}{m_k} \boldsymbol{E}\big[F(x_k) - F^*\big]} + \sqrt{\tfrac{6L_1}{m_k} \boldsymbol{E}\big[F(z_k) - F^*\big]}. \end{aligned} \tag{4.3.19}$$

So, using the variance reduction for the gradients, and the basic estimate for the Hessians, we have, for every $k \geq 1$

$$
\begin{aligned}
&\boldsymbol{E}\big[F(x_k)\big] - F^* \\
&\overset{(4.3.3),(4.3.19),(4.3.13)}{\leq} \quad \tfrac{1}{A_k}\Bigg(\tfrac{L_2 \mathscr{D}^3}{2} \sum_{i=0}^{k-1} \tfrac{a_{i+1}^3}{A_{i+1}^2} \\
&\qquad + \mathscr{D}\sqrt{6L_1} \sum_{i=0}^{k-1} \tfrac{a_{i+1}}{\sqrt{m_i}} \Big(\sqrt{2\boldsymbol{E}\big[F(x_i) - F^*\big]} + \sqrt{\boldsymbol{E}\big[F(z_i) - F^*\big]} \,\Big) \\
&\qquad\qquad + \tfrac{L_1 \mathscr{D}^2 (6 + 7\log(2n))}{6} \sum_{i=0}^{k-1} \tfrac{a_{i+1}^2}{A_{i+1}\sqrt{m_i}} \Bigg) \\
&\overset{(4.3.15)}{=} \quad \tfrac{1}{A_k}\Bigg(\Big[\tfrac{3L_2\mathscr{D}^3 + L_1 \mathscr{D}^2(6+7\log(2n))}{6} \Big] \sum_{i=0}^{k-1} \tfrac{a_{i+1}^3}{A_{i+1}^2} \\
&\qquad + \mathscr{D}\sqrt{6L_1} \sum_{i=0}^{k-1} \tfrac{a_{i+1}^2}{A_{i+1}} \Big(\sqrt{2\boldsymbol{E}\big[F(x_i) - F^*\big]} + \sqrt{\boldsymbol{E}\big[F(z_i) - F^*\big]} \,\Big)\Bigg).
\end{aligned}
$$

Now, let us set $A_{i+1} := (i+1)^3$, and thus $a_{i+1} := (i+1)^3 - i^3 \leq 3(i+1)^2$, so we have

$$
\begin{aligned}
\boldsymbol{E}\big[F(x_k)\big] - F^* \quad &\leq \quad \tfrac{\alpha + \beta(\sqrt{2}+1)(F(x_0) - F^*)}{k^2} \\
+ \tfrac{\beta}{k^3} \sum_{i=1}^{k-1} \Big((i+1)\big(&\sqrt{2\boldsymbol{E}\big[F(x_i) - F^*\big]} + \sqrt{\boldsymbol{E}\big[F(z_i) - F^*\big]} \big) \Big),
\end{aligned}
\tag{4.3.20}
$$

where

$$
\alpha \quad \overset{\text{def}}{=} \quad 27 \cdot \Big[\tfrac{3L_2\mathscr{D}^3 + L_1\mathscr{D}^2(6+7\log(2n))}{6} \Big], \qquad \beta \quad \overset{\text{def}}{=} \quad 9 \cdot \mathscr{D}\sqrt{6L_1}.
$$

We are going to prove by induction, for every $k \geq 1$

$$
\boldsymbol{E}\big[F(x_k)\big] - F^* \quad \leq \quad \tfrac{c}{k^2}, \tag{4.3.21}
$$

with

$$
\begin{aligned}
c \quad &\overset{\text{def}}{=} \quad \big(4\beta + \sqrt{\alpha + 3\beta(F(x_0) - F^*) + 16\beta^2} \big)^2 \\
&\leq \quad 74\beta^2 + 2\alpha + 6\beta(F(x_0) - F^*) \\
&= \quad \mathcal{O}\big(L_2\mathscr{D}^3 + L_1\mathscr{D}^2(1 + \log(n)) + L_1^{1/2}\mathscr{D}(F(x_0) - F^*) \big).
\end{aligned}
\tag{4.3.22}
$$

Hence, if (4.3.21) is true, then we essentially obtain the claim of the theorem. For $k = 1$, (4.3.21) follows directly from (4.3.20). Assume that (4.3.21) holds for all $1 \leq i \leq k$, and consider iteration $k+1$:

$$\begin{aligned}
&\boldsymbol{E}\big[F(x_{k+1})\big] - F^* \\
&\overset{(4.3.20),(4.3.21)}{\leq} \quad \tfrac{\alpha + \beta(\sqrt{2}+1)(F(x_0) - F^*)}{k^2} + \tfrac{\beta}{k^3} \sum_{i=1}^{k} \Big((i+1) \Big(\tfrac{\sqrt{2c}}{i} + \tfrac{\sqrt{c}}{\pi(i)} \Big) \Big) \\
&\overset{(*)}{\leq} \quad \tfrac{\alpha + \beta(\sqrt{2}+1)(F(x_0) - F^*)}{k^2} + \tfrac{\beta\sqrt{c}}{k^3} \sum_{i=1}^{k} \Big((i+1) \Big(\tfrac{2\sqrt{2}+4}{i+1} \Big) \Big) \\
&= \quad \tfrac{\alpha + (\sqrt{2}+1)\beta(F(x_0) - F^*) + (2\sqrt{2}+4)\beta\sqrt{c}}{k^2} \\
&\leq \quad \tfrac{\alpha + 3\beta(F(x_0) - F^*) + 8\beta\sqrt{c}}{k^2} \overset{(4.3.22)}{=} \tfrac{c}{k^2},
\end{aligned}$$

where in $(*)$ we have used two simple bounds: $i \leq 2\pi(i)$, and $i + 1 \leq 2i$, valid for all $i \geq 1$. □

It is thanks to the variance reduction that we can use the same batch size for both estimators now. To solve the problem with ε-accuracy in expectation, we need $K = \mathcal{O}\big(\frac{1}{\varepsilon^{1/2}}\big)$ iterations of the method. And the total number of gradient and Hessian samples during these iterations is $\mathcal{O}\big(\frac{1}{\varepsilon^{3/2}}\big)$.

4.3.3 Experiments

Let us demonstrate computational results for the problem of training Logistic Regression model, regularized by ℓ_2-ball constraints. Thus, the smooth part of the objective has the finite-sum representation

$$f(x) \quad := \quad \tfrac{1}{M}\sum_{i=1}^{M} f_i(x),$$

each component is $f_i(x) := \log(1 + \exp(\langle a_i, x\rangle))$. The composite part is the indicator of a Euclidean ball,

$$\psi(x) \quad := \quad \begin{cases} 0, & \|x\|_2 := \Bigl(\sum_{i=1}^n |x^{(i)}|^2\Bigr)^{1/2} \quad \leq \quad \frac{\mathscr{D}}{2}, \\ +\infty, & \text{else.} \end{cases}$$

Diameter $\mathscr{D}$ is a regularization parameter. Vectors $\{a_i : a_i \in \mathbb{R}^n\}_{i=1}^M$ are determined by the dataset[8].

We compare the basic stochastic version of our method, using estimators (4.3.9) — SNewton, the method with the variance reduction (algorithm (4.3.14)) — SVRNewton, and first-order algorithms (with constant step-size, tuned for each problem): SGD and SVRG [75].

The results are shown in Figures 4.16 – 4.19.

We see that using the variance reduction strategy significantly improves the convergence for both first-order and second-order stochastic optimization methods. Second-order schemes usually outperform first-order methods, in terms of the number of iterations, and the number of epochs. Despite the fact that the Newton step is more expensive, in many situations we see superiority of the second-order schemes in terms of the total computational time. [9]

We can conclude that the second-order methods are preferable for solving ill-conditioned problems of small and medium dimension. A significant advantage of our stochastic second-order schemes is that they are *free from any unknown parameters*, while having both theoretical guarantees of fast global convergence and good empirical performance.

[8] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[9] CPU time was evaluated on a machine with Intel Core i5 CPU, 1.6GHz; 8 GB RAM. All methods have been implemented in C++. Operation system: macOS 10.15. Compiler: Clang 12.0.0. The source code can be found at https://github.com/doikov/contracting-newton/

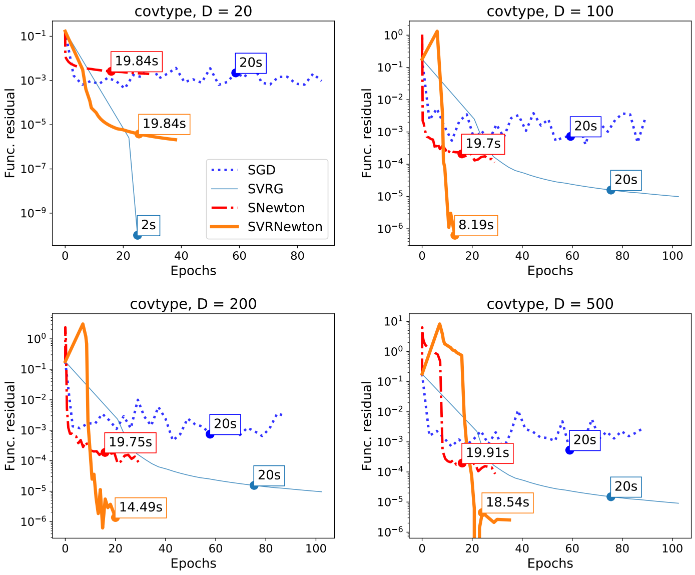


Figure 4.16: Logistic regression, *covtype* $(M = 581012, n = 54)$.

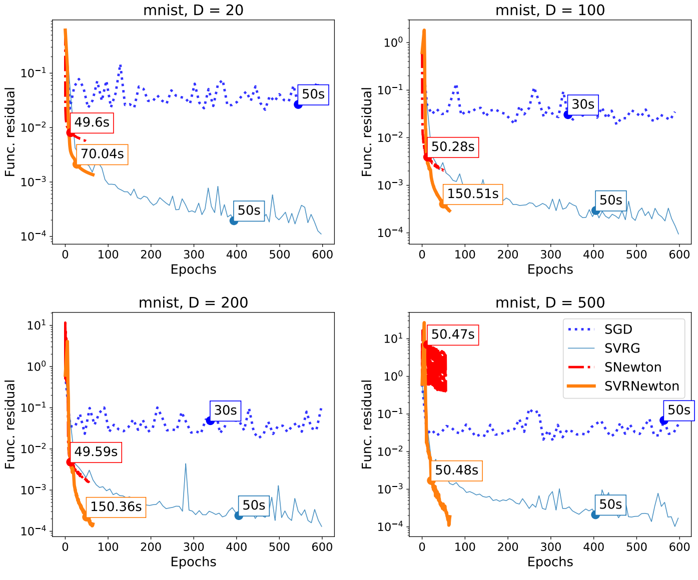


Figure 4.17: Logistic regression, *mnist* $(M = 60000, n = 780)$.

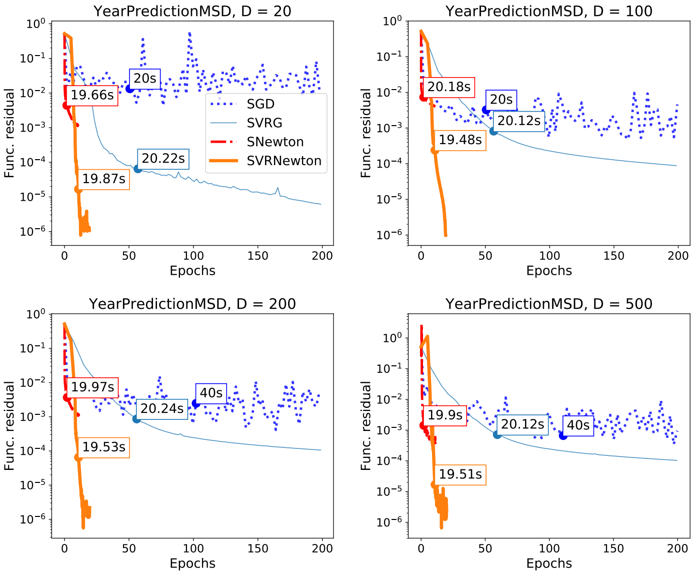


Figure 4.18: Logistic regression, *YearPredictionMSD* $(M = 463715, n = 90)$.

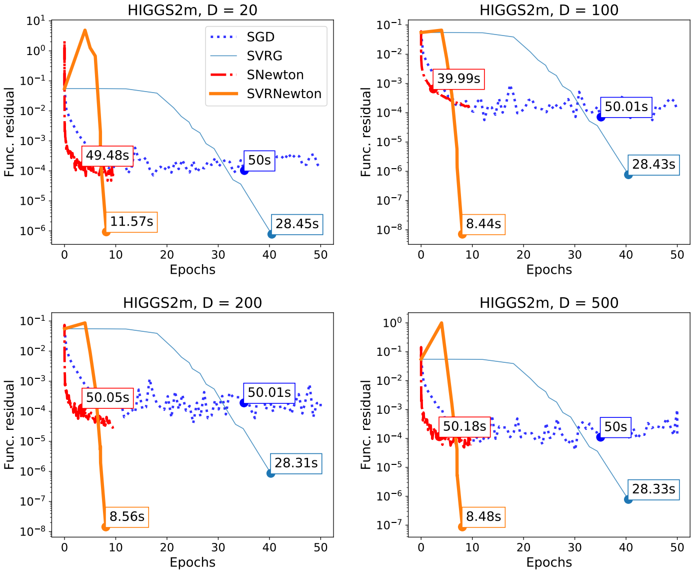


Figure 4.19: Logistic regression, *HIGGS2m* $(M = 2 \cdot 10^6, n = 28)$.

Chapter 5

Conclusions

5.1 Summary

In this thesis, we presented several new results on the theory of second-order and tensor methods, solving convex minimization problems.

Firstly, we studied the global and local performance of the prox-type algorithms that are based on the explicit regularization of Taylor's polynomial by a power of the Euclidean norm. We demonstrated that the problem classes with uniformly convex objectives are among the most favourable to the methods. They serve as an example of *nondegenerate* problems, and it is possible to introduce the *high-order condition number*, which is the main factor in the global complexity of these algorithms.

For the cubic regularization of Newton's method with adaptive estimation of the parameter, we established the fast global linear rate of convergence for uniformly convex functions with Hölder continuous Hessian. The method automatically achieves the best complexity estimate among these problem classes, without knowledge of any constants. As a consequence of this result, we proved that the global rate of the Cubic Newton is always better than that of the gradient method on the class of strongly convex functions with bounded second derivative.

For the high-order tensor methods, we established the rate of superlinear local convergence which is faster than that of the classical Newton's method. Moreover, we demonstrated that increasing the order of the method, we may extend the degree of uniform convexity for the objective. This justifies a reasonable belief that the methods of higher order have to be more powerful.

Secondly, we investigated the possibility of using the *contraction* of the objective as a natural implicit regularizer, for solving the convex problems with bounded domain. It appears that the contraction principle is used in the core of the conditional gradient methods (Frank-Wolfe algorithm). We showed that this idea can be successfully employed for constructing the second- and high-order algorithms as well.

Thus, we developed a new family of affine-invariant tensor methods equipped with the global complexity guarantees. We proved that the methods of higher order possess a faster rate of convergence, while all the constants in the complexity bounds do not depend on any particular norm or any choice of the coordinate system.

As particular cases, we obtained affine-invariant characterization of the first-order conditional gradient method, and developed new affine-invariant second-order algorithm called *Contracting Newton method.* The latter one has the same fast global rate as the cubically regularized Newton's method for convex functions with Lipschitz continuous Hessian.

Then, we demonstrated that the contracting-point and proximal-point regularization ideas are complementary to each other. Combining them together, it is possible to construct *accelerated* algorithms.

Finally, we addressed the questions of efficient implementation of the second- and high-order methods. We suggested to describe approximate solution of the subproblem in terms of the residual in function value, and proposed different strategies for choosing the inner accuracy, which are *dynamic* (changing with iterations). We proved that the inexact methods with these strategies achieve the same global convergence rate as in the error-free case.

For the Contracting Newton method, we developed the basic *stochastic version*, and the version with the *variance reduction*, which are suitable for solving modern huge-scale problems. We were able to reach the fast global rate of the full method, when the batch size for stochastic estimation of the gradient and the Hessian is gradually increasing with iterations.

Numerical experiments demonstrated that the new methods are competitive with the contemporary first-order algorithms in terms of the total computational time.

5.2 Directions for Further Research

Let us indicate some possible directions for the future research on the theory of second-order and high-order optimization methods.

Moving beyond the Lipschitz continuity. Most of the methods that we analysed in our work were built under the assumption of the boundness of certain derivatives. Note that in many cases, such a condition on the target objective naturally prescribes the method we use. At the same time, it is clear that there are several possible options here. For example, for second-order methods, we can assume that the Hessian is Hölder continuous (with respect to some fixed global norm). Or, we can bound the maximum of the variation of third derivative over the given compact convex set (see Section 3.1.2). Alternatively, one can use some local norm induced by the Hessian of the objective (which leads to the definition of *self-concordant* functions [123]). For the first-order methods, we also have a recently developed notion of *relative smoothness* [152, 7, 95]. It seems to be an interesting theoretical questions, whether we can unify some of these assumptions on the derivatives and move beyond them.

Lower complexity bounds and optimal methods. We presented a general framework of Contracting-Point methods, which provides a systematic way of constructing high-order algorithms. In the implementation of our conceptual scheme, it is enough to use just one step of the Taylor approximation. This gives the global convergence with the same rate as that of the basic high-order Proximal-Point methods [120]. It is known that the latter ones can be accelerated, when using the Euclidean norm. Additionally, it has been revealed that one step of the third-order Proximal-Point scheme might be implemented by using only the second-order oracle for the initial objective. This makes the whole picture more complicated, since we do not have any more one-to-one correspondence between the order of the method and the order of the derivative which is assumed to be Lipschitz continuous. Filling the gaps in this picture, especially related to the optimal methods, is an important direction for research.

Consequences for the theory of quasi-Newton methods. The current theory of quasi-Newton methods is mainly dedicated to their local behaviour [138]. Certainly, it would be very interesting to see any combinations of the globally convergent second-order schemes with some efficient strategies for the Hessian approximation.

Implementation of high-order tensor methods. First of all, it would be interesting to understand if we can implement third-order Contracting-Point Tensor Method with a reasonable amount of computations at each step. Note that absence of the explicit regularizer, as in the prox-type methods, makes the subproblem nonconvex and more difficult to solve.

Then, it remains to be an open problem — how to implement the Tensor Method when $p \geq 4$, by possibly taking into account the structure of convex polynomials.

Moreover, it is well known that using adaptive estimation of the Lipschitz constants helps to improve practical performance of the methods. We see it as additional challenge to provide high-order Tensor Methods with efficient line search strategies which would ensure convexity of the model.

Finally, the most efficient implementation of high-order methods must take into account modern computing architectures, which include multi-core and distributed parallel systems. We believe that some of the breakthroughs in the development of second- and high-order optimization methods can be achieved by investigating this direction.

Appendix

A Maximization of Multilinear Forms

Let us state some simple auxiliary facts about maximization of multilinear symmetric forms. For a fixed $p \geq 1$, let us consider $(p+1)$-linear symmetric form A. For a set of vectors $h_1, \ldots, h_{p+1} \in \mathbb{E}$, we have

$$A[h_1, \ldots, h_{p+1}] \quad \in \quad \mathbb{R}.$$

For two vectors $u, v \in \mathbb{E}$ and integers $i, j \geq 0$ such that $i + j = p + 1$, we use the following shorter notation:

$$A[u]^i[v]^j \quad \overset{\text{def}}{=} \quad A[\underbrace{u, \ldots, u}_{i \text{ times}}, \underbrace{v, \ldots, v}_{j \text{ times}}].$$

Let us fix arbitrary compact convex set $S \subset \mathbb{E}$. We are interested to bound the variation of A over two vectors from S, by that over the only one vector:

$$\sup_{u,v \in S} |A[u]^p[v]| \quad \leq \quad \mathcal{C}_p \sup_{h \in S} |A[h]^{p+1}|, \tag{A.1}$$

for some constant $\mathcal{C}_p$. Note, that if S is a ball in the Euclidean norm, then $\mathcal{C}_p = 1$, and the values of both supremums are equal (see Appendix 1 in [123], and Section 2.3 in [103]). In what follows, our aim is to estimate the value of $\mathcal{C}_p$ for arbitrary S. Namely, we establish the following bound.

Proposition A.1. *For any compact convex set S,* (A.1) *holds with*

$$\mathcal{C}_p \quad = \quad \tfrac{(p+1)^{p+1} + p^{p+1} + 1}{(p+1)!} \quad \leq \quad \tfrac{2(p+1)^p}{p!}. \tag{A.2}$$

Proof. For a pair of integers $n, k \geq 0$, let us denote by $\binom{n}{k}$ the binomial coefficients, given by the formula

$$\binom{n}{k} \quad \overset{\text{def}}{=} \quad \frac{n(n-1)\cdots(n-k+1)}{k!},$$

and by $\left\{ n \atop k \right\}$ we denote the Stirling numbers of the second kind. By definition, $\left\{ n \atop k \right\}$ is equal to the number of ways to partition a set of n objects into k nonempty subsets. The following important identity holds (see, for example, [59]):

$$k^n \quad = \quad k! \sum_{r=1}^{k} \frac{\left\{ n \atop r \right\}}{(k-r)!}, \qquad n \geq 1. \tag{A.3}$$

Note also, that $\left\{ n \atop n \right\} = 1$, and $\left\{ n \atop k \right\} = 0$, for $k > n$.

Now, let us fix arbitrary vectors $u, v \in S$, and consider the set of their convex combinations $h_i = \alpha_i u + (1-\alpha_i) v \in S$ for some $\alpha_i \in (0,1)$, $1 \leq i \leq p$. The binomial theorem yields the system of equations, for $1 \leq i \leq p$

$$A[h_i]^{p+1} \quad = \quad \sum_{j=0}^{p+1} \binom{p+1}{j} \alpha_i^j (1-\alpha_i)^{p+1-j} A[u]^j [v]^{p+1-j}. \tag{A.4}$$

For the choice $\alpha_i = \frac{i}{i+1}$, we have $1 - \alpha_i = \frac{1}{i+1}$, and

$$\alpha_i^j (1-\alpha_i)^{p+1-j} \quad = \quad \frac{i^j}{(i+1)^{p+1}}.$$

Therefore, introducing a vector $x \in \mathbb{R}^p$,

$$x^{(j)} \quad \equiv \quad \binom{p+1}{j} A[u]^j [v]^{p+1-j}, \qquad 1 \leq j \leq p,$$

from (A.4) we obtain the linear system $\boxed{Bx = c}$ with matrix

$$B^{(i,j)} \quad \equiv \quad i^j, \qquad 1 \leq i, j \leq p, \tag{A.5}$$

and the right hand side vector

$$\begin{aligned} c^{(i)} \quad \equiv \quad & (i+1)^{p+1} \Big(A[h_i]^{p+1} - (1-\alpha_i)^{p+1} A[v]^{p+1} \\ & \qquad - \alpha_i^{p+1} A[u]^{p+1} \Big), \qquad 1 \leq i \leq p. \end{aligned} \tag{A.6}$$

The matrix given by (A.5) looks as follows

$$B = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 2 & 2^2 & 2^3 & \dots & 2^p \\ 3 & 3^2 & 3^3 & \dots & 3^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p & p^2 & p^3 & \dots & p^p \end{pmatrix}.$$

This structure is similar to that one of the Vandermonde matrix. By the Gaussian elimination process we can build a sequence of matrices

$$B = B_1 \mapsto B_2 \mapsto \dots \mapsto B_p,$$

such that B_p is upper triangular, and the corresponding sequence of the right hand side vectors

$$c = c_1 \mapsto c_2 \mapsto \dots \mapsto c_p,$$

having the same solution x as the initial system:

$$B_t x = c_t, \qquad 1 \le t \le p.$$

Then, the last component of the solution can be easily found:

$$(p+1)A[u]^p[v] = x^{(p)} = \frac{c_p^{(p)}}{B_p^{(p,p)}}, \tag{A.7}$$

from which we may obtain the required bound for the left hand side of (A.1). Thus, we are interested to investigate the elements of B_p and c_p.

Let us prove by induction, that for every $1 \le t \le p$, it holds

$$B_t^{(i,j)} = \begin{cases} i!\left\{ {j \atop i} \right\} & \text{if } i \le t; \\ i! \sum\limits_{r=t}^{i} \frac{\left\{ {j \atop r} \right\}}{(i-r)!} & \text{otherwise.} \end{cases} \tag{A.8}$$

For $t = 1$, (A.8) follows from (A.3), and this is the base of the induction. At step t of the Gaussian elimination, we have the matrix B_t. First, we freeze its t-th row for all the following matrices:

$$B_t^{(t,j)} = B_{t+1}^{(t,j)} = B_{t+2}^{(t,j)} = \dots = B_p^{(t,j)} = t!\left\{ {j \atop t} \right\}, \qquad 1 \le j \le p.$$

Then, we subtract this row from all the rows located below, scaled by an appropriate factor, for $t < i \le p$:

$$B_{t+1}^{(i,j)} \quad = \quad B_t^{(i,j)} - \frac{B_t^{(i,t)}}{B_t^{(t,t)}} \cdot B_t^{(t,j)}, \qquad 1 \le j \le p.$$

Note, that $B_t^{(t,t)} = t!$ and $B_t^{(i,t)} = \frac{i!}{(i-t)!}$. Therefore, we obtain

$$B_{t+1}^{(i,j)} \quad = \quad i! \sum_{r=t}^{i} \frac{\left\{ {j \atop r} \right\}}{(i-r)!} \quad - \quad \frac{i! \left\{ {j \atop t} \right\}}{(i-t)!} \quad = \quad i! \sum_{r=t+1}^{i} \frac{\left\{ {j \atop r} \right\}}{(i-r)!}, \qquad 1 \le j \le p,$$

and this is (A.8) for the next step. Hence (A.8) is established by induction for all $1 \le t \le p$.

Similarly, we have the update rules for the right hand sides:

$$c_t^{(t)} \quad = \quad c_{t+1}^{(t)} \quad = \quad \dots \quad = \quad c_p^{(t)},$$

and for $t < i \le p$:

$$\begin{aligned} c_{t+1}^{(i)} \quad &= \quad c_t^{(i)} - \frac{B_t^{(i,t)}}{B_t^{t,t}} c_t^{(t)} \quad = \quad c_t^{(i)} - \binom{i}{t} c_t^{(t)} \\ &= \quad c_{t-1}^{(i)} - \binom{i}{t-1} c_{t-1}^{(t-1)} - \binom{i}{t} c_t^{(t)} \quad = \quad \dots \\ &= \quad c_1^{(i)} - \sum_{r=1}^{t} \binom{i}{r} c_r^{(r)}. \end{aligned}$$

Therefore, we have a recurrence:

$$c_p^{(i)} = c_1^{(i)} - \sum_{r=1}^{i-1} \binom{i}{r} c_p^{(r)}, \qquad 1 \le i \le p. \tag{A.9}$$

From (A.9) we obtain an explicit expression for c_p using only the initial values:

$$c_p^{(i)} \quad = \quad \sum_{j=1}^{i} (-1)^{i-j} \binom{i}{j} c_1^{(j)}, \qquad 1 \le i \le p. \tag{A.10}$$

Indeed, (A.10) follows directly from (A.9) for $i = 1$. Assume by induction that (A.10) holds for all $1 \le i \le n$, for some n. Then, for the next index, we have

$$\begin{aligned}
c_p^{(n+1)} &\overset{(A.9)}{=} c_1^{(n+1)} - \sum_{r=1}^{n} \tbinom{n+1}{r} c_p^{(r)} \\
&\overset{(A.10)}{=} c_1^{(n+1)} - \sum_{r=1}^{n} \sum_{j=1}^{r} (-1)^{r-j} \tbinom{n+1}{r} \tbinom{r}{j} c_1^{(j)} \\
&= c_1^{(n+1)} - \sum_{j=1}^{n} \left(\sum_{r=j}^{n} (-1)^{r-j} \tbinom{n+1}{r} \tbinom{r}{j} \right) c_1^{(j)} \\
&= c_1^{(n+1)} + \sum_{j=1}^{n} (-1)^{n+1-j} \tbinom{n+1}{j} c_1^{(j)},
\end{aligned}$$

where the last equation follows from simple observations:

$$\begin{aligned}
\sum_{r=j}^{n} (-1)^{r-j} \tbinom{n+1}{r} \tbinom{r}{j} &= \sum_{r=j}^{n} (-1)^{r-j} \tfrac{(n+1)!\, r!}{r!\,(n+1-r)!\, j!\,(r-j)!} \\
&= \tbinom{n+1}{j} \sum_{r=j}^{n} (-1)^{r-j} \tbinom{n+1-j}{r-j} \\
&= \tbinom{n+1}{j} \sum_{l=0}^{n-j} (-1)^{l} \tbinom{n+1-j}{l} \\
&= \tbinom{n+1}{j} \Big((1-1)^{n+1-j} - (-1)^{n+1-j} \Big) \\
&= (-1)^{n-j} \tbinom{n+1}{j}.
\end{aligned}$$

Hence (A.10) is established by induction for all $1 \leq i \leq p$.

Let us denote by $\mathcal{V}$ the supremum from the right hand side of (A.1):

$$\mathcal{V} \overset{\text{def}}{=} \sup_{h \in S} |A[h]^{p+1}|.$$

Then, in view of (A.6), we have

$$|c_1^{(j)}| = |c^{(j)}| \leq ((p+1)^{p+1} + p^{p+1} + 1)\mathcal{V}, \qquad 1 \leq j \leq p, \tag{A.11}$$

and, consequently

$$
\begin{aligned}
|A[u]^p[v]| \;&\overset{(A.7),(A.8)}{=}\; \tfrac{|c_p^{(p)}|}{(p+1)!} \;\overset{(A.10)}{=}\; \tfrac{1}{(p+1)!}\Big| \sum_{j=1}^{p} (-1)^{p-j}\tbinom{p}{j} c_1^{(j)} \Big| \\
&\overset{(A.11)}{\leq}\; \tfrac{((p+1)^{p+1}+p^{p+1}+1)\mathcal{V}}{(p+1)!}\Big| \sum_{j=1}^{p} (-1)^{p-j}\tbinom{p}{j} \Big| \;=\; \mathcal{C}_p\mathcal{V}.
\end{aligned}
$$

Since $u, v \in S$ are arbitrary vectors, we have (A.2) established. □

Let us consider the most important cases, when $p = 1$ and $p = 2$.

Corollary A.2. *For any symmetric bilinear form $A : \mathbb{E} \times \mathbb{E} \to \mathbb{R}$ and any compact convex set $S \subset \mathbb{E}$, it holds*

$$
\sup_{u,v \in S} |A[u,v]| \;\leq\; 3 \sup_{h \in S} |A[h,h]|. \tag{A.12}
$$

Corollary A.3. *For any symmetric trilinear form $A : \mathbb{E} \times \mathbb{E} \times \mathbb{E} \to \mathbb{R}$ and any compact convex set $S \subset \mathbb{E}$, it holds*

$$
\sup_{u,v \in S} |A[u,u,v]| \;\leq\; 6 \sup_{h \in S} |A[h,h,h]|. \tag{A.13}
$$

It appears that the bound in (A.12) is tight.

Example A.4. Consider the following symmetric bilinear form on two-dimensional space $\mathbb{E} = \mathbb{R}^2$:

$$
A[u,v] \;=\; u^{(1)}v^{(1)} - 2u^{(2)}v^{(2)}, \qquad u, v \in \mathbb{R}^2,
$$

and let

$$
S \;=\; \big\{ x \in \mathbb{R}^2 \;:\; x^{(1)} = 1, \; x^{(2)} \in [-1,1] \big\}.
$$

Then,

$$
\sup_{u,v \in S} |A[u,v]| \;=\; \sup_{\alpha,\beta \in [-1,1]} |1 - 2\alpha\beta| \;=\; 3.
$$

However,

$$
\sup_{h \in S} |A[h,h]| \;=\; \sup_{\alpha \in [0,1]} |1 - 2\alpha| \;=\; 1. \qquad \square
$$

If it happens that our bilinear form is positive semidefinite (e.g. it is determined by the Hessian of a convex function), the constant in (A.12) can be improved to be 1, so the both supremums are equal.

Proposition A.5. *Let symmetric bilinear form $A : \mathbb{E} \times \mathbb{E} \to \mathbb{R}$ be positive semidefinite:*

$$A[h,h] \quad \geq \quad 0, \qquad h \in \mathbb{E}.$$

Then, for *<u>any</u>* *set $S \subset \mathbb{E}$, it holds*

$$\sup_{u,v \in S} |A[u,v]| \quad = \quad \sup_{h \in S} A[h,h].$$

Proof. Indeed, by the Eigenvalue Decomposition, for some $r \geq 0$, there exists a set of linear forms $a_1, \dots, a_r \in \mathbb{E}^*$ and positive numbers $\lambda_1, \dots, \lambda_r > 0$ such that

$$A[u,v] \quad = \quad \sum_{i=1}^{r} \lambda_i \langle a_i, u \rangle \langle a_i, v \rangle, \qquad u, v \in S.$$

Therefore, using Cauchy-Bunyakovsky-Schwarz inequality, we get

$$\begin{aligned} |A[u,v]| \quad &\leq \quad \Bigl(\sum_{i=1}^{r} \lambda_i \langle a_i, u \rangle^2 \Bigr)^{1/2} \Bigl(\sum_{i=1}^{r} \lambda_i \langle a_i, v \rangle^2 \Bigr)^{1/2} \\ &= \quad \bigl(A[u,u] \bigr)^{1/2} \bigl(A[v,v] \bigr)^{1/2} \\ &\leq \quad \sup_{h \in S} A[h,h]. \end{aligned}$$

□

Bibliography

[1] Naman Agarwal and Elad Hazan. Lower bounds for higher-order convex optimization. In *Conference On Learning Theory*, pages 774–792. PMLR, 2018.

[2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

[3] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119. PMLR, 2016.

[4] Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1-2):327–360, 2019.

[5] Francis Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

[6] Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[7] Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2016.

[8] Amir Beck. *First-order methods in optimization.* SIAM, 2017.

[9] 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.

[10] Aharon Ben-Tal and Nemirovski Arkadi. Optimization iii: Convex analysis, nonlinear programming theory, nonlinear programming algorithms. *Lecture notes*, page 339, 2021.

[11] Albert A Bennett. Newton's method in general analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 2(10):592, 1916.

[12] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[13] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Philippe L Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368, 2017.

[14] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[15] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[16] Brian Bullins. Highly smooth minimization of non-smooth problems. In *Conference on Learning Theory*, pages 988–1030. PMLR, 2020.

[17] Alejandro Carderera and Sebastian Pokutta. Second-order conditional gradients. *arXiv preprint arXiv:2002.08907*, 2020.

[18] Yair Carmon and John Duchi. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.

[19] Yair Carmon and John Duchi. First-order methods for nonconvex quadratic minimization. *arXiv preprint arXiv:2003.04546*, 2020.

[20] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *Siam journal on optimization*, 20(6):2833–2852, 2010.

[21] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[22] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011.

[23] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. *Optimization Methods and Software*, 27(2):197–219, 2012.

[24] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012.

[25] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. An example of slow convergence for Newton's method on a function with globally Lipschitz continuous Hessian. Technical report, Technical report, ERGO 13-008, School of Mathematics, Edinburgh University, 2013.

[26] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Universal regularization methods: varying the power, the smoothness and the accuracy. *SIAM Journal on Optimization*, 29(1):595–615, 2019.

[27] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM Journal on Optimization*, 30(1):513–541, 2020.

[28] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, 169(2):337–375, 2018.

[29] Pafnuty L Chebyshev. Polnoe sobranie sochinenii. [in Russian]. *Izd. Akad. Nauk SSSR*, 5:7–25, 1951.

[30] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods.* SIAM, 2000.

[31] Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[32] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I Jordan, and Gert RG Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

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

[34] Olivier Devolder. *Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization.* PhD thesis, ICTEAM and CORE, Université catholique de Louvain, 2013.

[35] Olivier Devolder, François Glineur, and Yurii Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization. *SIAM Journal on Optimization*, 22(2):702–727, 2012.

[36] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[37] Nikita Doikov and Yurii Nesterov. Affine-invariant contracting-point methods for convex optimization. *CORE Discussion Papers 2020/29*, 2020.

[38] Nikita Doikov and Yurii Nesterov. Contracting proximal methods for smooth convex optimization. *SIAM Journal on Optimization*, 30(4):3146–3169, 2020.

[39] Nikita Doikov and Yurii Nesterov. Convex optimization based on global lower second-order models. *Advances in Neural Information Processing Systems*, 33, 2020.

[40] Nikita Doikov and Yurii Nesterov. Inexact tensor methods with dynamic accuracies. In *International Conference on Machine Learning*, pages 2577–2586. PMLR, 2020.

[41] Nikita Doikov and Yurii Nesterov. Local convergence of tensor methods. *Mathematical Programming*, pages 1–22, 2021.

[42] Nikita Doikov and Yurii Nesterov. Minimizing uniformly convex functions by cubic regularization of Newton method. *Journal of Optimization Theory and Applications*, pages 1–23, 2021.

[43] Nikita Doikov and Yurii Nesterov. Optimization methods for fully composite problems. *CORE Discussion Papers 2021/1*, 2021.

[44] Nikita Doikov and Peter Richtárik. Randomized block cubic Newton method. In *International Conference on Machine Learning*, pages 1289–1297, 2018.

[45] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1):451–482, 2014.

[46] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[47] Pavel Dvurechensky and Alexander Gasnikov. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.

[48] Pavel Dvurechensky, Alexander Gasnikov, Petr Ostroukhov, César A Uribe, and Anastasiya Ivanova. Near-optimal tensor methods for minimizing the gradient norm of convex function. *arXiv preprint arXiv:1912.03381*, 2019.

[49] Pavel Dvurechensky and Yurii Nesterov. Global performance guarantees of second-order methods for unconstrained convex minimization. Technical report, CORE Discussion Paper, 2018.

[50] Yury G Evtushenko and Alexey A Tretyakov. p-th order methods for solving nonlinear system. [in Russian]. *Dokl. akad. nauk*, 455(5):512–515, 2014.

[51] Henry B Fine. On Newton's method of approximation. *Proceedings of the National Academy of Sciences of the United States of America*, 2(9):546, 1916.

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

[53] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[54] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, César A Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, Jiang Qijia, Yin Tat Lee, Li Yuanzhi, and Sidford Aaron. Near optimal methods for minimizing convex functions with Lipschitz p-th derivatives. In *Conference on Learning Theory*, pages 1392–1393, 2019.

[55] Alexander Gasnikov and Yurii Nesterov. Universal method for stochastic composite optimization problems. *Computational Mathematics and Mathematical Physics*, 58(1):48–64, 2018.

[56] Stephen M Goldfeld, Richard E Quandt, and Hale F Trotter. Maximization by quadratic hill-climbing. *Econometrica: Journal of the Econometric Society*, pages 541–551, 1966.

[57] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. *arXiv preprint arXiv:1905.11261*, 2019.

[58] Nicholas IM Gould, Daniel P Robinson, and H Sue Thorne. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computation*, 2(1):21–57, 2010.

[59] Ronald L Graham, Donald E Knuth, Oren Patashnik, and Stanley Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

[60] Geovani N Grapiglia and Yu Nesterov. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM Journal on Optimization*, 27(1):478–506, 2017.

[61] Geovani N Grapiglia and Yurii Nesterov. Accelerated regularized Newton methods for minimizing composite convex functions. *SIAM Journal on Optimization*, 29(1):77–99, 2019.

[62] Geovani N Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *arXiv preprint arXiv:1907.07053*, 2019.

[63] Geovani N Grapiglia and Yurii Nesterov. Tensor methods for minimizing functions with Hölder continuous higher-order derivatives. *arXiv preprint arXiv:1904.12559*, 2019.

[64] Andreas Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[65] Osman Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

[66] Osman Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[67] Cristóbal Guzmán and Arkadi Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1–14, 2015.

[68] Filip Hanzely, Nikita Doikov, Peter Richtárik, and Yurii Nesterov. Stochastic subspace cubic Newton method. In *International Conference on Machine Learning*, pages 4027–4038. PMLR, 2020.

[69] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1):75–112, 2015.

[70] Elad Hazan. Lecture notes: Optimization for machine learning. *arXiv preprint arXiv:1909.03550*, 2019.

[71] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

[72] Anastasiya Ivanova, Dmitry Grishchenko, Alexander Gasnikov, and Egor Shulgin. Adaptive catalyst for smooth convex optimization. *arXiv preprint arXiv:1911.11271*, 2019.

[73] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435, 2013.

[74] Bo Jiang, Tianyi Lin, and Shuzhong Zhang. A unified adaptive tensor approximation scheme to accelerate composite convex optimization. *arXiv preprint arXiv:1811.02427*, 2018.

[75] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[76] Dmitry Kamzolov and Alexander Gasnikov. Near-optimal hyperfast second-order method for convex optimization and its sliding. *arXiv preprint arXiv:2002.09050*, 2020.

[77] Dmitry Kamzolov, Alexander Gasnikov, and Pavel Dvurechensky. Optimal combination of tensor optimization methods. In *International Conference on Optimization and Applications*, pages 166–183. Springer, 2020.

[78] Leonid V Kantorovich. Functional analysis and applied mathematics. [in Russian]. *Uspekhi Matematicheskikh Nauk*, 3(6):89–185, 1948.

[79] Leonid V Kantorovich. On Newton's method for functional equations. [in Russian]. In *Dokl. Akad. Nauk SSSR*, volume 59, pages 1237–1240, 1948.

[80] Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

[81] Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 217–226. SIAM, 2014.

[82] Donghwan Kim and Jeffrey A Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical programming*, 159(1):81–107, 2016.

[83] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pages 1895–1904, 2017.

[84] GM Korpelevich. The extragradient method for finding saddle points and other problems. [in Russian]. *Ekonomika i Matematicheskie Metody*, 12:747–756, 1976.

[85] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates. *arXiv preprint arXiv:1912.01597*, 2019.

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

[87] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

[88] Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

[89] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[90] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 147–156. IEEE, 2013.

[91] Claude Lemaréchal. Cauchy and the gradient method. *Doc Math Extra*, 251(254):10, 2012.

[92] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[93] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

[94] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(212):1–54, 2018.

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

[96] Aurelien Lucchi and Jonas Kohler. A stochastic tensor method for non-convex optimization. *arXiv preprint arXiv:1911.10367*, 2019.

[97] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[98] Maicon Marques Alves, Renato DC Monteiro, and Benar F Svaiter. Iteration-complexity of a Rockafellar's proximal method of multipliers for convex programming based on second-order approximations. *Optimization*, pages 1–30, 2019.

[99] Konstantin Mishchenko and Yura Malitsky. Adaptive gradient descent without descent. In *37th International Conference on Machine Learning (ICML 2020)*, number 37, 2020.

[100] Renato DC Monteiro and Benar F Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

[101] Renato DC Monteiro and Benar F Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[102] Arkadi Nemirovski. *Information-based complexity of convex programming.* Lecture Notes, 1995.

[103] Arkadi Nemirovski. *Lecture notes: Interior point polynomial time methods in convex programming.* Georgia Institute of Technology, 2004.

[104] Arkadi Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[105] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[106] Arkadi Nemirovski and David Yudin. Problem complexity and method efficiency in optimization. 1983.

[107] Yurii Nesterov. A method for solving the convex programming problem with convergence rate O(1/kˆ2). [in Russian]. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

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

[109] Yurii Nesterov. Modified Gauss–Newton scheme with worst case guarantees for global performance. *Optimisation Methods and Software*, 22(3):469–483, 2007.

[110] Yurii Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2007.

[111] Yurii Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[112] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[113] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

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

[115] Yurii Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1-2):381–404, 2015.

[116] Yurii Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 171(1-2):311–330, 2018.

[117] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[118] Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, pages 1–27, 2019.

[119] Yurii Nesterov. Inexact basic tensor methods. *CORE Discussion Papers 2019/23*, 2019.

[120] Yurii Nesterov. Inexact accelerated high-order proximal-point methods. *CORE Discussion Papers 2020/8*, 2020.

[121] Yurii Nesterov. Inexact high-order proximal-point methods with auxiliary search procedure. *CORE Discussion Papers 2020/10*, 2020.

[122] Yurii Nesterov. Superfast second-order methods for unconstrained convex optimization. *CORE Discussion Papers 2020/7*, 2020.

[123] Yurii Nesterov and Arkadi Nemirovski. *Interior-point polynomial algorithms in convex programming.* SIAM, 1994.

[124] Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton's method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[125] Yurii Nesterov and Sebastian U Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

[126] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621, 2017.

[127] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[128] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables.* SIAM, 2000.

[129] Boris T Polyak. Gradient methods for minimizing functionals. [in Russian]. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

[130] Boris T Polyak. *Introduction to optimization.* Optimization Software, 1987.

[131] Boris T Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181(3):1086–1096, 2007.

[132] R Tyrrell Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970.

[133] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[134] Anton Rodomanov and Dmitry Kropotov. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, pages 2597–2605, 2016.

[135] Anton Rodomanov and Dmitry Kropotov. A randomized coordinate descent method with volume sampling. *SIAM Journal on Optimization*, 30(3):1878–1904, 2020.

[136] Anton Rodomanov and Yurii Nesterov. Smoothness parameter of power of euclidean norm. *J. Optim. Theory Appl.*, 185(2):303–326, 2020.

[137] Anton Rodomanov and Yurii Nesterov. Greedy quasi-Newton methods with explicit superlinear convergence. *SIAM Journal on Optimization*, 31(1):785–811, 2021.

[138] Anton Rodomanov and Yurii Nesterov. New results on superlinear convergence of classical quasi-Newton methods. *Journal of Optimization Theory and Applications*, pages 1–26, 2021.

[139] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods i: globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016.

[140] Saverio Salzo and Silvia Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex analysis*, 19(4):1167–1192, 2012.

[141] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[142] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466, 2011.

[143] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

[144] Mikhail V Solodov and Benar F Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical functional analysis and optimization*, 22(7-8):1013–1035, 2001.

[145] Chaobing Song and Yi Ma. Towards unified acceleration of high-order algorithms under Hölder continuity and uniform convexity. *arXiv preprint arXiv:1906.00582*, 2019.

[146] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning.* Mit Press, 2012.

[147] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.

[148] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *SIAM Journal on Optimization*, 27(3):1283–1313, 2017.

[149] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1-2):307–345, 2017.

[150] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2899–2908, 2018.

[151] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[152] Quang Van Nguyen. Forward-backward splitting with Bregman distances. *Vietnam Journal of Mathematics*, 45(3):519–539, 2017.

[153] Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372*, 2018.

[154] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularization methods. *Journal of Machine Learning Research*, 20(134):1–47, 2019.