Tensor Methods for Nonconvex Optimization



Kate Wenqi Zhu
Supervisor: Prof. Coralia Cartis
University of Oxford

A thesis submitted for the degree of
PhD in Mathematics

Hilary 2026

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Coralia Cartis. I was first introduced to continuous optimization through her course during my Master's studies, an experience that ultimately led me onto the path of research in optimization. For this initial guidance, and for everything that followed, I am profoundly grateful.

Throughout my PhD, Professor Cartis has provided me with rigorous and comprehensive research training, as well as a remarkably far-sighted research vision. At the very beginning of my doctorate, she warned me that this research direction would be challenging, and indeed, my first year was marked by several unsuccessful attempts. At every critical juncture, however, she encouraged me to persist, while offering clear direction and thoughtful supervision. Under her guidance, I gradually came to understand what it means to have research taste: how to identify good and meaningful problems, how to focus on the right questions, and how to allocate time and effort wisely.

Beyond research, she has shown exceptional care for my academic and personal well-being. When our work required additional input or resources, she actively connected me with the right people, both within and beyond the Oxford community, helping to establish balanced and fruitful collaborations. She consistently paid attention to my mental well-being and work–life balance. I cannot imagine a better mentor, and I very much look forward to continuing our collaboration and producing further results in the future. There is a saying in Chinese: "There are many fine horses, but few who can recognize them." I consider myself extremely fortunate to have encountered such a mentor.

I am also deeply thankful to my co-authors and colleagues—Karl Welzel, Shiwei Liu, Yang Liu, Haibin Chen, Congzheng Liu, Raphael Hauser, Renjie Xu and many others—for the many long and intellectually stimulating discussions and brainstorming sessions. Through challenging each other's ideas and meaningful discussions, these collaborations broadened my perspective and exposed me to diverse ways of thinking. Also, I would like to acknowledge the financial support from CIMDA, which enabled me to continue my research in mathematics and to attend conferences across the United States, Canada, and Europe. I am especially grateful to Professor Hong Yan for his guidance and encouragement during my visits to Hong Kong.

Arriving in Oxford at the age of sixteen, I have spent my undergraduate years, two master's degrees, my PhD, and now my postdoctoral work here—nearly half of my life in this city. This town and this university have given me some of the most formative and unforgettable memories I will ever carry. As I write these acknowledgements, it is already 9 p.m.—just one among the countless late nights that have quietly accompanied my years of study. Outside my window, it is a beautiful Oxford summer evening: the blue hour settling over the ancient colleges. It feels like the curtain is slowly falling on the ending of one long journey, and the quiet beginning of another. I am sincerely grateful for Oxford's intellectual clarity and its deeply rooted academic spirit, which have strengthened my determination to pursue a life in academia. I hope, if I am fortunate, to remain here longer and to one day continue my work in this place that has shaped me so profoundly.

Finally, I thank my parents and family for introducing me to mathematics and for providing the best possible early education. I am especially grateful to my mother, whose tough love instilled in me a kind of relentless persistence—to not give up, to not walk away, and to pursue things proudly with full commitment; may not say much, but I love her deeply. I am deeply grateful to Plato, who has accompanied me through periods of depression and darkness, giving me a sense of calm and strength, and accompanying me as I rediscovered both mathematics and life. I also thank Lily (Lele)—for bringing colour into my life and transforming the way I see the world, from a glass-bead game into something vivid, real, and alive.

I am grateful as well to my readers, followers, and critics alike, all of whom have witnessed my growth. Looking ahead, the goal is not merely to be seen, but to choose how to be seen—and, through that choice, to reshape the gaze of others. I hope to set a precedent, and to fully unleash and fulfil my aspirations.

Abstract

Traditionally, first-order gradient-based techniques and second-order Newton-type methods have dominated the field of optimization. In recent years, *high-order tensor methods with regularization* for nonconvex optimization have garnered significant research interest. These methods offer superior local convergence rates and improved worst-case evaluation complexity, where complexity is measured by the number of function and derivative evaluations (oracle calls) rather than by iteration count. In particular, the well-known pth-order adaptive regularization (ARp) method, which allows higher derivatives to be incorporated into adaptive regularization frameworks, leads to algorithms with optimal global rates of convergence and local rates that are faster than Newton's method.

On each iteration, these methods require the unconstrained local minimization of a (potentially nonconvex) multivariate polynomial of degree higher than two, constructed using third-order (or higher) derivative information and regularized by an appropriate power of the change in the iterates. The most critical challenge in implementing the pth-order method (i.e., ARp), with $p \geq 3$, lies in efficiently minimizing the pth-order subproblem, which typically consists of a pth-degree multivariate Taylor polynomial combined with a $(p+1)$th-order regularization term. For polynomial subproblems of order $p \geq 3$, fundamental questions remain open, including minimizing these polynomials locally or globally and characterizing the global minimizers. This thesis establishes global optimality conditions for $p = 3$ and designs several efficient algorithms for minimizing the ARp subproblem.

The contributions of this thesis are twofold. Firstly, we develop a series of efficient optimization algorithmic frameworks that minimize these higher-order polynomial subproblems locally or globally, including *Quadratic Quartic Regularization Method (QQR)*, *the Cubic Quartic Regularization Method (CQR)*, *Diagonal Tensor Method (DTM)*, and *the Sums-of-Squares Convex Taylor Method*. These theoretical advancements and algorithms are central to making higher-order methods feasible for nonconvex objectives. Secondly, the research fills gaps by characterizing the local and global optimality of the subproblem, investigating the potential NP-hardness of the subproblem, and designing an efficient global algorithmic framework for minimizing the pth-order subproblem.

The QQR method is a second-order method for minimizing the ARp subproblem for $p = 3$ inspired by Nesterov. QQR iteratively approximates the third-order tensor term by a linear combination of quadratic and quartic terms, yielding (possibly nonconvex) local models for which global optimality conditions have been fully characterized and efficient minimization algorithms are available. The proposed QQR method exhibits a linear rate of convergence in locally strictly convex iterations and an error reduction of at least $O(\epsilon^{\frac{4}{3}})$ for locally nonconvex iterations, where ϵ denotes the accuracy of the first-order optimality of the quartic regularized polynomials. Preliminary numerical experiments indicate that two QQR variants perform competitively with state-of-the-art approaches such as the adaptive cubic regularization methods, ARC (i.e., ARp with $p = 2$), achieving either lower objective values or iteration counts.

Building on the success of the QQR algorithm, we propose another local method, the CQR algorithmic framework, for minimizing the ARp subproblem for $p = 3$, by sequentially minimizing a sequence of local quadratic models that also incorporate both simple cubic and quartic terms. The role of the cubic term is to crudely approximate local tensor information, while the quartic one provides model regularization and controls progress. We provide necessary and sufficient optimality conditions that fully characterize the global minimizers

of these cubic-quartic models. We then turn these conditions into secular equations that can be solved efficiently using root-finding procedures. The practical variants of CQR judiciously exploit local tensor information to construct cubic–quartic models. We test these variants numerically and observe them to be competitive with cubic regularization and other subproblem solvers on typical instances and even show superior performance on ill-conditioned subproblems with special structure.

We establish the global optimality condition for the ARp subproblem for $p = 3$ and identify cases where the conditions are exact. This condition generalizes existing results and recovers the optimality results for quadratic polynomials with regularization. Building on this optimality condition, we develop a Diagonal Tensor Method (DTM) framework for minimizing the quartically-regularized cubic subproblem by iteratively minimizing a sequence of local quadratic models that incorporate both a diagonal cubic term and a quartic regularization term (DTM model). When special structures are found in the tensor term (such as low rank or a diagonal), DTM can exactly solve the given problem (in one iteration). In our numerical experiments, we propose practical DTM variants that exploit local problem information for model construction, which we then show to be competitive with cubic regularization and other subproblem solvers, with superior performance on problems with special structure.

Through the investigation of iterative solvers for AR3 subproblems (such as QQR, CQR, and DTM), we face a key theoretical question,

> "*Does there exist a polynomial-time algorithm for globally minimizing a sufficiently regularized ARp subproblem ($p \geq 3$)?*"

A standard approach to certifying the global nonnegativity of a polynomial is to represent it as a sum of squares of polynomials of a prescribed degree. When such a representation is exact, the corresponding nonnegativity condition can be expressed through a semidefinite program. A foundational milestone in real algebraic geometry is Hilbert's seminal 1888 paper [95], which provides a complete classification of the degrees and dimensions for which every nonnegative polynomial admits a representation as a sum of squares of polynomials. In this thesis, we establish new SoS representability results for generally nonconvex, nonnegative, quartically regularized cubic polynomials. We develop an understanding of how the regularization strength influences SoS certification, and provide sufficient conditions under which such certificates hold. We also identify structured subclasses for which exact SoS representations are available, together with counterexamples that demonstrate the limitations of these guarantees under alternative regularization structures. These results complement Hilbert's classical classification by revealing new structure-dependent regimes of exact SoS certifiability within multivariate quartic polynomial optimization.

Lastly, we introduce an algorithmic framework that combines the sum-of-squares (SoS) convex Taylor model with adaptive regularization techniques for nonconvex smooth optimization problems. Each iteration minimizes an SoS-convex Taylor model, offering a polynomial cost per iteration. This is the first global rate analysis for an adaptive regularization algorithm with a tractable high-order subproblem in nonconvex smooth optimization, opening the way for further improvements.

Contents

1 Introduction **1**
- 1.1 High-Order Optimization Methods . . . 4
  - 1.1.1 An Early Approach in Tensor Methods . . . 5
  - 1.1.2 ARp for Nonconvex High-Order Optimization . . . 5
  - 1.1.3 A Convex High-Order Method . . . 7
  - 1.1.4 Polynomial Optimization and Global Minimization . . . 7
  - 1.1.5 Tensor Eigenvalues and Tensor Operations . . . 8
- 1.2 Complexity Theory of ARp Algorithm and Motivating Examples for High-Order Methods 9
- 1.3 Contributions and Structure of the Thesis . . . 11
  - 1.3.1 Structure of the Thesis . . . 14

2 Local Methods for the Regularized Third-order Subproblem I **16**
- 2.1 Introduction . . . 16
- 2.2 Background and Motivation . . . 17
- 2.3 The Quadratic Quartic Regularization (QQR) Method . . . 21
  - 2.3.1 QQR Variant 1: Single Adaptive Model Parameter . . . 22
  - 2.3.2 QQR Variant 2: Two Adaptive Model Parameters . . . 26
  - 2.3.3 Complexity and Global Convergence of QQR . . . 32
- 2.4 Bounded Minimizer and Lipschitz Constants . . . 38
  - 2.4.1 Proof of Lemma 2.3.3 . . . 38
  - 2.4.2 Local Lipschitz Constants . . . 38
- 2.5 Implementations of the QQR Algorithmic Framework . . . 39
  - 2.5.1 Implementation of Variant 1 . . . 39
  - 2.5.2 Implementation of Variant 2 . . . 41
  - 2.5.3 Numerical Set-up and Illustrations . . . 41
- 2.6 Numerical Results: Comparison Across Methods . . . 42
  - 2.6.1 An Illustration of Minimizing a General Objective using the AR3 algorithm . . 44

3 Local Methods for the Regularized Third-order Subproblem II **46**
- 3.1 Introduction . . . 46
- 3.2 Characterising Global Minima of the Cubic Quartic Regularization (CQR) Polynomial 48
  - 3.2.1 Necessary Optimality Conditions for the CQR Polynomial . . . 48
  - 3.2.2 Sufficient Optimality Conditions for the CQR Polynomial . . . 50
  - 3.2.3 An Algorithm for Minimizing the CQR Polynomial . . . 56
- 3.3 The CQR Algorithmic Framework for Minimizing Local Models . . . 58
  - 3.3.1 Convergence and Complexity of CQR Algorithmic Variant for Minimizing m_3 . 60
  - 3.3.2 A CQR Variant in the Presence of Separability . . . 67
- 3.4 Numerical Implementation and Preliminary Results . . . 70
  - 3.4.1 Numerical Results: Comparison Across Methods . . . 71
  - 3.4.2 CQR Performance in Challenging Scenarios . . . 72

4 Global Optimality and Local Methods for the Regularized Third-order Subproblem III **76**
4.1 Introduction . . . 76
4.2 Global Optimality of AR3 Model . . . 77
4.2.1 Necessary Global Optimality Conditions . . . 79
4.2.2 Sufficient Global Optimality Conditions . . . 79
4.2.3 Difference Between the Global Necessary and Sufficient Conditions . . . 80
4.3 Optimality Conditions in Special Cases . . . 84
4.3.1 Transforming a Low-rank Quartically Regularized Polynomial into a Diagonal Tensor Polynomial . . . 85
4.3.2 An Algorithm for Minimizing the AR3 Model . . . 87
4.4 A Diagonal Tensor Algorithmic Framework for Minimizing the AR3 Model . . . 90
4.4.1 Convergence and Complexity . . . 91
4.5 Numerical Implementation and Preliminary Results . . . 95
4.5.1 Minimizing m_3 with Diagonal, Low Rank and Full Tensor . . . 95
4.5.2 Numerical Examples with Moré–Garbow–Hillstrom test set . . . 99
4.5.3 A Low Rank Example: Packing Problem . . . 100

5 Global Optimization Techniques for Regularized Polynomial Subproblems **103**
5.1 Introduction . . . 103
5.2 Preliminaries and Background on Nonnegative Polynomial and SoS . . . 105
5.2.1 Connection to SDP and Matrix Generalizations . . . 106
5.3 Nonnegative Quartically Regularized Polynomial is SoS for Large Regularization . . . 107
5.3.1 SoS Representation of Shifted AR3 Subproblem and Conditions for the Global Minimum . . . 108
5.3.2 Bound on Global Minimizer . . . 111
5.3.3 Behaviour of $B(s^*)$ when σ large . . . 112
5.4 Classes of SoS Quartically Regularized Polynomials . . . 115
5.4.1 Univariate Quartically Regularized Cubic Mode . . . 115
5.4.2 Quartically Regularized Cubic Model with Special T . . . 117
5.4.3 Schnabel Model . . . 119
5.5 Examples and Preliminary Numerical Results . . . 122
5.5.1 Examples with Alternative Quartic Norms . . . 122
5.5.2 Cubic and Quartic Maximization over the Euclidean Ball . . . 124

6 Global Optimization Techniques for Regularized Polynomial Subproblem: Convexified Models **127**
6.1 Introduction . . . 127
6.2 The SoS-convex Taylor Model with Adaptive Regularization . . . 128
6.3 An Upper Bound on the Regularization Parameter . . . 131
6.3.1 A Uniform Upper Bound on $\bar{\sigma}_k$ . . . 133
6.3.2 A Uniform Upper Bound on σ_k^r . . . 133
6.4 Convergence and Complexity Analysis . . . 136
6.4.1 Bounding Objective Decrease in Successful Iterations . . . 137
6.4.2 Overall Complexity Bound . . . 143
6.5 Numerical Results for Chapter 5 and Chapter 6 . . . 148
6.5.1 Preliminary Numerical Results on MGH Test Set . . . 148
6.5.2 Sensor Network Localization Problem . . . 153

7 Conclusions and Future Work **156**
7.1 Future Research Plan . . . 158

A Appendix **159**
A.1 Appendix for Chapter 3 . . . 159
A.1.1 A Uniform Upper Bound for the Iterates . . . 159
A.1.2 Proof of Lemma 3.3.3 . . . 161
A.1.3 Analysis on the Additional Condition (3.31) . . . 161
A.1.4 Numerical Results for Section 3.4.1.2 . . . 163
A.1.5 More Examples of Badly Scaled Subproblems . . . 163
A.1.6 Alternative Proof for Lemma 4.2.2 . . . 163
A.2 Appendix for Chapter 4 . . . 165
A.2.1 Global Optimality Condition for (SQR) . . . 165
A.2.2 Optimality of Subproblems in the AR3 algorithm . . . 167
A.2.3 Convergence and Complexity and DTM . . . 168
A.2.4 Supplement for Numerical Results . . . 171
A.3 Appendix for Chapter 5 . . . 171
A.3.1 Supplement for the Main Proof . . . 171
A.3.2 Supplement for Classes of SoS Quartic Regularized Polynomial . . . 174
A.3.3 Supplementary Results for Perturbing $m_{\mathcal{E}}$ . . . 176
A.4 Appendix for Chapter 6 . . . 176
A.4.1 Proof of Lemma 6.3.1 . . . 176
A.4.2 A Discussion for $a > \frac{1}{2}$ . . . 177
A.4.3 Supplement for Numerical Results . . . 177

Chapter 1

Introduction

This thesis focuses on the optimization of nonconvex functions, specifically algorithms designed to identify inputs that minimize (or maximize) a given 'objective' function. Such problems are a cornerstone of computational science, engineering, and operations research, with applications spanning structural design, weather forecasting, molecular configuration, efficient resource allocation, and optimal portfolio management [150; 19; 10; 178; 116].

Dating back to 1847, gradient-based methods can be traced to the seminal work of Cauchy [58], where the method of steepest descent was first proposed. It appears that ideas closely related to modern steepest descent was already known at that time, although rigorous convergence analyses were only developed nearly a century later. These early developments laid the foundations for first-order methods, which remain the backbone of modern large-scale optimization, particularly in machine learning, due to their simplicity, scalability, and low per-iteration cost. Beyond classical steepest descent, a wide range of first-order methods have been developed to improve practical efficiency while retaining low per-iteration cost. Momentum-based schemes, such as Polyak's heavy-ball method [161] and Nesterov's accelerated gradient method [136], were introduced to speed up convergence, particularly for convex problems. In the convex setting, Nesterov's acceleration achieves the optimal first-order complexity bound and has had a profound influence on modern optimization and machine learning. Variants such as proximal gradient methods [16] further extend first-order techniques to nonsmooth and composite optimization problems, enabling efficient handling of regularization terms commonly arising in statistics and signal processing. Despite their success, first-order methods are fundamentally limited by their reliance on gradient information alone, which often results in slow convergence near stationary points and sensitivity to ill-conditioning [151].

Second-order methods aim to overcome these limitations by exploiting curvature information. Newton's method, one of the classical second-order numerical algorithms, has an even longer history. Early versions were introduced by Newton in 1669 for solving polynomial equations, and later generalized by Raphson in 1690 [165]. Its convergence properties were subsequently studied by Fine and Bennett in 1916 [73; 17], and later by Kantorovich in 1948 [99]. From an optimization perspective, Newton's method [66; 151] can be interpreted as a second-order algorithm based on a local quadratic approximation of the objective function. When initialized sufficiently close to a minimizer and under appropriate regularity assumptions, Newton's method enjoys a locally quadratic convergence rate, which is significantly faster than that of first-order methods. Despite its attractive local behaviour, Newton's method suffers from well-known global limitations. When the initial iterate is far from an optimal solution, the method may converge very slowly, stagnate, or even diverge altogether, particularly in nonconvex settings. Moreover, the requirement of solving linear systems involving the matrix of second derivatives at each iteration can be computationally prohibitive in large-scale problems. These challenges have motivated the development of more scalable variants, globalization strategies, and regularization techniques that aim to retain the fast local convergence of Newton-type methods while improving their robustness and global behavior. Quasi-Newton methods such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [33; 75] and its limited-memory variant L-BFGS [117] have become standard tools in large-scale optimization, particularly due to their ability to approximate Hessian information at low cost. These methods often achieve superlinear local convergence without explicitly forming or

factorizing the Hessian matrix.

Globalization techniques, such as *linesearch* [151] and *trust-region* methods [62] can restrict and/or perturb the local steepest descent or Newton step to decrease the objective and ensure (sufficient) progress towards the optimum on each algorithm iteration. Trust-region approaches restrict each step to lie within a region where the local model is considered reliable, adapting the region size dynamically based on model accuracy. Trust-region methods [62] construct local quadratic models whose validity is restricted to a neighborhood where the approximation is reliable. By adaptively adjusting the trust-region radius, these methods ensure global convergence while preserving fast local rates. Trust-region Newton and quasi-Newton methods are now widely regarded as robust and effective algorithms for nonlinear optimization. Efficient implementations of standard Newton-type methods, equipped with line-search or trust-region globalization strategies, are available in both commercial and open-source software packages, and are routinely used to solve nonlinear optimization problems with thousands or even hundreds of thousands of variables. Examples of state-of-the-art solvers include `GALAHAD`, `IPOPT`, `KNITRO`, `LOQO`, `PENNON`, and `SNOPT` [176; 37; 175; 106; 78].

Regularization methods, on the other hand, modify the local quadratic model by adding a regularization term—typically involving a power of the norm—to control the step size and ensure sufficient decrease. Among these, adaptive cubic regularization (ARC) methods [46], which iteratively minimize a second-order Taylor expansion with a cubic regularization term, have received considerable attention, as they achieve optimal worst-case evaluation complexity for finding first- and second-order critical points in nonconvex optimization. Related cubic regularization approaches were independently proposed by Nesterov and Polyak (2006) [146], as well as by Weiser, Deuflhard, and Erdmann (2007) [181]. These works also rediscovered an earlier, unpublished method due to Griewank (1981) [90]. Building on these results, researchers have recently extended the adaptive cubic framework to higher orders (with $p \geq 3$) [53]. Among these, adaptive cubic regularization (ARC) methods [46], which iteratively minimize a second-order Taylor expansion with a cubic regularization term, have received considerable attention, as they achieve optimal worst-case evaluation complexity (an oracle-based measure counting the number of function and derivative evaluations rather than iterations) for finding first- and second-order critical points in nonconvex optimization.

Due to the wide range of applicability of these methods, it is crucial to establish rigorous convergence guarantees across broad classes of problems and algorithmic parameter settings. Much research has been devoted to analyzing the local and global convergence properties of standard optimization algorithms, but an equally important question concerns the *rate* at which convergence occurs. Such guarantees are essential, since faster convergence rates imply that fewer iterations are required, thereby reducing computational cost and wall-clock time. This consideration is particularly important when the evaluation of objective functions and their derivatives is computationally expensive. A central question in the convergence analysis of these algorithms is therefore the following: starting from an arbitrary initial point, does the algorithm converge to a point (or set of points) at which appropriate necessary optimality conditions are satisfied, and under what assumptions? Beyond mere convergence, one seeks quantitative bounds on the efficiency of the algorithm. Given the wide class of problems under consideration, it is natural to ask, for a fixed algorithm with prescribed termination criteria, what is the worst performance that can occur over all instances within a given problem class?

For the algorithms considered in this thesis, this question is addressed by upper-bounding the number of objective and/or derivative evaluations required to reach approximate optimality. We refer to such bounds as the *worst-case function-evaluation complexity* (or oracle complexity) of an algorithm. Throughout, complexity is measured by the number of calls to a black-box oracle returning function and derivative values—that is, by oracle calls—rather than by the nominal number of iterations, so that methods of different orders are compared on a common basis. This computational model, which counts calls to a black-box oracle returning function and derivative values, is both suitably general and well adapted to nonlinear optimization, given the wide diversity of problem structures and scales encountered in practice. Fundamental contributions and foundational results in this area can be found, for example, in [134; 144; 171].

While first- and second-order methods form the backbone of classical nonlinear optimization, their practical success crucially relies on suitable globalization strategies. From a worst-case perspective, the

steepest descent methods applied to sufficiently smooth nonconvex objectives require at most $O(\epsilon_g^{-2})$ function evaluations to reach a first-order optimal point, where the gradient norm is no greater than a specified tolerance ϵ_g. This result implies a sublinear global convergence rate [144]. Moreover, [49] proved that Newton's method can, in some cases, exhibit convergence rates as slow as the steepest descent, even on sufficiently smooth functions. These results suggest that linesearch or trust region variants turn out to be essentially as inefficient as the steepest descent in the worst case. This observation, rigorously established only in recent decades [53], highlights a gap between local convergence guarantees and global worst-case complexity.

A major conceptual advance addressing this gap is the use of regularization in local models. By augmenting Taylor approximations with higher-order regularization terms, regularized Newton-type methods fundamentally alter the global behavior of the algorithm while preserving fast local convergence. A prominent example is adaptive cubic regularization (ARC), which replaces the quadratic model of Newton's method with a cubicly regularized one [45; 46] , yielding the optimal worst-case evaluation complexity $\mathcal{O}(\epsilon_g^{-\frac{3}{2}})$ for second-order methods. This idea was extended to the pth order Adaptive Regularization framework (ARp), which provides a unifying and flexible approach to globalization, complexity analysis, and practical implementation. In ARp methods, a regularization parameter is adjusted adaptively to ensure progress towards optimality over the iterations. The ARp methodology shows that higher-order information, when coupled with appropriate regularization, leads to optimal complexity bounds at $\mathcal{O}(\epsilon_g^{-\frac{p+1}{p}})$ and naturally bridges classical optimization, modern large-scale algorithms, and tensor methods [23; 53; 51; 137]. The higher-order method offers benefits, including superior local convergence rates, improved worst-case evaluation complexity, and better parallelization compared to stochastic gradient descent (SGD). The advantages make them an increasingly interesting choice when addressing nonconvex optimization challenges.

There are many frameworks and variants for high-order methods, including approaches by Schnabel, and the higher-order techniques for convex objective functions, the high-order trust region framework, its link to polynomial optimization, and tensor optimization (See Section 1.1). In this thesis, we mainly consider the ARp framework for several reasons: Firstly, the adaptive regularization term ensures sufficient function decrease in successful steps and naturally gives step-size control strategies to achieve the desired theoretical bounds. In terms of the worst-case complexity, the ARp method is optimal among pth order methods [42] and the complexity improves as the order p increases. The exceptional theoretical complexity of the ARp method serves as a strong motivation for us to develop an algorithmic implementation for this method.

The most critical challenge for higher-order methods is the efficient minimization of such a polynomial subproblem[1]. To the best of our knowledge, before my PhD research, the optimization of such nonconvex polynomials was largely unaddressed. The contribution of the thesis goal is twofold. Firstly, we develop a series of efficient optimization techniques and algorithm frameworks that minimize these higher-order polynomial subproblems either locally or globally, including the Quadratic Quartic Regularization Method (QQR), Cubic Quartic Regularization Method (CQR), and the Diagonal Tensor Method (DTM) [56; 191; 193]. Their methods are outlined in Chapter 2, Chapter 3, and Chapter 4. These theoretical advancements and algorithms are central to making higher-order methods feasible for nonconvex objectives.

Unlike the polynomial subproblem associated with the first- and second-order method, there are many open questions related to the polynomial subproblem with $p \geq 3$. The global optimality characterization and the behavior of its local minima (which differ significantly from the second-order case) are unknown in the literature. However, the tractability of a regularized polynomial minimization is at the heart of all regularized optimization methods. Yet, beyond degree 3, this has scarcely been studied. Our goal is to answer the crucial question

> "*Under what structural conditions can sufficiently regularized polynomials be globally minimized through tractable reformulations or certificates?*"

In Chapter 5 and Chapter 6, we leverage techniques such as SoS, Lasserre Hierarchy, convex relaxations, and semidefinite programming (SDP) to address this fundamental theoretical question regard-

[1]This subproblem is detailed in (ARp Subproblem), with further information provided in Section 1.1.2.

ing the characterization of the nonnegativity and global minima of these types of polynomials. In Section 1.1, we give an overview of the different classes of higher-order methods framework and a literature survey for techniques involved in solving the subproblem minimization of higher-order models. In Section 1.1.2, we give the details of the problem set-up for our thesis and introduce the higher-order algorithm that we focus on, i.e., the ARp. Section 1.2 gives the complexity analysis of the ARp framework and presented motivating examples for high-order optimization. Section 1.3 outlines with an overview of the key research questions and contributions presented in this thesis.

1.1 High-Order Optimization Methods

There are several classes of High-Order Optimization (HOO) models/methods. In this section, we provide a brief review of key second-order and higher-order methods with $p \geq 3$.

Second-Order Optimization. Newton's method is a classical numerical algorithm, with initial versions of the method considered by Newton in 1669, and in general form by Raphson in 1690 [162]. Its convergence was studied in the works of Fine and Bennett in 1916 [74; 18]. Newton's method has a locally quadratic rate, which is much faster than the rate of the gradient methods. However, [49] proved that Newton's method applied to sufficiently smooth nonconvex objectives requires at most $O(\epsilon_g^{-2})$ function evaluations to reach a first-order optimal point, where the gradient norm is no greater than a specified tolerance ϵ_g. This is as slow as the steepest descent, even on sufficiently smooth functions. A popular approach, which is often called the damped Newton method, first published in 1944 [114], performs a line search for the Newton direction. Powell's seminal work in 1970 [163] laid the groundwork for the trust-region framework. Key advancements include Moré and Sorensen's (1983) efficient algorithms for computing trust-region steps [130], while Conn, Gould, and Toint's (2000) expanded the methodology to constrained optimization and nonlinear programming [62]. The global optimality of the trust region subproblem was fully characterized by [150]. Yuan's (2000) review highlighted the theoretical and practical advancements of trust-region algorithms, emphasizing their robustness and versatility [188].

Higher Order Method: Let $p \geq 3$. Consider the optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{P}$$

where f is p-times continuously differentiable and f is bounded below by f_{low}. The pth-order Taylor expansion of $f(x+s)$ at x is

$$T_p(x,s) := f(x) + \sum_{j=1}^{p} \frac{1}{j!} \nabla_x^j f(x)[s]^j, \tag{1.1}$$

where $\nabla_x^j f(x) \in \mathbb{R}^{n^j}$ is a jth-order tensor and $\nabla^j f(x)[s]^j$ is the jth-order derivative of f at x along $s \in \mathbb{R}^n$. 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 multivariate polynomials, which are difficult to store, handle, and minimize. For $p = 3$, we get a multivariate polynomial of degree three which is generally nonconvex, and its optimization is an open question.

Remark 1.1.1 (Difference between second- and third-order methods)**.** *The distinction between second- and third-order methods is less sharp than the order count suggests. Although the AR3 model m_3 involves the third-order tensor $T = \nabla^3 f(x_k)$, much of the algorithmic benefit of third-order information can be realized without ever forming or factorizing T as a full tensor. The cubic term $T[s]^3$ enters the model only as a scalar contribution along the step s, and the methods developed in this thesis exploit precisely this fact: QQR (Chapter 2) replaces $T[s]^3$ by a linear combination of the quadratic term $s^T H s$ and the quartic regularization $\|s\|^4$; CQR (Chapter 3) replaces it by a single radial cubic term $\beta\|s\|^3$; and DTM (Chapter 4) by a diagonal cubic. In each case the resulting subproblem is solved using only second-order (matrix–vector) machinery—secular equations and Cholesky factorizations—so that third-order curvature is captured through essentially second-order computations.*

1.1.1 An Early Approach in Tensor Methods

There are several attempts in the literature for the efficient implementation of the higher-order method. One of the first attempts goes back to 1980s by Schnabel [170], which extends classical optimization techniques to incorporate higher-order derivative information. These methods are particularly suited for unconstrained optimization problems, where traditional second-order methods like Newton's can face limitations, especially when the Hessian is singular or poorly conditioned.

Schnabel's contributions began with tensor methods for nonlinear equations, as detailed in his collaboration with Frank (1984) [170], where the authors introduced a second-order tensor model for solving nonlinear systems. This work was extended in Schnabel and Chow (1991), which explored the use of fourth-order tensor models for unconstrained optimization [169]. These methods generalized the standard quadratic model by incorporating low-rank third- and fourth-order terms, creating a framework that interpolates function and gradient information from previous $\tilde{r}$ iterates. Schnabel, Robert, and Frank [170] showed that the minimization of the low-rank high-order polynomial problem can be reduced to the minimization of a fourth-degree polynomial with $\tilde{r}$ variables, plus a quadratic equation with $n - \tilde{r}$ unknowns. The work showed that the costs of forming, storing, and solving this tensor model are not significantly more than these costs for a standard method based on a quadratic Taylor series model. On all the test sets, the tensor method solves considerably more problems than a comparable standard method. However, note that this setup is designed involving a rank $\tilde{r}$ approximation of the tensor (usually $\tilde{r} \leq n^{1/3}$ in theory and $\tilde{r} = 1$ in practice) and the framework might not approximate the objective function when the tensor is full rank or high rank. Several globalization techniques and linesearch strategies were also later studied, for instance, trust region approach [31] and, later, curvilinear linesearch [14]. But to the best of our knowledge, no regularization technique was used under this framework. Later, Software or algorithms have been developed for small and medium-sized problems (up to 100 unknowns) or for sparse tensors [30; 31]. Tensor-GMRES Method and Tensor-Krylov methods are introduced in [72; 11] to address Large-Scale spare Systems of Nonlinear Equations under the Schnabel framework.

1.1.2 ARp for Nonconvex High-Order Optimization

In this section, we describe the adaptive regularization framework algorithm that uses Taylor models of the objective of order p and give a sketch of proof of the complexity of the algorithmic framework. Since this superior complexity is the primary motivation for our higher-order method, and many algorithmic frameworks employ the adaptive regularization approach and utilize techniques inspired by complexity analysis, we believe it is crucial to provide a literature review and a sketch of the proof for the complexity analysis. The descriptions in this section are based on [53].

We start with a formal statement of our target optimization problem and specify the basic notation necessary for all thesis statements. Let $p \geq 2$. Consider the optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$), meaning that

- f is p-times continuously differentiable,
- f is bounded below by f_{low},
- the pth derivative of f at x, the pth-order tensor

$$\nabla_x^p f(x) = \left[\frac{\partial^p f(x)}{\partial x_{i_1} \dots \partial x_{i_p}} \right]_{i_j \in \{1,\dots,n\}, j=1,\dots,p}$$

is globally Lipschitz continuous, that is, there exists a constant $L_p \geq 0$ such that, for all $x, y \in \mathbb{R}^n$,

$$\|\nabla_x^p f(x) - \nabla_x^p f(y)\| \leq (p-1)! L_p \|x - y\| \tag{1.2}$$

where $\|\cdot\|$ when applied to tensor (or matrix) is the tensor (or matrix) norm recursively induced by the standard Euclidean norm on the space of pth-order tensors, which is given by

$$\|\nabla_x^p f(x)\| := \max_{\|v_1\|=\ldots=\|v_p\|=1} |\nabla_x^p f(x)[v_1,\ldots,v_p]|,$$

where $T[v_1,\ldots,v_p]$ stands for the tensor of order $p-j \geq 0$ resulting from the application of the pth-order tensor T to the vectors $v_1,\ldots,v_j$. In this thesis, we use both $s^T Hs$ and $H[s]^2$ to represent matrix-vector multiplication. Note that the global Lipschitz-continuity assumption may be relaxed in many cases to Lipschitz continuity on a suitable level set containing the iterates. Further discussion is provided in Section 2.4.

To describe our algorithm and to ensure the method converges globally from an arbitrary starting point to first/second-order critical points, we add an adaptive $(p+1)$-power regularization term to T_p. This gives us the pth-order regularized Taylor model[2],

$$m_p(x,s) = T_p(x,s) + \frac{\sigma}{p+1}\|s\|^{p+1} \qquad \text{(AR}p\text{ Subproblem)}$$

where $\sigma > 0$ is adjusted adaptively to ensure progress toward optimality over the iterations. The gradient and Hessian are

$$\nabla_s m_p(x,s) = \nabla_s T_p(x,s) + \sigma\|s\|^{p-1}s, \tag{1.3}$$

$$\nabla_s^2 m_p(x,s) = \nabla_s^2 T_p(x,s) + \sigma\left[(p-1)\|s\|^{p-3}ss^\top + \|s\|^{p-1}I_n\right]. \tag{1.4}$$

The case of $p=1$ gives the steepest descent model and the case of $p=2$ gives a Newton-like model. For $p=3$, the subproblem is a possibly nonconvex, quartically-regularized multivariate polynomial,

$$m_3(x_k,s) = f(x_k) + \nabla_x f(x_k)^T s + \frac{1}{2}s^T\nabla_x^2 f(x_k)s + \frac{1}{6}\nabla_x^3 f(x_k)[s]^3 + \frac{\sigma_k}{4}\|s\|^4. \qquad \text{(AR3 Model)}$$

in this thesis, we focus on the case of $p \geq 3$. This construction corresponds to the well-known adaptive regularization algorithmic framework ARp [23; 51; 53]. Let the tolerance for the first-criticality be defined as ϵ_g and we say that the first-criticality is satisfied if

$$\|\nabla_x f(x)\| \leq \epsilon_g.$$

Similarly, for the model (ARp Subproblem), we consider the tolerance and the first-order criticality as

$$\|\nabla m_p(x,s)\| \leq \epsilon.$$

where ∇ denotes derivative with respect to s. The ARp algorithm is detailed in Algorithm 1.1.1. We will approach the problem using a derivative-based method. Namely, we find $m_p(x,s)$ that approximates $f(x+s)$ at $x=x_k$. We then update x_k iteratively by $s_k := \text{argmin}_{s\in\mathbb{R}^n} m_p(x_k,s)$ and $x_{k+1} := x_k + s_k$ until we reach the approximate local minimum of f, i.e., $\|\nabla_x f(x_k)\| \leq \epsilon_g$.

Algorithm 1.1.1: Adaptive pth-order Regularization Method: ARp [53]

Step 0: Initialization. An initial point x_0 and an initial regularization parameter $\hat{\sigma}_0 > 0$ are given, $\theta \in (0,1)$ as well as accuracy levels $1 \leq \epsilon \leq \epsilon_g \ll 1$. The constants η_1, η_2, γ_1, γ_2, γ_3 and $\sigma_{\min}$ are also given and satisfy $\sigma_{\min} \in (0,\hat{\sigma}_0]$, $0 \leq \eta_1 \leq \eta_2 < 1$, and $0 < \gamma_1 < 1 < \gamma_2 < \gamma_3$. Compute $f(x_0)$ and set $k=0$.

Step 1: Test for termination. If $\|\nabla_x f(x_k)\| \leq \epsilon_g$ terminate with the approximate solution $x_\epsilon = x_k$. Otherwise, compute derivatives of f from order 2 to p at x_k.

Step 2: Step calculation. Compute the step s_k by approximately minimizing the model $m_p(x,s)$ concerning s in the sense that the conditions $\|\nabla m_p(x,s)\| \leq \epsilon$.

Step 3: Acceptance of the trial point. Compute $f(x_k+s_k)$ and define $\rho_k = \frac{f(x_k)-f(x_k+s_k)}{f(x_k)-T_p(x_k,s_k)}$.

Step 4: Regularization parameter update.

Update x_k: $x_{k+1} := x_k + s_k$ if $\rho_k > \eta_1$, or $x_{k+1} := x_k$ otherwise.

Update σ_k: $\sigma_{k+1} = \max\{\gamma_1\sigma_k, \sigma_{\min}\}$ when $\rho_k \geq \eta_2$;
$\sigma_{k+1} = \sigma_k$ when $\eta_1 \leq \rho_k < \eta_2$;
$\sigma_{k+1} = \gamma_2\sigma_k$ when $\rho_k < \eta_1$.

[2]Unless otherwise stated, $\|\cdot\|$ denotes the Euclidean norm in this transfer thesis.

1.1.3 A Convex High-Order Method

Nesterov has made significant contributions to the development of high-order optimization methods for convex problems, particularly emphasizing their theoretical guarantees and computational performance [139; 137; 142].

In his 2018 work, Nesterov introduced methods based on third-order and even higher-order derivatives, proving that these methods could achieve convergence rates of $O(\frac{1}{k^3})$ for third-order schemes in convex problems [144] where k is the number of iterations. This work provided a theoretical foundation for high-order optimization, showing that higher-order methods could systematically outperform their lower-order counterparts in terms of iteration complexity, particularly for problems with smooth and structured objective functions.

One of the most important pieces of work is the work of Nesterov in 2019 [137]. By adding a $p+1$ power of regularization with sufficient large coefficients (dependent on the pth order Lipschitz constant) to the pth-order Taylor expansion of $f(x_k+s)$, we obtain a global convex model that overestimates $f(x_k+s)$. Hence, by minimizing the convex model, it became possible to speak about the efficient and iterative implementation of the tensor methods. Under the convexity assumption, the additional relative smoothness properties of the third-order tensor models allow the use of Bregman gradient methods to approximately minimize the model [137]. The paper also provides an accelerated implementable version of the third-order method for convex objective functions with the rate $O(\frac{1}{k^4})$ where k is the number of iterations. The global complexity bounds of the basic and accelerated tensor methods were studied by [137; 77; 15]. The local convergence is studied in [67] and recently in [184], where the papers justify local superlinear convergence under the assumption of uniform convexity of the smooth component, having Lipschitz-continuous high-order derivative.

Following that, several methods for minimizing the convex Taylor model with regularization are proposed, including second-order methods utilizing Bregman gradient methods under relative smoothness properties [137], inexact basic tensor methods [140], inexact methods with dynamic accuracy [68], difference approximation of the third-order tensor term [139], and high-order proximal-point operators [141; 138]. Tensor methods for finding approximate stationary points of convex functions are given in [86]. [84] extended high-order methods to composite convex optimization, addressing challenges in separable structures and non-Euclidean settings. The application of high-order methods for the optimization of a smooth approximation of convex nonsmooth functions was considered by Bullins, 2020 [35]. More recently, Nesterov gave a linearly convergent second-order method for minimizing convex quartically-regularised cubic polynomials [142]. At each iteration, Nesterov approximates the third-order tensor term by a linear combination of second-order derivatives and fourth-order regularization. Universal tensor methods, which can automatically adapt to the Hölder parameters of the objective, were developed by Grapiglia and Nesterov in 2019 [85].

Remark 1.1.2 (Uniform convexity and regularized Newton-type methods)**.** *For Newton-type methods, the appropriate structural assumption is instead* uniform convexity*. Recall that f is uniformly convex of degree $q \geq 2$ with constant $\mu_q > 0$ if $f(y) \geq f(x) + \langle \nabla f(x), y-x\rangle + \frac{\mu_q}{q}\|y-x\|^q$ for all $x, y \in \mathbb{R}^n$, the case $q=2$ recovering strong convexity. Uniform convexity of degree $q=3$ guarantees cubic growth of the objective around its minimizer, which is exactly the growth matched by cubic regularization; this is why uniform (rather than strong) convexity is the natural condition under which cubic-regularization / regularized Newton methods attain fast global and local rates.*

1.1.4 Polynomial Optimization and Global Minimization

Since the high-order Taylor approximation is a pth degree polynomial with certain regularization terms, its optimization is closely related to the polynomial optimization literature. The integration of higher-order methods and polynomial optimization offers an interesting avenue for (globally) minimizing the high-order Taylor subproblem.

Polynomial optimization involves the minimization or maximization of polynomial functions over a domain, often constrained by polynomial inequalities. The problem is inherently nonconvex. Seminal contributions in this area include Lasserre's (2001) hierarchy, which uses semidefinite programming

(SDP) relaxations to approximate global solutions [108]. The Lasserre hierarchy constructs a series of increasingly tight SDP relaxations, ultimately converging to the global minimum under mild conditions. Parrilo's (2003) work on SoS programming provides a complementary approach to addressing polynomial inequalities by expressing nonnegative polynomials as SoS [157]. It is well known that while a finite SoS relaxation scheme can theoretically achieve global optimality, the size of the resulting SDPs in the hierarchy grows with the problem dimension, which imposes limitations on addressing large-scale problems. Specifically, for each relaxation involving n variables and $2d$ degrees, the dimension of the semidefinite program is given by $\binom{n+2d}{2d}$. Therefore, many modified SoS optimizations are developed that have sparsity or special structure to address large-scale problems, for instance, (Scaled) Diagonally Dominant SoS Optimization (DSoS, SDSoS), Bounded Degree SoS (BSoS) [4; 113; 177; 182; 189].

Recently, Ahmadi, Chaudhry, and Zhang [1] present generalizations of Newton's method that incorporate derivatives of an arbitrary order p but maintain a polynomial dependence on dimension in their cost per iteration. At each step, the pth-order method uses semidefinite programming to construct and minimize a SoS-convex approximation to the pth-order Taylor expansion of the function we wish to minimize. Under additional assumptions, the method has a polynomial cost per iteration, which is globally convergent and has local convergence of order p.

If the shifted high-order regularized Taylor model can be expressed in SoS form or SoS convex, there is a tractable SDP reformulation for the global minimization problem. However, when and how polynomials can be expressed in SoS form remain open questions in the literature. It has been proven by Nesterov that finding the global minimum of a cubic polynomial over a unit ball is NP-hard [135]. Also, it has been proven that finding the global minimum of general even-degree polynomials with a degree at least 4 is also NP-hard [1; 8; 133], for instance, the sensor localization problem [120]. Moreover, determining the convexity of polynomials even for polynomials of degree 4 [5]. However, whether minimizing the high-order Taylor model with an adaptive regularization term is NP-hard is still unknown. Chapter 5 will be devoted to further investigations into this topic.

There are other global optimization techniques for polynomial functions, especially quartic polynomials. One such approach, proposed by Qi [164], focuses on normal quartic polynomials with a positive definite fourth-degree coefficient tensor. The authors develop a global descent algorithm to locate the global minimum of normal quartic polynomials in the case where $n = 2$. Another strategy is to utilize branch-and-bound algorithms, which involve recursively partitioning the feasible region and constructing nonconvex quadratic or cubic lower bounds. These algorithms, such as those presented in [43; 76; 96; 107; 147], aim to identify the global minimum through bounding procedures. In recent work [9; 36], a trajectory-type method is proposed for global optimization of multivariate quartic normal polynomials. This approach convexifies the quartic polynomial and generates a path originating from the minimizer of the convexified function. While convergence is proven for univariate quartic normal polynomials, the results for multivariate cases are only supported by numerical experiments. Despite the advancements in these methods, they all face challenges associated with the curse of dimensionality. The computational complexity can grow exponentially with the polynomial's number of variables, rendering certain methods impractical for large-scale problems.

1.1.5 Tensor Eigenvalues and Tensor Operations

Lastly, to end the section, we would like to mention that the high-order method is also closely related to tensor eigenvalues and tensor operations. It is well known that homogeneous polynomials can be reformulated as high-order symmetric tensors applied to a vector of variables. As such, the minimization of a higher-order Taylor polynomial can be reformulated as an extreme eigenvalue problem of tensor.

Kolda's methods and Proximal Alternating Minimization (PAM) approaches have significantly advanced tensor optimization and decomposition techniques, with notable connections to eigenvalue problems and higher-order optimization. Kolda and Bader (2009) [103] provided a comprehensive survey on tensor decompositions, including CP and Tucker decompositions, which have widespread applications in data compression, machine learning, and multilinear algebra. These methods often rely on solving eigenvalue-like problems within the tensor structure, where the spectral properties of

tensors play a critical role in optimization. For example, CP decomposition usually involves minimizing higher-order tensor approximations, a process that is closely tied to finding the dominant eigenvalues of tensors. Proximal Alternating Minimization (PAM) methods, such as those discussed by Xu and Yin (2013) [186], provide an effective framework for optimizing nonconvex objectives with separable structures. The incorporation of proximal terms ensures convergence to stationary points, which is particularly valuable in higher-order tensor optimization problems involving eigenvalue constraints or regularization terms. High-order optimization frameworks also benefit from these methods by leveraging the tensor eigenvalue structure to construct efficient algorithms for nonconvex landscapes. The integration of Kolda's tensor decomposition techniques with PAM approaches demonstrates the synergy between eigenvalue analysis, tensor optimization, and higher-order methods, enabling robust solutions for large-scale, structured problems.

We consider the solution of nonlinear least-squares problems. Such problems have traditionally been solved using a Gauss-Newton or Newton approximation. Gould, Rees, and Scott introduced the tensor-Newton approximation [82]. Convergence and evaluation-complexity analysis of a regularized tensor-Newton method for solving nonlinear least-squares problems are detailed in [81]. The local convergence rate of ARp variants is discussed in [67] which matches the local convergence result for the higher order Newton method in [1]. Comprehensive analysis and literature survey on methods utilizing third derivatives and beyond are given in [44, Ch 4].

1.2 Complexity Theory of ARp Algorithm and Motivating Examples for High-Order Methods

Under Lipschitz continuity assumptions on $\nabla^p f(x)$, ARp methods require no more than $\mathcal{O}\left(\epsilon_g^{-\frac{p+1}{p}}\right)$ evaluations of f and its derivatives to compute a 1st order minimizer to an accuracy of ϵ_g [53; 51; 23]. Firstly, the Lipschitz continuous assumption (1.2) gives the following crucial bounds.

Lemma 1.2.1. *[53, Lemma 2.1] Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$, and $T_p(x,s)$ be the Taylor approximation of $f(x+s)$ about x. Then, for all $x, s \in \mathbb{R}^n$,*

$$f(x+s) \leq T_p(x,s) + \frac{L_p}{p}\|s\|^{p+1} \tag{1.5}$$

$$\left\|\nabla_x f(x+s) - \nabla T_p(x,s)\right\| \leq L_p\|s\|^p. \tag{1.6}$$

As it is typical for a complexity analysis of (regularization and other) methods, we proceed by showing lower bounds on the Taylor model decrease and the length of the step at each iteration. The next three lemmas are based on [53].

Lemma 1.2.2. *[53, Lemma 3.1] (Lower bounds on the Taylor model decrease) The mechanism of Algorithm (1.1.1) guarantees that, for all $k \geq 0$,*

$$T_p(x_k, 0) - T_p(x_k, s_k) \geq \frac{\sigma_k}{p+1}\|s_k\|^{p+1}.$$

We next deduce a simple upper bound on the regularization parameter σ_k.

Lemma 1.2.3. *[53, Lemma 3.2] Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$. Then, for all $k \geq 0$,*

$$\sigma_k \leq \hat{\sigma}_{\max} := \max\left\{\hat{\sigma}_0, \frac{\gamma_3 L_p(p+1)}{p(1-\eta_2)}\right\}.$$

Our next move, very much in line of the theory proposed in [20; 53], is to show that the step cannot be arbitrarily small compared with the gradient of the objective function at the trial point $x_k + s_k$.

Lemma 1.2.4. *[53, Lemma 3.3] Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$. Then, for all $k \geq 0$,*

$$\|s_k\| \geq \left(\frac{\|\nabla f(x_k + s_k)\|}{2(L_p + \sigma_k)}\right)^{1/p}. \tag{1.7}$$

Note that the Lemma uses and the first-order criticality $\|\nabla m_p(x,s)\| \leq \epsilon.$ as the tolerance. The same order of complexity result can be derived when considering approximate minimization with a step-termination condition, such as $\|\nabla m_p(x_k,s_k)\| \leq \theta\|s_k\|^p$ for some $\theta \in (0,1)$, the bound in (1.7) becomes $\|s_k\| \geq \left(\frac{\|\nabla f(x_k+s_k)\|}{(L_p+\theta+\sigma_k)}\right)^{1/p}$.

We now bound the number of unsuccessful iterations as a function of the number of successful ones.

Lemma 1.2.5. *[53, Lemma 3.5] The mechanism of Algorithm 2.1 guarantees that if*

$$\sigma_k \leq \hat{\sigma}_{\max}$$

for some $\hat{\sigma}_{\max}$*, then*

$$k+1 \leq |\mathcal{S}_k|\left(1+\frac{\log\gamma_1}{\log\gamma_2}\right)+\frac{1}{\log\gamma_2}\log\left(\frac{\hat{\sigma}_{\max}}{\hat{\sigma}_0}\right).$$

Using all the above results, we give the main evaluation complexity result.

Theorem 1.2.1. *[53, Theorem 3.6] Let* $p \geq 2$*,* $f \in C^{p,1}(\mathbb{R}^n)$*. Then, ARp requires at most*

$$\left\lceil \hat{\kappa}_{1,2}(f(x_0)-f_{\text{low}})\epsilon_g^{-\frac{p+1}{p}}+\hat{\kappa}_{1,2}\right\rceil \tag{1.8}$$

functions'/derivatives' evaluation counts to reach the approximate local minimum with first and second-order criticality as $\|\nabla_x f(x_k)\| \leq \epsilon_g$*. More details on the proof can be found in [23, 51, 44, Ch 4].*

Theorem 1.2.1 shows that to achieve the same criticality conditions, the evaluation complexity bound improves as we increase the order p. Moreover, [42] proves that ARp has the optimal complexity for the pth-order methods. For instance, the AR3 model exhibits better worst-case performance than first/second derivative methods, with evaluation complexity of $\mathcal{O}\left(\epsilon_g^{-4/3}\right)$. This superior complexity result motivates us to develop an algorithmic implementation for AR3. Note that these complexities exclude the cost of minimizing sub-problem (ARp Subproblem).

Remark 1.2.1 (When high-order methods are most beneficial)**.** *The advantage of higher-order regularization is dictated by the relative magnitude of the high-order Lipschitz constant rather than by the order p in isolation. The model error of the pth-order Taylor expansion is controlled by* $L_p\|s\|^{p+1}$*, so a pth-order method is most beneficial when* L_p *is small relative to the gradient or Hessian Lipschitz constant. The extreme case is an objective that is itself a polynomial of degree* $p+1$*: its* $(p+1)$*th derivative is constant, the* $(p+1)$*th-order behaviour is captured exactly, and a pth-order regularized model represents the objective with no approximation error. For the third-order methods of this thesis* $(p=3)$*, this corresponds to a quartic objective.*

Remark 1.2.2. *We remark here that, as pointed out in [44, Sec 4.1.3], a trust-region method, involving a pth degree Taylor-series model of the form (1.1) does not improve on the worst-case complexity bound for finding first-order points. A p-times continuously differentiable function with Lipschitz continuous p-th derivative can be built on which a trust-region algorithm using p exact derivatives would still require* $O(\epsilon_g^2)$ *evaluations to find an iterate* x_k *for which* $\|\nabla f(x_k)\| \leq \epsilon_g$*.*

There are some existing algorithmic and numerical proposals involving related AR3 sub-problems. The ARp algorithm with $p=3$ was investigated by Birgin et al. [22] who compared the numerical performance of AR2 and AR3 in unconstrained optimization. Solvers for non-convex AR3 subproblems have been proposed in [57; 191] by generalizing a method of Nesterov suggested in [142]. However, for $p \geq 3$, both the minimization of the subproblems eq. (ARp Subproblem) and the efficient adaptive choice of the regularization parameter remain underexplored areas.

Illustrative Examples: We present two motivating examples that illustrate the superior numerical performance of third-order methods over first- and second-order methods. These examples are

drawn from the numerical experiments reported in [54] and originate from work by Liu and Welzel. The total number of iterations for the 'slalom' model and the 'hairpin turn' model using AR1, AR2, and AR3 were 29529, 935, and 18 with corresponding running times of 131.99, 2.91, and 0.08 seconds for the former, and 7630, 230, and 5 with corresponding running times of 10.80, 0.62, and 0.02 seconds for the latter. Remarkably, the third-order algorithm required only 0.02% of the iterations of the first-order method and just 2% of the iterations of the second-order method to achieve the same optimization results. In Figure 1.1, it is evident that by leveraging higher-order curvature information, the third-order method allows the iterates to avoid the slow descent paths governed by the gradient and Hessian, directly converging to the global minimum. This numerical behavior is not an isolated case. Identifying and understanding the types of problems third-order methods significantly outperform first- and second-order methods is an active area of our research. More details can be found in [54], which evaluates the AR3 method on a wide range of objective functions with applications to nonlinear programming.

- With Coralia Cartis, Raphael Hauser, Yang Liu, Karl Welzel. *Efficient Implementation of Third-Order Tensor Methods with Adaptive Regularization for Unconstrained Optimization.* [54].

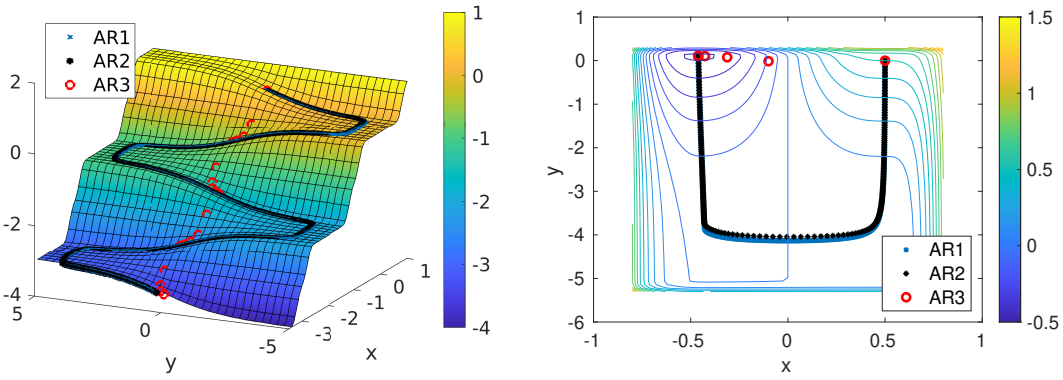


Figure 1.1: Two examples by Liu and Welzel, namely the 'slalom' (left) and 'hairpin turn' (right), illustrate the potential benefits of increasing p. In these examples, the third-order derivative contains key information about the local geometry. The left image shows the periodic slalom function. AR1 and AR2 both take small steps along a curving path. Only in the AR3 method does the local Taylor expansion include crucial information about the curvature in the negative x direction, enabling the iterates to follow the steep "downhill" path. The image on right-hand side is using the hairpin turn function, which has the same structure as one turn of the slalom function but is modified such that it has a global minimum close to $(-0.5, 0)^T$. Again, knowledge of the third derivative allows AR3 to find the minimizer using significantly fewer iterations than AR1 and AR2. For more details about the construction of these functions, see [54, Appendix A.4].

Although when close to local minimizer, composing two quadratically convergent second order (Newton) steps produces a subsequence with local order four, this apparent increase in convergence order results from grouping and re-indexing the iterates. Each composite step still requires two gradient evaluations, two Hessian evaluations, and two Newton-system solves. Therefore, when comparing optimization methods of different orders, the relevant measure is the number of oracle calls or derivative evaluations required to attain a prescribed accuracy, rather than the nominal number of outer iterations.

1.3 Contributions and Structure of the Thesis

The most critical challenge for higher-order methods is the efficient minimization of the pth-order subproblem. For third-order methods (and beyond), the subproblem becomes a nonconvex multivariate polynomial with regularization. To the best of our knowledge, the optimization of such nonconvex polynomials was largely unaddressed before my PhD research. Questions regarding the subproblem's global optimality characterization, theoretical tractability, and the behavior of its local minima (which

differ significantly from the second-order case) remained open. My PhD research narrows these research gaps by characterizing its local and global optimality, investigating the potential NP-hardness of the subproblem, and designing efficient algorithmic frameworks for minimizing the ARp subproblem. We have authored a series of papers addressing the aforementioned challenges in the field of pth-order methods for nonconvex optimization [54; 57; 56; 192; 191]. Our thesis is based on new results from six preprints/papers published or under review in leading peer-reviewed mathematical optimization journals.

Our first solver for the AR3 subproblem, the Quadratic Quartic Regularization (QQR) method, [57; 191] extends Nesterov's method [142] to address nonconvex ARp subproblems with $p = 3$ by approximating the third-order tensor term as a linear combination of quadratic and quartic terms, thereby creating a tensor-free expression for high-order methods. To the best of our knowledge, this is the first solver specifically designed to minimize the AR3 subproblem for nonconvex objective functions (locally). The QQR method can be viewed as a nonconvex generalization of Nesterov's method proposed in [142], which was originally designed for convex quartic regularized polynomials. Our generalization extends the applicability of higher-order tensor methods to a much broader range of objective functions. To achieve accuracy ϵ in the first-order criticality of the sub-problem in finitely many iterations, we show that the error in the QQR method decreases either linearly or by at least $\mathcal{O}(\epsilon^{\frac{4}{3}})$ for locally convex iterations, while in the nonconvex case, by at least $\mathcal{O}(\epsilon)$; thus improving, on these types of iterations, the general cubic-regularization bound. Preliminary numerical experiments indicate that two QQR variants perform competitively with state-of-the-art approaches such as ARC (also known as ARp with $p = 2$), achieving either lower objective value or iteration counts. We present QQR in Chapter 2 based on these papers:

- With Coralia Cartis. *Second-order methods for quartically-regularised cubic polynomials, with applications to high-order tensor methods.* Mathematical Programming, 1-47, 2024. [57]
- With Coralia Cartis. *Quartic polynomial sub-problem solutions in tensor methods for nonconvex optimization.* NeurIPS Workshop Spotlight Presentation, 2022. [191]

The Cubic Quartic Regularization (CQR) method [56] builds upon the QQR method by approximating the local tensor information using a single-parameter cubic term. The role of the cubic term is to crudely approximate local tensor information, while the quartic one provides model regularization and controls progress. We provide necessary and sufficient optimality conditions that fully characterize the global minimizers of these cubic-quartic models. We then turn these conditions into secular equations that can be solved efficiently using root-finding procedures. We show, using our optimality characterizations, that a CQR algorithmic variant has the optimal-order evaluation complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ when applied to minimizing our quartically-regularised cubic subproblem, which can be further improved in special cases. We propose practical CQR variants that judiciously use local tensor information to construct the local cubic-quartic models. We test these variants numerically and observe them to be competitive with cubic regularization and other subproblem solvers on typical instances and even show superior performance on ill-conditioned subproblems with special structures. While this paper develops the theoretical structure and optimality conditions of the CQR model, subsequent work [190], completed during the preparation of this thesis, proposes a tight semidefinite programming (SDP) relaxation for globally solving the CQR subproblem. We present CQR in Chapter 3 based on the paper:

- With Coralia Cartis. *Cubic-quartic regularization models for solving polynomial subproblems in third-order tensor methods.* Mathematical Programming, 1-53, 2025. [56].

The Diagonal Tensor Method (DTM) method [193] extends the CQR approach by minimizing quartically regularized cubic Taylor polynomials iteratively using models that incorporate both diagonal cubic terms and quartic regularization. We also establish the global optimality condition for the AR3 subproblem (ARp Subproblem) and identify the cases where the necessary and sufficient conditions coincide. This condition generalizes existing results and recovers the optimality results for quadratic polynomials with regularization. Leveraging on the optimality characterization, we develop the DTM

framework for minimizing the quartically-regularized cubic subproblem by iteratively minimizing a sequence of local quadratic models that incorporate both a diagonal cubic term and a quartic regularization term (DTM model). For quartically-regularized cubic subproblems with diagonal or low-rank third-order tensor information, the DTM model exactly recovers the subproblem (under a suitable change of basis). While for a general quartically-regularized cubic subproblem, the DTM algorithmic variant is convergent with an evaluation complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$. We propose practical DTM variants that use local diagonal or low-rank tensor information to construct the local cubic-quartic models. Numerical tests show that these variants are competitive with cubic regularization and other subproblem solvers. We present DTM in Chapter 4 based on the paper:

- With Coralia Cartis. *Global Optimality Characterizations and Algorithms for Minimizing Quartically Regularized Third-Order Taylor Polynomials.* Major Revision with Mathematics of Computation. [193].

Moving from QQR to CQR and then to DTM, we employ increasingly general models to approximate (AR3 Model). These methods are effective for different cases. The QQR model is comparatively simpler, while the CQR model is particularly well suited for rank-one tensor problems or tensors with a single dominant direction. The DTM approach performs better for diagonal tensor structures or low-rank subproblems. Through the investigation of iterative solvers for AR3 subproblems (such as QQR, CQR, and DTM), we face a key theoretical question,

> "*Does there exist an efficient and tractable algorithm for globally minimizing a sufficiently regularized ARp subproblem ($p \geq 3$)?*"

A foundational milestone in real algebraic geometry is Hilbert's seminal 1888 paper [95], which provides a complete classification of the degrees and dimensions for which every nonnegative polynomial admits a representation as a sum of squares of polynomials. In this work [95], Hilbert showed that there exist nonnegative polynomials that cannot be expressed as sums of squares, although explicit examples of such polynomials were only constructed nearly eighty years later. The study of the gap between nonnegativity and sums of squares remains an active area of research to this day. It is well known that for polynomials of degree four or higher, deciding nonnegativity is NP-hard (see, for example, the NP-hardness of testing matrix copositivity [132]). Also, Nesterov has shown that maximizing a cubic polynomial over the unit sphere is also NP-hard [143]. Although these results illustrate the general intractability of nonnegativity for high-degree polynomials, regularized polynomials form a more subtle and structured class. As an interesting numerical observation, Parrilo and Sturmfels [158, Sec. 5.1] considered several quartically regularized test functions, and found the corresponding shifted polynomials admit SoS representations. However, these observations were empirical; no theoretical proof, nor any explicit bound on the regularization magnitude, has been established in the literature. In Chapter 5, we establish the first theoretical framework connecting SoS results for a generally nonconvex nonnegative quartically regularized cubic polynomial (AR3 Model). We prove that, when the regularization parameter σ exceeds an explicit threshold, the shifted nonnegative polynomial $m_3 - m_*$ becomes SoS where m_* represents the global minimum value of m_3. This result provides a rigorous explanation of the numerical observations in Parrilo and Sturmfels [158, Sec. 5.1]. The quartically regularized cubic models often become SoS for sufficiently large regularization. Beyond the general case, we identify structured subclasses of quartically regularized cubic polynomials for which nonnegativity and SoS representations coincide, extending classical equivalence results of Hilbert to new families of multivariate nonconvex polynomials, including quadratic–quartic models and cubic terms expressible as norm–linear products. We investigate how regularization affects convexification and global optimality. We also identify structural regimes in which quartically regularized polynomials admit exact SoS certificates. We present the results in Chapter 5 based on the paper:

- With Coralia Cartis. *Sufficiently Regularized Nonnegative Quartic Polynomials are sum-of-squares.* In preparation; target for submission in Q1 2026.

Recently, Ahmadi, Chaudhry, and Zhang [1] introduced a tensor method based on SoS convex reformulations so that each Taylor polynomial sub-problem in their approach can be tractably minimized

using semidefinite programming (SDP) [1]. We introduce an algorithmic framework that combines the SoS convex Taylor model with adaptive regularization techniques for nonconvex smooth optimization problems. Each iteration minimizes an SoS-convex Taylor model, offering a polynomial cost per iteration. the method attains a first-order ϵ-stationary point, that is, a point x satisfying $\|\nabla f(x)\| \leq \epsilon$, within at most $\mathcal{O}(\epsilon^{-2})$ evaluations. For strongly convex functions, this bound improves to $\mathcal{O}(\epsilon^{-1/p})$. To the best of our knowledge, this is the first global rate analysis for an adaptive regularization algorithm with a tractable high-order sub-problem in nonconvex smooth optimization, opening the way for further improvements. We present these results in Chapter 6 based on the paper:

- With Coralia Cartis. *Global Convergence of High-Order Regularization Methods with Sums-of-Squares Taylor Models.* Major Revision with SIAM Journal on Optimization [192].

1.3.1 Structure of the Thesis

The thesis is structured as follows. In Chapter 2, we introduce the Quadratic Quartic Regularisation (QQR) method. In Chapter 3 and Chapter 4, we present the Cubic Quartic Regularisation Method (CQR) and Diagonal Tensor Method (DTM) methods, respectively. The algorithmic frameworks, global minimum characterizations, convergence and complexity analyses, and numerical experiments are discussed in detail in each chapter. In Chapter 5 and Chapter 6, we focus on global optimization techniques for the subproblem and generalizing our approach to $p \geq 3$. Chapter 5 investigates whether a polynomial-time algorithm for globally minimizing the ARp subproblem under sufficiently large regularization. In Chapter 6, we study techniques based on convexified models and introduce an algorithmic framework that combines SoS Convex Taylor models with adaptive regularization for nonconvex smooth optimization. Chapter 7 is concluding remarks and a discussion of future research directions.

Related research content left out of the thesis: Besides the aforementioned research, there are also collaborative papers and projects undertaken during my PhD that are not included in this thesis. For completeness, we provide only a brief review and summary of this research.

- With C Cartis, R Hauser, Y Liu, K Welzel. *Efficient Implementation of Third-order Tensor Methods with Adaptive Regularization for Unconstrained Optimization.* Major Revision with Mathematical Programming Computation [54].

The primary contributors in [54] are Liu and Welzel. We have implemented and tested third-order methods for optimizing mid- and small-scale nonconvex objective functions. Designing ARp variants with efficient regularization parameter updating strategies is also a critical area of interest. In [54], we extend the interpolation-based updating strategies, originally introduced by [80] for $p = 2$, to cases where $p \geq 3$. This updating strategy enables the regularization parameter to adjust adaptively based on interpolation models. A novel prerejection mechanism is also proposed that rejects unwanted subproblem minimizers prior to any function evaluations, thereby reducing computational costs for $p \geq 3$. Numerical results in [54] illustrate how the proposed interpolation-based updating strategy and the prerejection mechanism for $p = 3$ enhance the overall numerical performance.

- With H Chen, C Cartis. *Tensor-based Dinkelbach method for computing generalized tensor eigenvalues and its applications.* Minor Revision with Applied Numerical Mathematics [61].

Prof Chen is the lead contributor of [61]. In this paper, we propose a novel tensor-based Dinkelbach–Type method for computing extremal tensor generalized eigenvalues. We show that the extremal tensor generalized eigenvalue can be reformulated as a critical subproblem of the classical Dinkelbach–Type method, which can subsequently be expressed as a multilinear optimization problem (MOP). The MOP is solved under a spherical constraint using an efficient proximal alternating minimization method which we rigorously establish the global convergence. Additionally, the equivalent MOP is reformulated as an unconstrained optimization problem, allowing for the analysis of the Kurdyka-Lojasiewicz (KL) exponent and providing an explicit expression for the convergence rate of the proposed algorithm.

Preliminary numerical experiments on solving extremal tensor generalized eigenvalues and minimizing high-order trust-region subproblems are provided, validating the efficacy and practical utility of the proposed method. Lastly, we have undertaken other research [115; 116; 179; 194; 178] that is not related to the thesis's topics and therefore not included here.

Chapter 2

Local Methods for the Regularized Third-order Subproblem I

2.1 Introduction

Finding efficient methods to minimize (ARp Subproblem) for $p \geq 3$ (given in Section 1.1.2) on each iteration of ARp remains an open question. As shown in Section 1.1.2, an approximate local minimum of the ARp subproblem is sufficient to ensure the optimal complexity of the ARp algorithm for nonconvex objective functions. Therefore, the primary focus of this chapter, Chapter 3 and Chapter 4 is to find iterative algorithms that find approximate local minima of the AR3 subproblem. We aim to devise algorithms that are designed specifically for minimizing m_3 in (AR3 Model). In this chapter and the following chapters, for notational simplicity, when referring to a specific subproblem, we usually fix x_k and write $m_3(x_k, s)$ as

$$m_3(s) = f_0 + g^T s + \frac{1}{2} s^T H s + \frac{1}{6} T[s]^3 + \frac{\sigma}{4}\|s\|^4, \qquad \text{(Quartically Regularized Cubic Model)}$$

where $f_0 = f(x_k) = m_3(x_k, 0) \in \mathbb{R}$, $g = \nabla_x f(x_k) \in \mathbb{R}^n$, $H = \nabla_x^2 f(x_k) \in \mathbb{R}^{n\times n}$ and $T = \nabla_x^3 f(x_k) \in \mathbb{R}^{n\times n\times n}$. We also denote the fourth-order Taylor expansion of $m_3(s^{(i)} + s)$ at $s^{(i)}$ as

$$M(s^{(i)}, s) := f_i + g_i^T s + \frac{1}{2} s^T H_i s + \frac{1}{6} T_i[s]^3 + \frac{\sigma}{4}\|s\|^4, \tag{2.1}$$

where $f_i = m_3(s^{(i)}) \in \mathbb{R}$, $g_i = \nabla m_3(s^{(i)}) \in \mathbb{R}^n$, $H_i = \nabla^2 m_3(s^{(i)}) \in \mathbb{R}^{n\times n}$ and $T_i = \nabla^3 m_3(s^{(i)}) \in \mathbb{R}^{n\times n\times n}$ and ∇ denotes the derivative with respect to s. For all $s^{(i)}$, the Hessian H_i is a symmetric matrix, and $T_i \in \mathbb{R}^{n\times n\times n}$ is a symmetric third-order tensor[1]. The first- and second-order derivatives of $M(s^{(i)}, s)$ are given by

$$\nabla M(s^{(i)}, s) = g_i + H_i s + \frac{1}{2} T_i[s]^2 + \sigma\|s\|^2 s, \qquad \nabla^2 M(s^{(i)}, s) = H_i + T_i[s] + \sigma(\|s\|^2 I_n + 2ss^T),$$

respectively. Here, $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix, and the matrix ss^T is a rank-one matrix. We refer to the model $M(s^{(i)}, s)$ as locally convex at $s^{(i)}$ if $H_i \succeq 0$, and it is locally nonconvex at $s^{(i)}$ if the smallest eigenvalue of H_i, denoted as $\lambda_{\min}[H_i]$, is negative.

Since $m_3(s)$ is a fourth-degree multivariate polynomial, the fourth-order Taylor expansion is exact. Therefore, we have

$$M(s^{(i)}, s) = m_3(s + s^{(i)}), \qquad \text{and} \qquad \min_{s\in\mathbb{R}^n} m_3(s) = \min_{s\in\mathbb{R}^n} M(s^{(i)}, s).$$

Our aim is to find an iterate $s^{(i)}$ such that

$$\|g_i\| \leq \epsilon, \tag{2.2}$$

[1] A symmetric tensor means that the entries are invariant under any permutation of its indices.

where $\epsilon \in (0,1)$ is an arbitrary user-set tolerance. This implies that $\liminf_{i\to\infty} g_i = 0$, and at least one limit point of the sequence of iterates $\{s^{(i)}\}_{i\geq 0}$ is a first-order stationary point of m_3.

The primary challenge for minimizing (2.1) lies in the third-order tensor term. Notably, when $T_i = 0$, m_3 simplifies to a quadratic model with quartic regularization. We often refer to these kinds of polynomials as the *QQR polynomials* or polynomial models with QQR structure. QQR polynomials have been extensively studied as a variant in the (adaptive) cubic regularization framework [46; 70; 125; 101; 146]. These studies provide necessary and sufficient conditions for the global optimality of QQR polynomials (Theorem 8.2.8, [44]) and efficient algorithms based on secular equations for minimizing the QQR polynomials. Recently, Nesterov proposed a method for bounding the third-order term in a convex AR3 model using a linear combination of second-order derivatives and fourth-order regularization, resulting in a globally convergent algorithm with a linear convergence rate for minimizing the convex AR3 model [142]. Inspired by these developments, we designed our first solver for the nonconvex AR3 subproblem. Extending Nesterov's results to the nonconvex m_3, we estimate the third-order term (locally) using a linear combination of second-order derivatives and fourth-order regularization. We refer to this approach as the *QQR method* or *QQR framework*. This chapter discusses the motivation, algorithmic setup, theoretical convergence, and numerical results of the QQR method.

While the QQR model provides a second-order model devoid of a tensor term, in the next stage of research, we naturally considered approximating the third-order tensor term of m_3 with progressively more accurate approximations. Instead of approximating the third-order tensor term using only a linear combination of the second-and fourth-order terms, in Chapter 3, we introduced a simple cubic term with a scalar coefficient to approximate the tensor term. We propose the *CQR algorithmic framework*, for minimizing (Quartically Regularized Cubic Model), by sequentially minimizing a sequence of local quadratic models that also incorporate both simple cubic and quartic terms. The details of the CQR method are presented in Chapter 2.

Given that the third-order tensor term T_i can contain multiple tensor directions, our next step was to develop a tensor model that can approximate these tensor directions. Leveraging on the optimality characterization and building further on the CQR framework, we developed the *Diagonal Tensor Method (DTM) framework* for minimizing (AR3 Model) by iteratively minimizing a sequence of local quadratic models that incorporate both a diagonal cubic term and a quartic regularization term (DTM model). This method incorporates a diagonal tensor approximation for T_i, making it particularly suitable for subproblems where the tensor term is diagonal or low-rank. Details of the DTM method are outlined in Chapter 4.

From QQR to CQR to DTM method, the models in these methods use increasingly accurate approximations of the tensor term, thereby providing more precise approximations of m_3. However, these models are progressively more challenging to minimize. Therefore, for each model, we carefully derive its optimality conditions and design specific algorithms for its minimization. Each method performs well under different problem characteristics. For example, the DTM excels in AR3 problems with diagonal or low-rank third-order tensors, the CQR method is effective for problems with one dominant direction or entry, and the QQR method is more universal. A comparison of these methods is provided Table 2.1.

2.2 Background and Motivation

The efficiency and scalability of minimizing $m_3(s)$ are crucial to the performance of the AR3 algorithm. Recently, Nesterov gave a linearly convergent second-order method for minimizing convex quartically-regularised cubic polynomials [142]. We refer to this method as *Nesterov's method*. At each iteration, Nesterov approximates the third-order tensor term by a linear combination of second-order derivatives and fourth-order regularization. Then, the algorithm minimizes m_3 by minimizing a sequence of convex quadratic models with quartic regularization, iteratively. To the best of our knowledge, there are no solvers specifically designed for the (local) minimization of nonconvex quartically regularized cubic polynomials.

The key idea presented in this chapter involves the generalization of Nesterov's method to the

Table 2.1: **Overview of Models and Their Properties**. f_i, g_i, H_i, and σ are defined as in (2.1). The models are presented in their most general form, with slight modifications potentially made for algorithmic implementation or to handle special cases. The parameters listed indicate their typical value ranges and are chosen specifically/adaptively in each iteration. More details can be found in the corresponding sections.

Model Name	**Model Notation**	**Parameters To Choose**	**Complexity/ Convergence Rate**	**Applications**
Nesterov Model [142]	$M_\mu^{\mathcal{N}}(s^{(i)}, s) = f_i + g_i^T s + \mu_{\mathcal{N}}^{-1}(\frac{1}{2} - \frac{1}{6\tau_{\mathcal{N}}})H_i[s]^2 + \mu_{\mathcal{N}}^{-3}(1 - 2\tau_{\mathcal{N}})\frac{\sigma}{4}\|s\|^4$.	$\tau_{\mathcal{N}} \approx 0.4929$, $\mu_{\mathcal{N}} \approx 0.193$	Linear rate	Convex AR3 subproblem
QQR-v1 (in Section 2.3.1)	$M_\mu(s^{(i)}, s) = f_i + g_i^T s + \mu^{-\bar{\iota}}(\frac{1}{2} - \frac{1}{6\tau})\big[H_i + \tilde{p}I_n\big][s]^2 + \mu^{-\bar{\kappa}}(1 - 2(1-\hat{q})\tau)\frac{\sigma}{4}\|s\|^4$	$\tau, \bar{\iota}, \bar{\kappa} > 0$, $\tilde{p} \geq 0$, $\hat{q} \in (0, 3]$, $\mu = \mu_{\mathcal{N}}$ $\tau = \tau_{\mathcal{N}}$ for locally convex	Same as `QQR-v2`	AR3 subproblem
QQR-v2 (in Section 2.3.2)	$M_\alpha(s^{(i)}, s) = f_i + g_i^T s + \frac{\alpha_1}{2}\big[H_i + \tilde{p}I_n\big][s]^2 + \frac{\sigma\alpha_2}{4}\|s\|^4$	$\alpha_1 \in (0, 1)$ for locally nonconvex; $\alpha_1 \geq 1$ for locally convex; $\tilde{p}, \alpha_2 \geq 0$	$\mathcal{O}(\epsilon^{\frac{4}{3}})$ for locally convex, $\mathcal{O}(\epsilon)$ for locally nonconvex, $\mathcal{O}(\epsilon^{\frac{3}{2}})$ for locally nearly convex	AR3 subproblem
CQR (in Chapter 3)	$M_c(s^{(i)}, s) = f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{\beta_i}{6}\|s\|_W^3 + \frac{1}{4}(\sigma + d_i)\|s\|_W^4$	$\beta_i \in \mathbb{R}$, $d_i \geq 0$, symmetric pos-def $W \in \mathbb{R}^{n\times n}$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$, w/ acceleration for small tensor, or single entry tensor	AR3 subproblem, particularly if the tensor term has a dominant direction or entry
DMT (in Chapter 4)	$M_d(s^{(i)}, s) = f_0 + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{1}{6}\sum_{j=1}^n t_j^{(i)} s_j^3 + \frac{1}{4}(\sigma + d_i)\|s\|_W^4$.	$\{t_j^{(i)}\}_{1\leq j\leq n} \in \mathbb{R}$, $d_i \geq 0$, symmetric pos-def $W \in \mathbb{R}^{n\times n}$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$, w/ acceleration for low rank subproblem	AR3 subproblem, particularly if the tensor term is diagonal or low-rank

minimization of nonconvex m_3 models. We refer to this framework as the **Quadratic Quartic Regularization Method** (QQR method). In summary, the QQR method minimizes convex or nonconvex AR3 sub-problems of the form (Quartically Regularized Cubic Model) by minimizing a series of quadratic polynomials (that maybe nonconvex) with quartic regularization with the following The chapter makes three key highlights.

- We proposed an algorithmic framework, the QQR method, for efficiently minimizing possibly, nonconvex, quartically-regularised cubic polynomials. To the best of our knowledge, this is the first solver designed to minimize the AR3 sub-problem in (Quartically Regularized Cubic Model) for a nonconvex objective function. QQR method can be seen as a nonconvex generalization of Nesterov's method proposed in [142], which was originally designed for convex m_3 functions. When applied to convex m_3 functions, the QQR method exhibits a similar linearly convergent behaviour as Nesterov's method.
- Theoretically, we show that the QQR method achieves better complexity bounds than the cubic regularization method in locally strictly convex and locally nonconvex cases. For locally convex iterations, the QQR method either gives a linear rate of convergence for the norm of the gradient or an error reduction of at least $\mathcal{O}(\epsilon^{\frac{4}{3}})$, where ϵ represents the accuracy of the first-order optimality condition (2.2). In locally nonconvex iterations, the QQR method achieves an error reduction of at least $\mathcal{O}(\epsilon)$.
- We propose two practical variants of the QQR method. In our numerical experiments, QQR performs competitively with state-of-the-art approaches such as ARC (also known as ARp with

$p = 2$ [46]) for the minimization of m_3. It achieves better results by finding a minimum with fewer algorithmic iterations or function evaluations, or by finding a lower minimum.

Our QQR method is inspired by the works of Cartis, Gould and Toint [44] and Nesterov [142]. In this section, we motivate using the quadratic quartically-regularised model and discuss optimality conditions for this model. We also give a summary of Nesterov's findings and then present our ideas for extending these results to the nonconvex case of quartically regularised cubic polynomials.

The choice of a quadratic quartically-regularised model. The first question to address regarding the QQR method is why we choose a quadratic quartically-regularised model to minimize the AR3 sub-problem (Quartically Regularized Cubic Model). Consider the second-order model with an $\check{r}$th-power regularization:

$$m_2^{\check{r}}(s) = f_0 + g^T s + \frac{1}{2} s^T H s + \frac{\sigma}{\check{r}} \|s\|^{\check{r}},$$

where $\check{r} > 2$ and $f_0 \in \mathbb{R}$, $g \in \mathbb{R}^n$ and $H \in \mathbb{R}^{n \times n}$ are scalar, vector and symmetric matrix coefficients. $m_2^3(s)$ has the same form as the ARC sub-problem and $m_2^4(s)$ has the same form as our quadratic quartic regularization model. The minimization of the potentially nonconvex $m_2^{\check{r}}(s)$ has been widely researched as part of the (adaptive) cubic regularization framework [46; 70; 125; 101; 146]. Theorem 2.2.1 summarizes the necessary and sufficient optimality conditions for the global minimizer of $m_2^{\check{r}}(s)$.

Theorem 2.2.1. *(Theorem 8.2.8 [44]) Let $\check{r} \geq 3$, any global minimizer of $m_2^{\check{r}}(s)$, $\hat{s}_*^{\check{r}}$, satisfies $(H + \hat{\lambda}_* I_n)\hat{s}_*^{\check{r}} = -g$, where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix, $\hat{\lambda}_* \geq 0$, $H + \hat{\lambda}_* I_n \succeq 0$, and $\hat{\lambda}_* = \sigma \|\hat{s}_*^{\check{r}}\|^{\check{r}-2}$. If $H + \hat{\lambda}_* I_n$ is positive definite, then $s_*^{\check{r}}$ is unique.*

Remark 2.2.1 (Connection between the QQR subproblem and the S-lemma)**.** *The global optimality conditions for the quadratic-with-quartic-regularization subproblem in Theorem 2.2.1 with $\check{r} = 2$ are an instance of the optimality theory underpinned by the S-lemma [160; 187]. The S-lemma (or S-procedure) guarantees that, despite nonconvexity, this single-constraint quadratic problem has no duality gap and admits a complete characterization through one Lagrange multiplier: there exists $\hat{\lambda} \geq 0$ with $(H + \hat{\lambda} I)\hat{s} = -g$ and $H + \hat{\lambda} I \succeq 0$. The quartic regularization then ties the multiplier to the step length through the secular equation $\hat{\lambda} = \sigma \|\hat{s}\|^{\check{r}-2}$, turning global minimization into a one-dimensional root-finding problem.*

Theorem 2.2.1 converts the task of identifying the global minimum of $m_2^{\check{r}}$ into a problem involving the solution of a nonlinear equation combined with a matrix linear system. In broad terms, we use the matrix system $(H + \hat{\lambda}_* I_n)\hat{s}_* = -g$ to express $\hat{s}_*$ as a function of $\hat{\lambda}_*$, such that $\hat{s}_* = \hat{s}(\hat{\lambda}_*)$. We obtain an approximate global minimizer $(\hat{s}_*, \hat{\lambda}_*)$ by solving the nonlinear equation $\hat{\lambda}_* = \sigma \|\hat{s}(\hat{\lambda}_*)\|^{\check{r}-2}$. There are many scalable methods for minimizing the sub-problems in $m_2^{\check{r}}$ [46; 123; 119; 97]. Specifically, [48; 44] describe a factorization-based minimization solver for quadratic problems with fourth-order regularization, which we refer to here as the `MS` algorithm (due to its connection to Moré-Sorensen techniques as described next). The `MS` algorithm finds (an approximate) global minimizer of both convex and nonconvex $m_2^{\check{r}}$, by applying Newton's method for root finding to a secular equation together with a Cholesky factorization of the optimality conditions that hold at a global minimizer of $m_2^{\check{r}}$; this is in the same vein as the Moré-Sorensen approach for solving the trust-region subproblem. More details can be found in Ch 8 [44].

Nesterov's method for convex m_3: In [142], Nesterov shows that when m_3 is strictly convex and iteratively minimized using a quadratic quartic model, then the iterates $s^{(i)}$ converge to the minimizer of m_3 at a linear rate as i increases. A summary of Nesterov's results is given in Remark 2.2.2.

Remark 2.2.2. *Assume that $m_3(s)$ is **strictly convex** for all $s \in \mathbb{R}^n$. Then,*

- ***Convex Upper and Lower Bounds** [142, Corollary 1]: For any $\tau \in (\frac{1}{3}, \frac{1}{2})$, let $\alpha^{\pm} =$*

$[\alpha_1^{\pm}, \alpha_2^{\pm}] = [1 \pm \frac{1}{3\tau}, 1 \pm 2\tau]$, *such that*

$$M_{\pm}^{\mathcal{N}}(s^{(i)}, s) = f_i + g_i^T s + \frac{1}{2}(1 \pm \frac{1}{3\tau}) s^T H_i s + \frac{\sigma}{4}(1 \pm 2\tau)\|s\|^4 \tag{2.3}$$

Then, both $M_{\pm}^{\mathcal{N}}(s^{(i)}, s)$ *are convex and* $M_{-}^{\mathcal{N}}(s^{(i)}, s) \leq M(s^{(i)}, s) \leq M_{+}^{\mathcal{N}}(s^{(i)}, s)$ *for all* $s \in \mathbb{R}^n$*, and the equality only occurs when* $s \equiv 0$*. Moreover, we have* $\min_{s\in\mathbb{R}^n} M_{-}^{\mathcal{N}}(s^{(i)}, s) \leq \min_{s\in\mathbb{R}^n} m_3(s) \leq \min_{s\in\mathbb{R}^n} M_{+}^{\mathcal{N}}(s^{(i)}, s)$.

- ***Linear Convergence Result*** *[142, Theorem 3]: Fix* $\tau = \tau_{\mathcal{N}} := \frac{1}{6}\sqrt{(3 + \sqrt{33})}$*. We update the iterates* $s^{(i)}$ *by* $s_+^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_+(s^{(i)}, s)$*, and* $s^{(i+1)} = s^{(i)} + s_+^{(i)}$*. Then, at each iteration, the values of* $m_3(s^{(i)})$ *strictly decrease until they eventually converge to the minimum of* $m_3(s)$*. Then,* $m_3(s^{(i)})$ *satisfies*

$$m_3(s^{(i)}) - m_* \leq (1 - \mu_{\mathcal{N}})^i \left(m_3(s^{(0)}) - m_*\right)$$

where $m_* = \min_{s\in\mathbb{R}^n} m_3(s)$ *and* $\mu_{\mathcal{N}} = 0.193$.
- *The upper and lower bounds,* $M_{\alpha^{\pm}}^{\mathcal{N}}$*, play crucial roles in proving the linear rate of convergence.* $M_{+}^{\mathcal{N}}$ *ensures the monotonic model decrease in* $m_3(s^{(i)})$ *as* i *increases.* $M_{-}^{\mathcal{N}}$ *is used in the complexity proof to construct* $M_{\mu}^{\mathcal{N}}$ *that is used to show the linear rate of convergence. The upper and lower bounds are linked by* $M_{\mu}^{\mathcal{N}}$ *defined as*

$$M_{\mu}^{\mathcal{N}}(s^{(i)}, s) := f_i + g_i^T s + \mu_{\mathcal{N}}^{-1}\left(\frac{1}{2} - \frac{1}{6\tau_{\mathcal{N}}}\right) s^T H_i s + \mu_{\mathcal{N}}^{-3}\left(1 - 2\tau_{\mathcal{N}}\right)\frac{\sigma}{4}\|s\|^4 \tag{2.4}$$

where $\mu_{\mathcal{N}} = 0.193$.

Based on the above approach, we make three key remarks.

- In [142], $M_{\alpha^{\pm}}^{\mathcal{N}}$ is a family of τ-parameterized upper and lower bounds. For convex quartically-regularised cubic polynomials minimization, user-chosen $(\tau_{\mathcal{N}}, \mu_{\mathcal{N}})$ are sufficient to achieve the linear convergence result. However, in the nonconvex case, it is necessary to incorporate some adaptive algorithm parameters to ensure the convergence of $m_3(s^{(i)})$.
- Although the convergence proof for Nesterov's algorithm only applies to globally convex quartic polynomials, we extend it to handle quartic polynomials with only local convexity. We show that under suitable local convexity assumptions, both the linear rate of convergence and upper/lower bound results for convex m_3 can be maintained.
- We use well-established techniques, such as the `MS` method, to obtain the global minimum of the quadratic model with quartic regularization for both m_3.

These observations inspired us to create a QQR framework that minimizes generally nonconvex quartically-regularised cubic polynomials. We demonstrate that, for convex m_3 problems, QQR recovers both of Nesterov's results, namely the upper/lower bound results and the linear convergence result.[2] Figure 2.1 is a simple illustration of the QQR algorithm for a nonconvex univariant quartic polynomial.

Example 2.2.1. *QQR gives strict descent in* $m_3(s^{(i)})$ *in each iteration. In locally nonconvex iterations (i.e.,* $\lambda_{\min}[H_i] < 0$*), we also have a nonconvex upper and a nonconvex lower bound.*

This chapter is structured as follows. Section 2.3 serves as the core of the chapter, introducing the QQR algorithmic framework and two main variants. In this section, we explore the connection between the QQR methods and Nesterov's method, specifically for convex m_3 or locally convex iterations. We provide a comprehensive theoretical analysis of the complexity and convergence properties of the

[2] It is worth noting that Nesterov's method can be applied directly to the minimization of a convex objective function (not just m_3). Similarly, if the objective function satisfies suitable Lipschitz conditions and is bounded below, the QQR method can also be used for local minimization of a potentially nonconvex objective function. However, our interest here is not to create a general new second-order method, but to specifically attempt to efficiently minimize m_3.

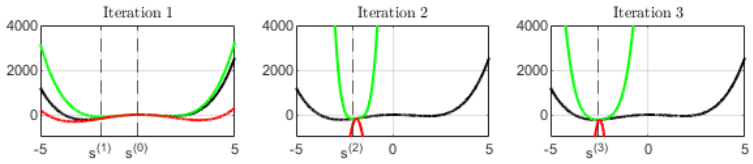


Figure 2.1: Minimizing $m_3(s) = 10s - 50s^2 + 5s^3 + 5s^4$. m_3, M_{α^+}, M_- plotted in black, green, and red respectively. M_- is bounded below on each iteration but the lower bound is outside the range of the plots.

QQR framework. In Section 2.4, we investigate the technical lemmas required for the convergence of the QQR algorithm. These theoretical findings form the basis for the two implementations of the QQR method presented in Section 2.5. We provide numerical examples in Section 2.6, comparing our proposed QQR variants with the state-of-the-art cubic regularization method.

2.3 The Quadratic Quartic Regularization (QQR) Method

QQR minimizes the AR3 sub-problem (i.e., m_3 as defined in (Quartically Regularized Cubic Model)) by globally optimizing a series of potentially nonconvex quadratic models with quartic regularization. The QQR algorithm divides iterations into three distinct cases: **locally strictly convex**, **locally nearly convex**, and **locally nonconvex**, which we refer to as Cases 1, 2, and 3, respectively. Depending on the local convexity of m_3 at each iteration i, the parameters for the quadratic term and quartically regularized term are specifically chosen. Algorithm 2.3.1 gives the framework of the QQR algorithm.

Algorithm 2.3.1: General QQR Framework

Initialization: Initial guess at $s^{(0)} = 0 \in \mathbb{R}^n$, $i := 0$. Set the tolerance for the first-order optimality ϵ. Set $\lambda_c > 0$; λ_c is a user-defined constant used to distinguish among the three distinct cases: locally nonconvex, locally strictly convex, and locally nearly convex. Usually, set $\lambda_c := \epsilon^{1/3}$ and $\eta \in (0, \frac{1}{3})$.

while $\|g_i\| > \epsilon$ **do**

Step 1: QQR Model Selection. Set $M_{\alpha^+}(s^{(i)}, s) := f_i + g_i^T s + \frac{\alpha_1^+}{2} s^T \big(H_i + \tilde{p} I_n\big) s + \frac{\sigma}{4} \alpha_2^+ \|s\|^4$ where $\alpha_1^+ > 0$, $\alpha_2^+ > 0$ and $\tilde{p} \geq 0$ are chosen specifically such that

$$M_+(s^{(i)}, s) \text{ is a (local) upper bound/approximation for } m_3(s) \text{ at } s = s^{(i)}.$$

The method for selecting $\alpha^+ = (\alpha_1^+, \alpha_2^+)$ will be discussed in later sections. Here, we provide a summary of the key features of these constants. At each iteration, compute $\lambda_{\min}[H_i]$ first. The following case distinction is then applied to choose the parameters α_1^+ and $\tilde{p}$ according to the local curvature information.

1. *Locally nonconvex iteration:* i.e., $\lambda_{\min}[H_i] \leq -\lambda_c$. Choose $0 < \alpha_1^+ \leq 1$ and $\tilde{p} = 0$.
2. *Locally strictly convex iteration:* i.e., $\lambda_{\min}[H_i] \geq \lambda_c$. Choose $\alpha_1^+ > 1$ and $\tilde{p} = 0$.
3. *Locally nearly convex iteration:* i.e., $\big|\lambda_{\min}[H_i]\big| < \lambda_c$. Choose $\alpha_1^+ = 1$ and $\tilde{p} > 0$.

Step 2: Iterative Minimization.

$$s_+^{(i)} = \operatorname*{argmin}_{s \in \mathbb{R}^n} M_+(s^{(i)}, s), \qquad s^{(i+1)} = s^{(i)} + s_+^{(i)}, \qquad i := i + 1. \tag{2.5}$$

Step 3: Test for Acceptance. Let

$$\rho(s_+^{(i)}) := \frac{m_3(s^{(i)}) - m_3(s^{(i)} + s_+^{(i)})}{m_3(s^{(i)}) - M_+(s^{(i)}, s_+^{(i)})}. \qquad \text{(QQR Ratio Test)}$$

If $\rho(s_+^{(i)}) \geq \eta$, set $s^{(i+1)} := s^{(i)} + s_+^{(i)}$. Otherwise, set $s^{(i+1)} := s^{(i)}$ and adjust α_1^+, α_2^+ and $\tilde{p}$ adaptively. Let $i = i + 1$.

The section is organized as follows. In Section 2.3.1, we present QQR variant 1 with a single adap-

tive model parameter and discuss the relationship between the QQR methods and Nesterov's method. In Section 2.3.2, we present QQR variant 2 with two adaptive model parameters. In Section 2.3.3, we provide a case-by-case theoretical analysis of the convergence and complexity of QQR methods. We show that QQR methods achieve better complexity than the cubic regularization method during locally strictly convex or nonconvex iterations.

2.3.1 QQR Variant 1: Single Adaptive Model Parameter

Similarly to Nesterov's approach, we construct a QQR variant that adaptively adjusts one parameter. This parameter governs the interplay between the quadratic term and the quartic term.

At the ith iteration of the QQR Framework (Algorithm 2.3.1), we iteratively minimize the QQR variant 1 model (`QQR-v1`) given by:

$$M_\mu(s^{(i)}, s) := f_i + g_i^T s + \frac{\alpha_1^\mu}{2} s^T \big[H_i + \tilde{p} I_n\big] s + \underbrace{\mu^{-\bar{\kappa}}\Big(1 - 2(1-\hat{q})\tau\Big)}_{:=\alpha_2^\mu} \frac{\sigma}{4}\|s\|^4 \tag{2.6}$$

where

$$\begin{aligned}
\text{Locally strictly convex iteration:} \quad & \alpha_1^\mu = \mu^{-\bar{\iota}}\Big(1 - \frac{1}{3\tau}\Big), \qquad \tilde{p} = 0 \\
\text{Locally nearly convex iteration:} \quad & \alpha_1^\mu = \mu^{-\bar{\iota}}\Big(1 - \frac{1}{3\tau}\Big), \qquad \tilde{p} > 0 \\
\text{Locally nonconvex iteration:} \quad & \alpha_1^\mu = 1, \qquad \tilde{p} = 0.
\end{aligned}$$

$\tau, q, \bar{\kappa}, \bar{\iota}$ and $\tilde{p}$ are fixed constants[a] and (only) μ is adaptively adjusted[b].

[a] Details regarding choosing these parameters are given in Section 2.5.1.
[b] For theoretically suitable choices of μ, see Remark 2.3.8 and Corollary 2.3.4; for practical μ, see Section 2.5.2.

Remark 2.3.1. *$\tau, \bar{\iota}, \bar{\kappa} > 0$ and $\hat{q} \in [0, 3)$ are initialized as fixed constants. For a sufficiently small positive μ, we can guarantee $\alpha_1^\mu > 1$ for locally strictly convex iterations (see Algorithm 2.3.1). In practice, we achieve the desired small value of μ by iteratively reducing its magnitude.*

Remark 2.3.2. *For a strictly convex m_3, if we set $\tau = \tau_{\mathcal{N}}$, $\hat{q} = 0$, $\tilde{p} = 0$, $\bar{\iota} = 1$, $\bar{\kappa} = 3$, then the QQR-v1 model in (2.6) coincides with the model proposed by Nesterov in (2.4); namely, $M_\mu = M_\mu^{\mathcal{N}}$. However, it is important to note that when minimizing a convex m_3, one can choose fixed values of (τ, μ) at $(\tau_{\mathcal{N}}, \mu_{\mathcal{N}})$. Therefore, in Nesterov's method, only the upper bound $M_+^{\mathcal{N}}$ needs to be iteratively minimized. In contrast, since we are minimizing a nonconvex m_3, we need to adaptively adjust μ, and thus we iteratively minimize M_μ to allow for changes in the model based on convexity.*

In the following sections, we show that in locally strictly convex iterations when $\lambda_{\min}[H_i] > \lambda_c$, `QQR-v1` locally recovers the upper/lower bounds in Section 2.3.1.1. Additionally, we establish that `QQR-v1` achieves a linear rate of convergence within a region (Section 2.3.1.2).

2.3.1.1 Upper and Lower Bounds in $\mathcal{D}^{(i)}(\hat{q})$

We first define the region within which `QQR-v1` achieves a linear rate of convergence.

Definition 2.3.1. *Let i be a locally strictly convex iteration of `QQR-v1` so that $\lambda_{\min}[H_i] \geq \lambda_c > 0$. For a(ny) $\hat{q} \geq 0$, define a region $\mathcal{D}^{(i)}(\hat{q}) \subseteq \mathbb{R}^n$ around $s^{(i)}$, such that for $s \in \mathcal{D}^{(i)}(\hat{q})$, the following conditions are satisfied:*

1. *$s^T \nabla^2 m_3(s^{(i)} + s)s \geq \hat{q}\|s\|^4$ and $s^T \nabla^2 m_3(s^{(i)} - s)s \geq \hat{q}\|s\|^4$;*
2. *For any $s \in \mathcal{D}^{(i)}(\hat{q})$, we also have $\tau s \in \mathcal{D}^{(i)}(\hat{q})$.*

Remark 2.3.3 (Dependence on the base point and the range of $\hat{q}$)**.** *The set $\mathcal{D}^{(i)}(\hat{q})$ depends on the current iterate $s^{(i)}$, since the Hessian is evaluated at the shifted points $s^{(i)} \pm s$. Thus, more explicitly, one may write $\mathcal{D}^{(i)}(\hat{q}) = \mathcal{D}(s^{(i)}, \hat{q})$. The parameter $\hat{q}$ is chosen in the range $\hat{q} \in [0, \infty)$. Larger values of $\hat{q}$ impose a stronger local curvature requirement and therefore generally lead to a smaller region $\mathcal{D}^{(i)}(\hat{q})$. In particular, $\hat{q} = 0$ corresponds to the least restrictive case. Also, the first condition in Definition 2.3.1 indicates that, if $s \in \mathcal{D}^{(i)}(\hat{q})$, then $-s \in \mathcal{D}^{(i)}(\hat{q})$.*

Lemma 2.3.1. *There exists $\delta_i > 0$ such that $\bar{\mathcal{B}}(0, \delta_i) := \{s \in \mathbb{R}^n, \|s\| \leq \delta_i\} \subseteq \mathcal{D}^{(i)}(\hat{q})$.*

Proof. Since $H_i = \nabla^2 m_3(s^{(i)}) \succ \lambda_c I_n$, there exists $\hat{\delta}_i > 0$ such that for all $s \in \bar{\mathcal{B}}(0, \hat{\delta}_i)$, we have $\nabla^2 m_3(s^{(i)} + s) \succeq \lambda_c I_n$. If $\hat{q} = 0$, set $\delta_i := \hat{\delta}_i$; otherwise choose $\delta_i := \min\{\frac{\lambda_c}{\hat{q}}, \hat{\delta}_i\}$. For all $s \in \bar{\mathcal{B}}(0, \delta_i)$, we have $s^T \nabla^2 m_3(s^{(i)} + s)s \geq \lambda_c \|s\|^2 \geq \hat{q}\|s\|^4$. Clearly, the latter inequalities also hold for $-s$ and τs for $\tau \in (0, 1)$. Therefore, $s \in \mathcal{D}^{(i)}(\hat{q})$ and $\bar{\mathcal{B}}(0, \delta_i) \subseteq \mathcal{D}^{(i)}(\hat{q})$. □

Remark 2.3.4. *($\mathcal{D}^{(i)}(\hat{q})$ and Convexity) If m_3 is strictly convex for all $s \in \mathbb{R}^n$, then $s^T \nabla^2 m_3(s^{(i)} + s)s > 0$ for all $s, s^{(i)} \in \mathbb{R}^n$ implying that $\mathcal{D}^{(i)}(0) = \mathbb{R}^n$. More generally, the size of the region $\mathcal{D}^{(i)}(\hat{q})$ depends on the value of $\hat{q}$. When we set $\hat{q} = 0$, condition 1 in Definition 2.3.1 is less stringent, resulting in a larger region $\mathcal{D}^{(i)}(\hat{q})$ around $s^{(i)}$. Conversely, larger values of $\hat{q}$ lead to smaller regions $\mathcal{D}^{(i)}(\hat{q})$.*

Within the region $\mathcal{D}^{(i)}(\hat{q})$, we can provide a convex upper and lower bound for m_3, Theorem 2.3.1 shows.

Theorem 2.3.1. *(Upper and Lower Bounds in $\mathcal{D}^{(i)}(\hat{q})$) Let i be a locally strictly convex iteration of* `QQR-v1` *such that $\lambda_{\min}[H_i] \geq \lambda_c > 0$. Fix $\hat{q} \in [0, 3)$ and τ such that*

$$\frac{1}{3} < \tau < \min\left\{\frac{1}{2(1 - \hat{q}/3)}, 1\right\}. \tag{2.7}$$

Let

$$M_{\pm}(s^{(i)}, s) = f_i + g_i^T s + \frac{1}{2}\left(1 \pm \frac{1}{3\tau}\right) s^T H_i s + \left[1 \pm 2\big(1 - \frac{\hat{q}}{3}\big)\tau\right] \frac{\sigma}{4}\|s\|^4 \tag{2.8}$$

Then,

1) $M_{\pm}(s^{(i)}, s)$ are convex functions for all $s \in \mathbb{R}^n$.

2) $M_-(s^{(i)}, s) \leq M(s^{(i)}, s) \leq M_+(s^{(i)}, s)$ for all $s \in \mathcal{D}^{(i)}(\hat{q})$, with equality if and only if $s = 0$.

Proof. **Result 1):** For $\frac{1}{3} < \tau < \min\{\frac{1}{2(1-\hat{q}/3)}, 1\}$, we have $\frac{1}{2} \pm \frac{1}{6\tau} > 0$ and $1 \pm 2(1 - \hat{q}/3)\tau > 0$. Thus, $\nabla^2 M_{\alpha^{\pm}}(s^{(i)}, s) = \underbrace{\left(\frac{1}{2} \pm \frac{1}{6\tau}\right) H_i}_{\succ 0} + \underbrace{\frac{\sigma}{4}\left[1 \pm 2\big(1 - \frac{\hat{q}}{3}\big)\tau\right]\Big(\|s\|^2 I_n + 2ss^T\Big)}_{\succ 0} \succ 0.$

Result 2): For any $s \in \mathcal{D}^{(i)}(\hat{q})$, we have $\pm\tau s \in \mathcal{D}^{(i)}(\hat{q})$. Specifically,

$$(\tau s)^T \nabla^2 m_3(s^{(i)} \pm \tau s)(\tau s) - \hat{q}\|\tau s\|^4 > 0. \tag{2.9}$$

An (exact) second-order Taylor expansion of $\nabla^2 m_3(s^{(i)} \pm \tau s)$ implies

$$\nabla^2 m_3(s^{(i)} \pm \tau s) = H_i \pm \tau T_i[s] + 3\sigma\tau^2\|s\|^2 I_n, \tag{2.10}$$

Substituting (2.10) into (2.9), we deduce that $s^T\big[H_i \pm \tau T_i[s] + (3 - \hat{q})\sigma\tau^2\|s\|^2\big]s > 0$ for $0 < \tau < 1$. Now, consider any $s \in \mathcal{D}^{(i)}(\hat{q})$,

$$M_{\alpha^+}(s^{(i)}, s) - M(s^{(i)}, s) = \frac{1}{6\tau} s^T H_i s - \frac{1}{6} T_i[s]^3 + \frac{3 - \hat{q}}{6}\sigma\tau\|s\|^4 = \frac{1}{6\tau} s^T \left[H_i - \tau T_i[s] + (3 - \hat{q})\sigma\tau^2\|s\|^2\right] s > 0.$$

The proof for $M_-(s^{(i)}, s)$ follows similarly. □

Remark 2.3.5 (Self-concordance and the role of the third-order term)**.** *The constructions of this thesis admit a natural interpretation in the language of self-concordance. Recall that a convex f is self-concordant when its third derivative is dominated by its second, $|\nabla^3 f(x)[h]^3| \leq 2\,(\nabla^2 f(x)[h]^2)^{3/2}$; that is, the third-order term is "weak" relative to the second-order one. The methods here exploit exactly such a domination, but on both sides: the cubic Taylor term $T[s]^3$ is bounded above and below by combinations of the quadratic term $s^T H s$ and the quartic regularization $\|s\|^4$. This two-sided control is what makes the convex upper and lower models $M_\pm$ of QQR possible (Theorem 2.3.1). From this viewpoint the quartic regularization plays the role that the 3/2-power of the Hessian plays in self-concordance.*

Remark 2.3.6. *When $\hat{q} = 0$, the upper and lower bounds $M_\pm$ in (2.8) coincides with $M_\pm^{\mathcal{N}}$ in (2.3) in Nesterov's model. Additionally, choosing $0 \leq \hat{q} \leq 3$ ensures that the regularization parameter of the quartic term stays positive, i.e. $1 \pm 2\big(1 - \frac{\hat{q}}{3}\big) \geq 0$.*

2.3.1.2 Linear Convergence of `QQR-v1` in $\mathcal{D}^{(i)}(\hat{q})$

In this subsection, we prove that `QQR-v1` converges linearly to the minimum of the quartically-regularised cubic polynomial in $\mathcal{D}^{(i)}(\hat{q})$ over locally strictly convex iterations. To prove this, we interpret $M_\mu(s^{(i)}, s)$ in (2.6) as an adjustable bound that increases from M_- to M_{α^+} as μ decreases from 1 to 0. It is clear that when $\mu = 1$, M_μ coincides with our local lower bound M_-, and as $\mu \to 0$, $M_\mu(s^{(i)}, s)$ tends to infinity for all $s \in \mathbb{R}^n/\{0\}$. Before presenting the complexity proof for a locally strictly convex iteration, we provide a technical lemma for $M_\mu(s^{(i)}, s)$.

Lemma 2.3.2. *Let $M_\mu(s^{(i)}, s)$ be defined in (2.6) with $\bar{\iota} > 0$, $\bar{\kappa} > 0$, $\tilde{p} = 0$ and $(\hat{q}, \tau)$ satisfying (2.7). If μ satisfies*

$$\mu \leq \min\left\{ \left[\frac{3\tau - 1}{3\tau + 1}\right]^{1/\bar{\iota}}, \left[\frac{1 - 2(1 - \hat{q}/3)\tau}{1 + 2(1 - \hat{q}/3)\tau}\right]^{1/\bar{\kappa}} \right\}, \tag{2.11}$$

then $M_\mu(s^{(i)}, s) \geq M_+(s^{(i)}, s)$ for all $s \in \mathbb{R}^n$.

Proof. The condition on μ ensures $\mu^{-\bar{\iota}}(\frac{1}{2} - \frac{1}{6\tau}) \geq \frac{1}{2} + \frac{1}{6\tau} \geq 0$ and $\mu^{-\bar{\kappa}}[1 - 2(1 - \hat{q}/3)\tau] \geq [1 + 2(1 - \hat{q}/3)\tau] \geq 0$. Additionally, using the fact that $s^T H_i s > 0$ for any $s \in \mathbb{R}^n$ yields the desired result. □

Now, we present the theorem that establishes the linear reduction in the function value in a locally strictly convex iteration.

Theorem 2.3.2. *(Linear Rate of Convergence in $\mathcal{D}^{(i)}(\hat{q})$) Assume that i is a locally strictly convex iteration of `QQR-v1` satisfying $\lambda_{\min}[H_i] \geq \lambda_c$ and $\|g_i\| > \epsilon$, and construct $M_{\alpha^+}(s^{(i)}, s)$ as in Theorem 2.3.1. We update the iterates via iterative minimization in $\mathcal{D}^{(i)}(\hat{q})$,*

$$s_q^{(i)} = \operatorname*{argmin}_{s \in \mathcal{D}^{(i)}(\hat{q})} M_+(s^{(i)}, s), \qquad s^{(i+1)} = s^{(i)} + s_q^{(i)}. \tag{2.12}$$

Then, $m_3(s^{(i+1)})$ satisfies

$$m_3(s^{(i+1)}) - m_{*,\mathcal{D}^{(i)}(\hat{q})} \leq (1 - \mu)\,(m_3(s^{(i)}) - m_{*,\mathcal{D}^{(i)}(\hat{q})}), \tag{2.13}$$

where $m_{*,\mathcal{D}^{(i)}(\hat{q})} := \min_{s \in \mathcal{D}^{(i)}(\hat{q})} m_3(s)$ and μ satisfying (2.11) is an iteration-independent constant.

Proof. Choose parameters

$$0 < \bar{\iota} < 1, \qquad 0 < \bar{\kappa} < 3, \qquad \tilde{p} = 0,$$

so that the convexity conditions in Theorem 2.3.1 and Lemma 2.3.2 are satisfied. For this choice of parameters, the auxiliary model $M_\mu(s^{(i)}, s)$ is jointly convex in (s, μ) for $s \in \mathbb{R}^n$ and $\mu > 0$. Hence the value function

$$M_\mu^*(\mu) = \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M_\mu(s^{(i)}, s) \qquad \text{is convex for } \mu > 0. \tag{2.14}$$

Note that with parameters $\bar{\iota} > 0$, $\bar{\kappa} > 0$ and $\tilde{p} = 0$

$$M_\mu^*(1) = \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M_-(s^{(i)}, s) \leq \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M(s^{(i)}, s) = m_{*,\mathcal{D}^{(i)}(\hat{q})}. \tag{2.15}$$

On the other hand, by continuity, we have

$$M_\mu^*(0) = \min_{s \in \mathcal{D}^{(i)}(\hat{q})} \lim_{\mu \to 0} M_\mu(s^{(i)}, s) = m_3(s^{(i)}). \tag{2.16}$$

Hence,

$$\begin{aligned}
m_3(s^{(i+1)}) \quad &= \quad m_3(s^{(i)} + s_q^{(i)}) = M(s^{(i)}, s_q^{(i)}) \\
&\underset{\text{Thm 2.3.1(2)}}{\leq} \quad M_+(s^{(i)}, s_q^{(i)}) = \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M_+(s^{(i)}, s) \underset{\text{Lemma 2.3.2}}{\leq} M_\mu(s^{(i)}, s) \\
&\underset{(2.14)}{\leq} \quad \mu M_\mu^*(1) + (1-\mu) M_\mu^*(0) \underset{(2.16),(2.15)}{\leq} \mu \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M_-(s^{(i)}, s) + (1-\mu) m_3(s^{(i)}) \\
&\underset{\text{Thm 2.3.1(2)}}{\leq} \quad \mu \min_{s \in \mathcal{D}^{(i)}(\hat{q})} M(s^{(i)}, s) + (1-\mu) m_3(s^{(i)}) = \mu m_{*,\mathcal{D}^{(i)}(\hat{q})} + (1-\mu) m_3(s^{(i)}).
\end{aligned}$$

Subtracting $m_{*,\mathcal{D}^{(i)}(\hat{q})}$ from both sides and rearranging gives the result. □

Remark 2.3.7. *(Minimizer $s_q^{(i)}$ is well defined) The quartic regularization term of $M_{\alpha^+}(s^{(i)}, s) \to \infty$ as $\|s\| \to \infty$. Thus, there exists a constant r_q such that if $\|s\| > r_q$, then $M_{\alpha^+}(s^{(i)}, s) \geq f_i = M_+(s^{(i)}, 0)$. Given that $0 \in \mathcal{D}^{(i)}(\hat{q})$, this implies that the minimizer of $M_{\alpha^+}(s^{(i)}, s)$ must satisfy*[3] *$\|s_q^{(i)}\| \leq r_q$. Therefore, we can write the constrained optimization problem as*

$$s_q^{(i)} = \operatorname*{argmin}_{s \in \mathcal{D}^{(i)}(\hat{q})} M_+(s^{(i)}, s) = \operatorname*{argmin}_{s \in \mathcal{D}^{(i)}(\hat{q}) \cup \{s \in \mathbb{R}^n, \|s\| \leq r_q\}} M_+(s^{(i)}, s).$$

As $\mathcal{D}^{(i)}(\hat{q}) \cap \{s \in \mathbb{R}^n, \|s\| \leq r_q\}$ is a closed and bounded domain, the minumum of m_3 within this domain must be attained, and so $s_q^{(i)}$ is well defined.

The proof of Theorem 2.3.1 follows a similar approach as Nesterov's proof [142, Theorem 2]. Theorem 2.3.1 can be considered as a generalization of the linear convergence result for Nesterov's method to the case of nonconvex m_3 models. In Corollary 2.3.1, we present a concise proof illustrating how the linear convergence result of Nesterov's method can be derived from a special case of Theorem 2.3.1.

Corollary 2.3.1. *(Linear Rate of Convergence for Convex m_3.) Assume $m_3(s)$ is strictly convex for all $s \in \mathbb{R}^n$, and construct $M_{\alpha^+}(s^{(i)}, s)$ as in Theorem 2.3.1 and $s^{(i)}$ is updated by Step 2 of QQR algorithm using (2.5). Then, $m_3(s^{(i)})$ satisfies*

$$m_3(s^{(i)}) - m_* \leq (1-\mu)^i \left(m_3(s^{(0)}) - m_*\right),$$

where $m_* := \min_{s \in \mathbb{R}^n} m_3(s)$ and μ satisfies $0 < \mu \leq \min\left\{\frac{3\tau-1}{3\tau+1}, \left[\frac{1-2\tau}{1+2\tau}\right]^{1/3}\right\}$ for $\tau \in (\frac{1}{3}, \frac{1}{2})$.

Proof. According to (2.4) and Remark 2.3.2, if $m_3(s)$ is strictly convex, then M_μ takes the form $M_\mu^{\mathcal{N}}$ with $\tau = \tau_{\mathcal{N}}$, $\hat{q} = 0$, $\tilde{p} = 0$, $\bar{\iota} = 1$, $\bar{\kappa} = 3$. Additionally, the region $\mathcal{D}^{(i)}(0)$ encompasses the entire space $\mathbb{R}^n$. As a result, the constrained minimization $m_{*,\mathcal{D}^{(i)}(\hat{q})} = \min_{s \in \mathcal{D}^{(i)}(\hat{q})} m_3(s)$ in (2.12) simplifies to an unconstrained minimization $m_* = \min_{s \in \mathbb{R}^n} m_3(s)$. Since m_* is independent of the iteration, we can apply induction on i to the inequality $m_3(s^{(i+1)}) - m_* \leq (1-\mu)(m_3(s^{(i)}) - m_*)$ and obtain the desired result. □

Remark 2.3.8. *The constant μ is crucial for determining the linear convergence rate. For nonconvex m_3, μ needs to satisfy (2.11). In Nesterov's method, which is a special case of the nonconvex framework, we have $\hat{q} = 0$, $\bar{\iota} = 1$, and $\bar{\kappa} = 3$; therefore, $\tau_{\mathcal{N}}$ is the unique positive root of the equation $\frac{3\tau-1}{3\tau+1} = \left[\frac{1-2\tau}{1+2\tau}\right]^{1/3}$. This gives $\tau_{\mathcal{N}} = \frac{1}{6}\sqrt{(3+\sqrt{33})} \approx 0.4929$ and $\mu_{\mathcal{N}} \approx 0.193$.*

[3] A similar argument is used in the proof of Lemma 2.3.3 (see Section 2.4.1 for more details).

As we have derived a more general model than the model given by Nesterov, we gain increased flexibility in selecting the parameters $\hat{q}$, $\bar{\iota}$, and $\bar{\kappa}$ to attain a faster linear rate within the region $\mathcal{D}^{(i)}(\hat{q})$. In Section 2.5.1, we will explore the practical implementation of the algorithm and provide detailed discussions on the choices of $\hat{q}$, $\bar{\iota}$, and $\bar{\kappa}$ to achieve a faster linear rate within $\mathcal{D}^{(i)}(\hat{q})$.

It is important to highlight that if the following condition holds

$$\sigma > \max(s_0^{-2}, 1)\left[\max[-\lambda_{\min}(H), 0] + \hat{L}_H\right] > 0, \tag{2.17}$$

where $s_0 > 0$ and $\hat{L}_H := \max_{v_j \in \mathbb{R}^n, |v_j|=1, j=1,2,3} \left|T_i[v_1][v_2][v_3]\right|$, then $m_3(s)$ is convex for all $\|s\| \geq s_0$. Condition (2.17) implies that for an arbitrarily small $s_0 > 0$, there exists a sufficiently large σ such that m_3 is globally convex apart from a small neighbourhood, $\|s\| < s_0$. Several efficient algorithms, such as Nesterov's algorithm [142] or `QQR-v1`, could be employed to minimize a (nearly) convex m_3, when initialised in the (convex) region and provided the iterates remain within the respective region of convexity of m_3. However, a large σ value enforces a stringent step control due to the regularization term, resulting in possibly exceedingly small step sizes for minimizing m_3. This will typically be detrimental to the practical performance of the AR3 algorithm [23; 51; 53][4] when applied to minimizing a general objective f by repeatedly minimizing m_3 local models, as it is crucial then to adapt σ to the local landscape, allowing longer steps opportunistically and also to make it independent of difficult-to-calculate or unavailable problem constants. Thus, within each AR3 sub-problem, we cannot and should not adjust σ to satisfy the bound in (2.17), and as a result, the sub-problem can become nonconvex when dealing with nonconvex objective functions. This justifies the necessity for QQR methods, which apply to both convex and nonconvex m_3.

To conclude this subsection, we note that we did not provide convergence or complexity bounds for nonconvex iterations of `QQR-v1` in this subsection. The discussions for the convergence of `QQR-v1` are deferred to Section 2.3.2 and Section 2.5.1. This is because QQR variant 1 can be considered as a special case of QQR variant 2, as shown in Corollary 2.3.4. On the other hand, QQR variant 2, introduced in Section 2.3.2, contains two adaptive model parameters and is considered a more generic framework. By selecting suitable values for $\hat{q}$, τ, $\tilde{p}$, $\bar{\iota}$, and $\bar{\kappa}$, and adaptively decreasing μ to a sufficiently small value, QQR variant 1 can conform to the requirements of QQR variant 2 and, thus, exhibit similar convergence behaviour.

2.3.2 QQR Variant 2: Two Adaptive Model Parameters

In this section, we construct a QQR variant that has two adaptive model parameters, α_1 and α_2. We provide the quadratic and quartic bounds and approximations for $m_3(s)$ at locally convex, locally nonconvex, and locally nearly convex iterations, which we refer to as Cases 1, 2, and 3, respectively.

At the ith iteration, we iteratively minimize the QQR variant 2 model (`QQR-v2`) given by:

$$M_\alpha(s^{(i)}, s) := f_i + g_i^T s + \frac{\alpha_1}{2} s^T \left[H_i + \tilde{p} I_n\right] s + \frac{\sigma \alpha_2}{4} \|s\|^4 \tag{2.18}$$

where $\tilde{p} > 0$ is a fixed constant for locally nearly convex iterations and $\tilde{p} = 0$ otherwise.

- The key idea is to adaptively adjust α_1 to switch between locally nonconvex iterations and locally strictly convex iterations, and adaptively adjust α_2 to control regularization.
- In general, we choose $\alpha_1 \in (0, 1)$ for locally nonconvex iterations and $\alpha_1 > 1$ for locally strictly convex iterations[a].

[a] For theoretically suitable choices of (α_1, α_2), see Theorems 2.3.3 and 2.3.4; for practical choices of (α_1, α_2), see Section 2.5.2.

[4] It remains to be seen if such a large pre-defined regularization parameter as in (2.17) would be detrimental to the theoretical complexity of AR3.

We require the following technical lemmas to establish the convergence results for `QQR-v2`. Proofs of these lemmas will be presented in Section 2.4.

Definition 2.3.2. *(Following [53]) Let $T \in \mathbb{R}^{n^3}$ be a third-order tensor. The tensor norm of T is*

$$\Lambda_0 := \max_{\|v_1\|=\|v_2\|=\|v_3\|=1} \left|T[v_1][v_2][v_3]\right| \tag{2.19}$$

where $v_1, v_2, v_3 \in \mathbb{R}^n$. $T[v_1][v_2][v_3]$ stands for applying the third-order tensor T to the vectors v_1, v_2, v_3.

According to [43, Thm 2.4], $\Lambda_0 \leq \left[\sum_{1\leq i,j,k\leq,n} |T_{[i,j,k]}|^2.\right]^{1/2}$ where $T_{i,j,k}$ denotes the i,j,kth entry of T.

Lemma 2.3.3. *(Upper Bound for $\|s_*\|$) Let $s_* := \operatorname{argmin}_{s\in\mathbb{R}^n} m_3(s)$ and $r_c > 0$ be the unique solution of the nonlinear equation $\sigma = 4\left(\frac{\|g\|}{r_c^3} + \frac{\lambda_0}{2r_c^2} + \frac{\Lambda_0}{6r_c}\right)$, where $\lambda_0 := \max\{-\lambda_{\min}(H), 0\}$ where Λ_0 is defined in Definition 2.3.2, and H denotes the symmetric coefficient matrix of the quadratic terms in m_3. Moreover, for any $\|s\| > r_c$, $m_3(s) > f_0$.*

Proof. The proof is given in Section 2.4.1. □

Corollary 2.3.2. *There exist iteration-independent positive constants, such that*

$$\max_{u\in\mathbb{R}^n,\|u\|=1,\|s\|\leq r_c} \nabla^3 m_3(s)[u]^3 \leq L_H, \qquad \max_{u\in\mathbb{R}^n,\|u\|=1,\|s\|\leq r_c} u^T\nabla^2 m_3(s)u \leq L_g$$

where $L_H = \Lambda_0 + 6\sigma r_c$, $L_g = |\lambda_{\max}(H)| + \frac{1}{2}\Lambda_0 r_c + 3\sigma r_c^2$, and r_c is defined in Lemma 2.3.3.

Proof. Corollary 2.3.2 is a consequence of Lemma 2.4.1 which is discussed and proved in Section 2.4.2. Specifically, by substituting $\|s\| \leq r_c$ in (2.56) and (2.57) in Lemma 2.4.1 gives the result. □

We refer to L_g and L_H as the local Lipschitz constants for the gradient and Hessian of m_3, respectively (which are global in the neighbourhood $\|s\| \leq r_c$). A discussion of alternative definitions of Lipschitz constants is given in Remark 2.4.1.

2.3.2.1 Bound for Locally Strictly Convex and Nearly Convex Iterations

In Theorem 2.3.3, we introduce a quadratic quartically-regularised upper/lower bound for m_3 that applies in both locally strictly convex and locally nearly convex regimes.

Theorem 2.3.3. *Given $s^{(i)} \in \mathbb{R}^n$ and $\|s^{(i)}\| \leq r_c$, let $M_{\alpha^\pm}(s^{(i)}, s) := f_i + g_i^T s + \frac{\alpha_1^\pm}{2} s^T(H_i \pm \tilde{p}I_n)s + \frac{\sigma}{4}\alpha_2^\pm\|s\|^4$. Let $\alpha_1^\pm = 1 \pm a$ and $\alpha_2^\pm = 1 \pm d$. Choose $\lambda_c > 0$ as a constant, a and $\tilde{p}$ as*

$$\textbf{\textit{Case 1}}\ (\lambda_{\min}[H_i] > \lambda_c)\textbf{:} \qquad a > 0, \qquad \tilde{p} = 0, \qquad d > \frac{L_H^2}{18a\lambda_c\sigma}, \tag{2.20}$$

$$\textbf{\textit{Case 2}}\ (\left|\lambda_{\min}[H_i]\right| \leq \lambda_c)\textbf{:} \qquad a = 0, \qquad \tilde{p} > 0, \qquad d > \frac{L_H^2}{18\tilde{p}\sigma}, \tag{2.21}$$

where L_H is defined in Corollary 2.3.2. Then,

1. *$M(s^{(i)}, s)$ satisfies the following upper and lower bounds $M_-(s^{(i)}, s) \leq M(s^{(i)}, s) \leq M_+(s^{(i)}, s)$ for all $s \in \mathbb{R}^n$ with the equality only at $s \equiv 0$.*
2. *Letting $s_+^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_+(s^{(i)}, s)$ and $s^{(i+1)} := s^{(i)} + s_+^{(i)}$, we have $\|s^{(i+1)}\| \leq r_c$.*

Proof. **Case 1:** For any $\|s^{(i)}\| \leq r_c$, we have $\|T_i\| = \max_{u\in\mathbb{R}^n,\|u\|=1} \nabla^3 m_3(s^{(i)})[u]^3 \leq L_H$ by Corollary 2.3.2. Therefore,

$$\begin{aligned} M_{\alpha^+}(s^{(i)},s) - M(s^{(i)},s) &= \frac{a}{2}s^T H_i s - \frac{1}{6}T_i[s]^3 + \frac{d\sigma}{4}\|s\|^4 \\ &\geq \|s\|^2\left(\frac{a\lambda_c}{2} - \frac{L_H}{6}\|s\| + \frac{d\sigma}{4}\|s\|^2\right). \end{aligned} \tag{2.22}$$

The discriminant of the quadratic equation in (2.22) is $\Delta_s = \frac{L_H^2}{36} - \frac{a\lambda_c}{2}d\sigma$. Using (2.20), we obtain that $\Delta_s < 0$ and $\frac{a\lambda_c}{2} - \frac{L_H}{6}\|s\| + \frac{d\sigma}{4}\|s\|^2 > 0$. Consequently, $M_{\alpha^+}(s^{(i)},s) - M(s^{(i)},s) \geq 0$ for all $s \in \mathbb{R}^n$ with equality only at $s \equiv 0$. Similar proof follows for $M_-(s^{(i)},s)$.

Now, we prove the second result. We assume by contradiction that $\|s^{(i)}\| \leq r_c$ but $\|s^{(i+1)}\| = \|s^{(i)} + s_+^{(i)}\| > r_c$. By Result 1), we have

$$M_{\alpha^+}(s^{(i)},s_+^{(i)}) - f_i \geq M(s^{(i)},s_+^{(i)}) - f_i. \tag{2.23}$$

Using $\|s^{(i)} + s_+^{(i)}\| > r_c$, Lemma 2.3.3 and the monotonically decreasing nature of $\{f_i\}_{i>0}$, we have

$$M(s^{(i)},s_+^{(i)}) - f_i = m_3(s^{(i)} + s_+^{(i)}) - f_i > m_3(s^{(i)} + s_+^{(i)}) - f_0 > 0.$$

Therefore, we obtain that $M_{\alpha^+}(s^{(i)},s_+^{(i)}) - f_i > 0$ which contradicts $s_+^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_+(s^{(i)},s)$.

Case 2: We deduce that $M_{\alpha^+}(s^{(i)},s) - M(s^{(i)},s) \geq \|s\|^2\left(\frac{\tilde{p}}{2} - \frac{L_H}{6}\|s\| + \frac{d\sigma}{4}\|s\|^2\right)$. The same argument follows if we replace $a\lambda_c$ by $\tilde{p}$. □

Remark 2.3.9. *(A Discussion of Performance Ratio Test) Using the performance ratio to govern progress is a standard numerical technique employed, for example, in ARp algorithms [45; 51; 53], trust region algorithms [44, Sec 3.2.1] and the ARC algorithm [48]. Result 1 of Theorem 2.3.3 corresponds to* (QQR Ratio Test) *with* $\eta = 1$. *Using* $M(s^{(i)},s) \leq M_+(s^{(i)},s)$, *we obtain that*

$$\rho(s) = \frac{m_3(s^{(i)}) - m_3(s^{(i)}+s)}{m_3(s^{(i)}) - M_+(s^{(i)},s)} \geq 1 \geq \eta \tag{2.24}$$

for all $\eta \in (0,1]$ *and all* s. *Note that, for* $\eta \in (0,1]$, *Result 2 of Theorem 2.3.3 also holds for* $s_+^{(i)}$ *satisfying* $\rho(s_+^{(i)}) \geq \eta$. *To see this, we can rewrite* (2.23) *as* $M_{\alpha^+}(s^{(i)},s_+^{(i)}) - f_i \geq \eta\big(M(s^{(i)},s_+^{(i)}) - f_i\big)$ *with* $\eta \in (0,1)$, *and then Result 2 of Theorem 2.3.3 follows.*

Theorem 2.3.3 establishes the upper and lower for $M(s^{(i)},s)$ for locally convex iterations. With appropriate parameter selections, these bounds can be simplified to (2.3) in the Nesterov model. Corollary 2.3.3 establishes this relationship.

Corollary 2.3.3. *(Link to Nesterov Bound) For a strictly convex* m_3, *Theorem 2.3.3 simplifies to the upper and lower bounds results in* (2.3), *which is given by Nesterov in [142, Corollary 1].*

Proof. To see this, assume that we have $H_i \succeq \lambda_c I_n \succ 0$ for all i, then

$$0 \prec \nabla^2 m_3(s^{(i)}+s) = H_i + T_i s + \sigma(\|s\|^2 + 2ss^T). \tag{2.25}$$

Applying s twice to (2.25), we deduce $0 < \lambda_c - L_H\|s\| + 3\sigma\|s\|^2$. Ensuring the discriminant of the quadratic equation stays negative gives $L_H^2 - 12\sigma\lambda_c \leq 0$. By setting $a = \frac{1}{3\tau}$, we have

$$d \geq \frac{L_H^2}{18a\lambda_c\sigma} = \frac{L_H^2\tau}{6\lambda_c\sigma} \geq 2\tau.$$

where the last inequality uses $L_H^2 \leq 12\sigma\lambda_c$. Therefore, for strictly convex m_3, the upper and lower bounds also reduce to $\alpha^\pm = (\alpha_1^\pm, \alpha_2^\pm) = (1 \pm \frac{1}{3\tau}, 1 \pm 2\tau)$. Moreover, the restriction for $\alpha_1^\pm > 0$ and $\alpha_2^\pm > 0$ gives the bound for $\tau \in (\frac{1}{3}, \frac{1}{2})$. □

2.3.2.2 Bound for Locally Nonconvex Iterations

In a locally nonconvex iteration, where $\lambda_{\min}[H_i] \leq -\lambda_c$, instead of proving a upper and lower bound for all $s \in \mathbb{R}^n$, we prove that, given a constant $\eta \in (0, \frac{1}{3})$, there exist suitable values of $(\alpha_1^+, \alpha_2^+) > 0$

$$M_{\alpha^+}(s^{(i)}, s) := f_i + g_i^T s + \frac{\alpha_1^+}{2} s^T H_i s + \frac{\sigma}{4}\alpha_2^+ \|s\|^4 \tag{2.26}$$

such that (QQR Ratio Test) is satisfied at the minimizer $s_+^{(i)} := \operatorname{argmin}_{s\in\mathbb{R}} M_+(s^{(i)}, s)$. We first present a technical lemma and then prove the result in Theorem 2.3.4.

Lemma 2.3.4. *Let M_{α^+} be defined as in (2.26) and $s_+^{(i)} := \operatorname{argmin} M_+(s^{(i)}, s)$. Assume that $\lambda_{\min}[H_i] \leq -\lambda_c$ for some $\lambda_c > 0$. Then,*

$$g_i^T s_+^{(i)} + \frac{\alpha_1^+}{2} s_+^{(i)T} H_i s_+^{(i)} \leq \frac{\alpha_1^+}{2}\lambda_{\min}[H_i]\|s_+^{(i)}\|^2 \leq -\frac{\alpha_1^+}{2}\lambda_c \|s_+^{(i)}\|^2 < 0. \tag{2.27}$$

Proof. Let $\pm v_1$ be the unit eigenvector that corresponds to the smallest eigenvalue of H_i, such that $(\pm v_1)^T H_i (\pm v_1) = \lambda_{\min}[H_i] \leq -\lambda_c$. We choose the sign v_1 such that $g^T v_1 \leq 0$. Since $s_+^{(i)} = \operatorname{argmin} M_\alpha(s^{(i)}, s)$, therefore we have $M_\alpha(s^{(i)}, s_+^{(i)}) - f_i \leq M_\alpha(s^{(i)}, v_1\|s_+^{(i)}\|) - f_i$. Consequently,

$$\begin{aligned} &g_i^T s_+^{(i)} + \frac{\alpha_1^+}{2} s_+^{(i)T} H_i s_+^{(i)} + \frac{\sigma}{4}\alpha_2^+ \|s_+^{(i)}\|^4 \\ &\leq \underbrace{g^T v_1}_{<0}\|s_+^{(i)}\| + \frac{\alpha_1^+}{2} \underbrace{v_1^T H_i v_1}_{=\lambda_{\min}[H_i]\leq -\lambda_c} \|s_+^{(i)}\|^2 + \frac{\sigma}{4}\alpha_2^+ \underbrace{\|v_1\|^4}_{=1}\|s_+^{(i)}\|^4. \end{aligned}$$

Thus, we obtain (2.27). □

Theorem 2.3.4. *Let $\eta \in (0, \frac{1}{3})$ and M_{α^+} be defined as in (2.26). Assume that the minimum of $M(s^{(i)}, s)$ is not attained at $s = 0$. Given $s^{(i)} \in \mathbb{R}^n$ and $\|s^{(i)}\| < r_c$, and $\lambda_{\min}[H_i] \leq -\lambda_c$. We denote $\alpha_2^+ := \eta(1+d)$ and choose*

$$\alpha_1^+ \in \left[\max\left\{\frac{2}{3(1-\eta)}, 1 - \frac{|\lambda_{\min}[H_i]|}{2|\lambda_{\max}[H_i]}\right\}, 1\right] \subset (0, 1], \qquad d > \frac{L_H^2}{3\sigma\lambda_c}. \tag{2.28}$$

Then, the following holds

1. *$M(s^{(i)}, s_+^{(i)}) - f_i \leq \eta\big[M_{\alpha^+}(s^{(i)}, s_+^{(i)}) - f_i\big]$. Namely, the ratio test in (QQR Ratio Test) is satisfied.*
2. *Let $s_+^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_+(s^{(i)}, s)$ and $s^{(i+1)} := s^{(i)} + s_+^{(i)}$, then $\|s^{(i+1)}\| \leq r_c$.*

Proof. **Result 1:** To prove the first result, we define $\mathcal{E}$ as below. We aim to prove that $\mathcal{E} \geq 0$ provided the choice of (α_1^+, d) in Theorem 2.3.4 holds.

$$\begin{aligned} \mathcal{E}: &= \eta\left[M_{\alpha^+}(s^{(i)}, s_+^{(i)}) - f_i\right] - \left[M(s^{(i)}, s_+^{(i)}) - f_i\right] = \left[f_i - M(s^{(i)}, s_+^{(i)})\right] - \eta\left[f_i - M_{\alpha^+}(s^{(i)}, s_+^{(i)})\right] \\ &= \eta\left[g_i^T s_+^{(i)} + \frac{\alpha_1^+}{2} s_+^{(i)T} H_i s_+^{(i)} + \frac{\sigma}{4}\left[\eta^{-1}(1+d)\right]\|s_+^{(i)}\|^4\right] \\ &\quad - \left[g_i^T s_+^{(i)} + \frac{1}{2} s_+^{(i)T} H_i s_+^{(i)} + \frac{1}{6} T_i[s_+^{(i)}]^3 + \frac{\sigma}{4}\|s_+^{(i)}\|^4\right] \\ &= \underbrace{(\eta - 1)}_{<0} g_i^T s_+^{(i)} + \frac{1}{2}\underbrace{(\alpha_1^+\eta - 1)}_{<0} s_+^{(i)T} H_i s_+^{(i)} - \frac{1}{6} T_i[s_+^{(i)}]^3 + \frac{\sigma}{4} d\|s_+^{(i)}\|^4. \end{aligned}$$

Using Lemma 2.3.4, we know that

$$\underbrace{(\eta - 1)}_{<0} g_i^T s_c + (\eta - 1)\frac{\alpha_1^+}{2} s_+^{(i)T} H_i s_+^{(i)} \geq \frac{\alpha_1^+}{2}(\eta - 1)\lambda_{\min}[H_i]\|s_+^{(i)}\|^2 > 0. \tag{2.29}$$

Substituting (2.29) into $\mathcal{E}$, we deduce that

$$\begin{aligned}\mathcal{E} &\geq \frac{\alpha_1^+}{2}(\eta-1)\lambda_{\min}[H_i]\|s_+^{(i)}\|^2 - (\eta-1)\frac{\alpha_1^+}{2}s_+^{(i)^T}H_i s_+^{(i)} \\ &\quad + \frac{1}{2}\underbrace{(\alpha_1^+\eta - 1)}_{<0}s_+^{(i)^T}H_i s_+^{(i)} - \frac{T_i[s_+^{(i)}]^3}{6} + \frac{\sigma}{4}d\|s_+^{(i)}\|^4 \\ &\geq \underbrace{\frac{\alpha_1^+}{2}(\eta-1)\lambda_{\min}[H_i]}_{>0}\|s_+^{(i)}\|^2 + \frac{1}{2}\underbrace{(\alpha_1^+ - 1)}_{<0}s_+^{(i)^T}H_i s_+^{(i)} - \frac{T_i[s_+^{(i)}]^3}{6} + \frac{\sigma}{4}d\|s_+^{(i)}\|^4. \end{aligned} \tag{2.30}$$

Case A: If $s_+^{(i)^T}H_i s_+^{(i)} \leq 0$, for all $\alpha_1^+ \in (0,1]$, following (2.30), we have

$$\mathcal{E} \geq \Bigg[\overbrace{\underbrace{\frac{\alpha_1^+}{2}(1-\eta)\lambda_c}_{>0} - \underbrace{\frac{L_H}{6}\|s_+^{(i)}\|}_{<0} + \frac{\sigma}{4}d\|s_+^{(i)}\|^2}^{\amalg_e(\|s_+^{(i)}\|)}\Bigg]\|s_+^{(i)}\|^2 \tag{2.31}$$

where we use $T_i[s_+^{(i)}]^3 \leq L_H\|s_+^{(i)}\|$ from Corollary 2.3.2 and $\lambda_{\min}[H_i] \leq -\lambda_c$. By applying the condition $\alpha_1^+ \geq \frac{2}{3}(1-\eta)^{-1}$ and $d > \frac{L_H^2}{3\sigma\lambda_c}$, this makes the discriminant of the quadratic equation $\amalg_e(\|s_+^{(i)}\|)$, $\Delta = \frac{L_H^2}{36} - \frac{\alpha_1^+}{2}(1-\eta)\lambda_c\sigma d < 0$. Therefore, we deduce that $\mathcal{E} \geq 0$.

Case B: If $s_+^{(i)^T}H_i s_+^{(i)} > 0$, let $\alpha_1^+ \in \left(\max\left\{\frac{2}{3}(1-\eta)^{-1}, 1 - \frac{|\lambda_{\min}[H_i]|}{|2\lambda_{\max}[H_i]|}\right\}, 1\right] \subset (0,1]$.

- Since we have $1 - \frac{-\lambda_{\min}[H_i]}{2\lambda_{\max}[H_i]} \leq \alpha_1^+ \leq 1$, we deduce that

$$0 < \underbrace{(1-\alpha_1^+)}_{>0}s_+^{(i)^T}H_i s_+^{(i)} \leq \frac{-\lambda_{\min}[H_i]}{2\lambda_{\max}[H_i]}s_+^{(i)^T}H_i s_+^{(i)} \leq \frac{-\lambda_{\min}[H_i]}{2}\|s_+^{(i)}\|^2.$$

  where the last inequality uses $\lambda_{\max}[H_i] \geq \frac{s_+^{(i)^T}H_i s_+^{(i)}}{\|s_+^{(i)}\|^2} > 0$ and $\lambda_{\min}[H_i] \leq -\lambda_c$.

Following (2.30) and $T_i[s_+^{(i)}]^3 \leq L_H\|s_+^{(i)}\|$, we have

$$\begin{aligned}\mathcal{E} &\geq \left[\frac{\alpha_1^+}{2}(\eta-1) + \frac{1}{4}\right]\lambda_{\min}[H_i]\|s_+^{(i)}\|^2 - \frac{L_H}{6}\|s_+^{(i)}\|^3 + \frac{\sigma}{4}d\|s_+^{(i)}\|^4. \\ &\geq \left[\frac{\alpha_1^+}{2}(1-\eta) - \frac{1}{4}\right](-\lambda_{\min}[H_i])\|s_+^{(i)}\|^2 - \frac{L_H}{6}\|s_+^{(i)}\|^3 + \frac{\sigma}{4}d\|s_+^{(i)}\|^4.\end{aligned}$$

Using $\lambda_{\min}[H_i] \leq -\lambda_c$, $\mathcal{E} \geq \frac{1}{4}\left[2\alpha_1^+(1-\eta)-1\right]\lambda_c\|s_+^{(i)}\|^2 - \frac{L_H}{6}\|s_+^{(i)}\|^3 + \frac{\sigma}{4}d\|s_+^{(i)}\|^4$. Using $\alpha_1^+ \geq \frac{2}{3}(1-\eta)^{-1}$, we arrive at $\frac{\alpha_1^+}{2}(1-\eta) - \frac{1}{4} \geq \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$,

$$\mathcal{E} \geq \Bigg[\underbrace{\frac{\lambda_c}{12}}_{>0} \underbrace{- \frac{L_H}{6}}_{<0}\|s_+^{(i)}\| + \underbrace{\frac{\sigma}{4}d}_{>0}\|s_+^{(i)}\|^2\Bigg]\|s_+^{(i)}\|^2 =: \amalg_2(\|s_+^{(i)}\|). \tag{2.32}$$

Using $d > \frac{L_H^2}{3\sigma\lambda_c}$, the discriminant of $\amalg_2(\|s_+^{(i)}\|)$, $\Delta = \frac{L_H^2}{36} - \frac{\lambda_c}{12}\sigma d < 0$. Therefore, we deduce that $\mathcal{E} \geq 0$.

In both cases, we have $\mathcal{E} \geq 0$. Since the minimum of $M(s^{(i)}, s)$ is not attained at $s = 0$, we have $0 < M(s^{(i)}, s_+^{(i)}) - f_i \leq \eta\left[M_{\alpha^+}(s^{(i)}, s_+^{(i)}) - f_i\right]$ and, therefore, (QQR Ratio Test) is satisfied.

Result 2: To prove the second result, we assume by contradiction that $\|s^{(i)}\| \leq r_c$, but $\|s^{(i+1)}\| = \|s^{(i)} + s_+^{(i)}\| > r_c$. Using Result 1), Lemma 2.3.3, we have

$$M_{\alpha^+}(s^{(i)}, s_+^{(i)}) - f_i \geq \eta\left[M(s^{(i)}, s_+^{(i)}) - f_i\right] > \eta\left[m_3(s^{(i)} + s_+^{(i)}) - f_0\right] > 0. \tag{2.33}$$

We arrive at $M_{\alpha^+}(s^{(i)}, s_+^{(i)}) - f_i > 0$ which is a contradiction to $s_+^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_+(s^{(i)}, s)$. □

Remark 2.3.10. *Note that $\alpha_1^+ = 1$ in Theorem 2.3.4 is allowed. This is because we choose $\eta \in (0, \frac{1}{3})$ and therefore then $\frac{2}{3} \leq \frac{2}{3}(1-\eta)^{-1} \leq 1$. The bound in (2.28) is well defined.*

Remark 2.3.11. *Choose $\alpha_1^+ = \max\left\{\frac{2}{3}(1-\eta)^{-1}, 1 - \frac{|\lambda_{\min}[H_i]|}{|2\lambda_{\max}[H_i]|}\right\}$, in all locally nonconvex iterations, α_1^+ has iteration independent upper bound, $\alpha_1^+ \leq 1 - \frac{\lambda_c}{2L_g}$.*

Remark 2.3.12. *Regarding the parameters in the QQR model, we note the following:*

- *$\lambda_c > 0$ is a user-selected fixed positive constant. Usually, we set $\lambda_c > \epsilon^{1/3}$ where ϵ is the first order optimality tolerance. The purpose of λ_c is to distinguish between the three cases. In Case 2, we can set $\tilde{p} > 0$ independently of λ_c.*
- *From (2.20), (2.21), (2.28), we observe that in locally strictly convex iterations, we have $\alpha_1^+ > 1$; in locally nearly convex iterations, we have $\alpha_1^+ = 1$, whereas, in locally nonconvex iterations, we have $0 < \alpha_1^+ \leq 1$. This observation explains the choice of α_1^+ in Algorithm 2.3.1.*

Remark 2.3.13. *$M_\pm$ in Theorems 2.3.3 and 2.3.4 preserves local convexity. Note that $\nabla^2 M_\pm(s^{(i)}, s) = \alpha_1^\pm H_i + \frac{\sigma}{4}\alpha_2^\pm(\|s\|^2 I_n + 2ss^T)$ where $\alpha_1^\pm > 0$. If m_3 exhibits local convexity with $H_i \succeq 0$, then the bound $M_\pm(s^{(i)}, s)$ also shows the same local convexity behaviour as $s \to \boldsymbol{0}$. The same applies for the locally nonconvex case, $\lambda_{\min}[H_i] \leq 0$. This observation explains Example 2.2.1.*

Remark 2.3.14. *If we start the `QQR-v2` Algorithm with $\|s^{(0)}\| \leq r_c$ and choose $(a, d, \tilde{p})$ using (2.21), (2.20), and (2.28) in each iteration, then Result 1 of Theorems 2.3.3 and 2.3.4 indicates that (QQR Ratio Test) is always satisfied. Additionally, Result 2 of Theorems 2.3.3 and 2.3.4 shows that the iterates are bounded, meaning that the next iterates satisfy $\|s^{(i)}\| \leq r_c$.*

With appropriate parameter selections, the bound in Theorems 2.3.3 and 2.3.4 to (2.6) in `QQR-v1`. Corollary 2.3.4 provides a direct connection between the bounds derived in Theorems 2.3.3 and 2.3.4 and the specific models of Nesterov and `QQR-v1`.

Corollary 2.3.4. *(Link to `QQR-v1` Model) Let the `QQR-v1` model be defined as in (2.6) with $\bar{\kappa}, \bar{\iota}$ and $\tilde{p}$ as specified in Remark 2.3.1. Assume that $\|s^{(i)}\| < r_c$ and denote $K_1 := \frac{1}{2} - \frac{1}{6\tau}$ and $K_2 := 1 - 2(1-\hat{q})\tau$, which are both positive constants. If $\mu > 0$ is sufficiently small, satisfying $\mu \leq \mu_*$, where*

$$\mu_* = \min\left\{4^{-1/|\bar{\iota}|}K_1^{1/\bar{\iota}}, \left(\frac{K_2}{2}\right)^{1/\bar{\kappa}}, \left(\frac{9\max\{\lambda_c, \tilde{p}\}\sigma}{2L_H^2}K_2\right)^{1/\bar{\kappa}}\right\}. \tag{2.34}$$

Then, $M_\mu(s^{(i)}, s)$ satisfies (2.20)–(2.21) in Theorem 2.3.3 for locally strictly convex iterations and locally nearly convex iterations. $M_\mu(s^{(i)}, s)$ satisfies (2.28) in Theorem 2.3.4 for locally nonconvex iterations. Therefore, the results in Theorem 2.3.3 and Theorem 2.3.4 apply for `QQR-v1`.

Proof. Let α_1^μ and α_2^μ be defined as in `QQR-v1` model in (2.6). We denote

$$a = \alpha_1^\mu - 1 = 2\mu^{-\bar{\iota}}K_1 - 1, \qquad d = \alpha_2^\mu - 1 = \mu^{-\bar{\kappa}}K_2 - 1 \tag{2.35}$$

where $K_1 = \frac{1}{2} - \frac{1}{6\tau}$ and $K_2 = 1 - 2(1-\hat{q})\tau$. In **locally strictly convex iterations**, since $\bar{\iota} > 0$, and $\mu \leq (\frac{K_1}{4})^{1/\bar{\iota}} \leq (\frac{4}{3}K_1)^{1/\bar{\iota}}$ gives $a = 2\mu^{-\bar{\iota}}K_1 - 1 \geq \frac{1}{2} > 0$. We have $a > 0$ which satisfies the requirement for a in local strictly convex iterations in Theorem 2.3.3. The requirement for d in Theorem 2.3.3 is also satisfied since $\mu \leq \left(\frac{K_2}{2}\right)^{1/\bar{\kappa}}$ and $\bar{\kappa} > 0$, then $d = \mu^{-\bar{\kappa}}K_2 - 1 \geq \frac{1}{2}\mu^{-\bar{\kappa}}K_2 > 0$. Thus, we deduce that

$$ad = \left|2\mu^{-\bar{\iota}}K_1 - 1\right|\left[\mu^{-\bar{\kappa}}K_2 - 1\right] \geq \frac{1}{4}\mu^{-\bar{\kappa}}K_2 > \frac{L_H^2}{18\lambda_c\sigma} \tag{2.36}$$

where the last inequality uses the third term in the lower bound of (2.34). **Locally nearly convex iterations** follows similarly if we replace ad by d and $\frac{L_H^2}{18\lambda_c\sigma}$ by $\frac{L_H^2}{18\tilde{p}\sigma}$ in (2.36).

In locally nonconvex iterations, we have $\alpha_1^\mu = 1$ satisfying (2.28) in Theorem 2.3.4. Also, $\alpha_2^\mu = \mu^{-\bar{\kappa}}\big(1 - 2(1-\hat{q})\tau\big) \geq \frac{2L_H^2}{3\lambda_c\sigma} \geq \eta(1 + \frac{L_H^2}{3\lambda_c\sigma})$ satisfies (2.28) in Theorem 2.3.4. Therefore, in all cases conditions (2.20) and (2.21) are satisfied for Theorem 2.3.3. □

Table 2.2: An illustration of the bound in (2.37), (2.43), (2.53) in three convexity cases for different $\lambda_c = \epsilon^\omega$ with $\omega \in [0, \frac{1}{2}]$.

Order of Bounds	$\lambda_{\min}[H_k] \geq \lambda_c = \epsilon^\omega$	$\lvert\lambda_{\min}[H_k]\rvert \leq \lambda_c = \epsilon^\omega$	$\lambda_{\min}[H_k] < -\lambda_c = -\epsilon^\omega$
$\omega = 0$	$\mathcal{O}(\epsilon)$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$	$\mathcal{O}(1)$
$\omega = \frac{1}{4}$	$\mathcal{O}(\epsilon^{\frac{5}{4}})$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$	$\mathcal{O}(\epsilon^{\frac{3}{4}})$
$\omega = \frac{1}{3}$	$\mathcal{O}(\epsilon^{\frac{4}{3}})$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$	$\mathcal{O}(\epsilon)$
$\omega = \frac{1}{2}$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$	$\mathcal{O}(\epsilon^{\frac{3}{2}})$

2.3.3 Complexity and Global Convergence of QQR

This section presents a detailed analysis and convergence proofs for the QQR method. The upper bound M_{α^+} in `QQR-v2` (or M_μ in `QQR-v1`) plays a crucial role in ensuring the convergence of the QQR algorithm. For simplicity in notation, we will use M_{α^+} to represent the general upper bound in the proofs throughout the subsequent sections. The results apply similarly to M_μ. For proving the theoretical results of `QQR-v2`, we employ the upper bound M_{α^+} throughout this section. Nonetheless, in practical implementations, we minimize M_α, which is a linear combination of the upper bound M_{α^+} and the lower bound M_- (see Section 2.5.2).

We prove that by iteratively minimizing the upper bound, QQR is guaranteed to converge the local minimum of the quartically-regularised cubic polynomials. Moreover, in cases where the local minimum of m_3 is not reached, the function value $m_3(s^{(i)})$ is strictly decreasing with i at each iteration. By imposing mild constraints on the parameters $(a, \tilde{p})$, we demonstrate that the QQR method achieves lower complexity than the cubic regularization method in some cases. The complexity bound for QQR depends on the first-order optimality tolerance ϵ and on a user-chosen threshold parameter $\lambda_c \in (0, 1)$, which is fixed a priori and used to distinguish locally strictly convex iterations from the remaining cases. More precisely, iterations satisfying $\lambda_{\min}(H_i) \geq \lambda_c$are treated as locally strictly convex, while those with $\lambda_{\min}(H_i) < \lambda_c$ are handled separately. Below, we summarize the complexity bounds for the QQR algorithm. The constants $\check{c}_1$ and $\hat{c}_1$ are independent of the iteration number.

- **Locally strictly convex case ($\lambda_{\min}[H_i] \geq \lambda_c$)** , the error reduction is $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_3\lambda_c\epsilon$, or we have a linear convergence in gradient norm.
- **Locally nonconvex case ($\lambda_{\min}[H_i] \leq -\lambda_c$)** , the error reduction is $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \max\left\{\check{c}_1\epsilon\lambda_c^{3/2}, \hat{c}_1\lambda_c^3\right\}$ or we have a constant error reduction.
- **Locally nearly convex case ($\lvert\lambda_{\min}[H_i]\rvert \leq \lambda_c$)** , the error reduction is $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_4\epsilon^{\frac{3}{2}}$ or we have a linear convergence in gradient norm.

We illustrate how the choice of $\lambda_c = \epsilon^\omega$ with $\omega \in [0, \frac{1}{2}]$ affects the complexity bounds in each case. Typically, selecting a larger value of $\lambda_c = \mathcal{O}(1)$ would increase the region of the locally nearly convex case, which has a complexity bound of $\mathcal{O}(\epsilon^{\frac{3}{2}})$. However, this choice provides a better complexity bound for the locally strictly convex and locally nonconvex cases. Conversely, choosing a smaller value of $\lambda_c = \mathcal{O}(\epsilon^{1/2})$ results in a smaller region for the locally nearly convex case (i.e., $\lvert\lambda_{\min}[H_k]\rvert \leq \epsilon^{1/2}$) but worsens the complexity bound for the locally strictly convex and locally nonconvex cases.

As mentioned in Algorithm 2.3.1, we typically set $\lambda_c := \epsilon^{1/3}$, but note that other values of $\lambda_c = \epsilon^\omega$ with $\omega \in [0, \frac{1}{2}]$ are permissible. We provide the complexity bounds of the QQR algorithm for $\lambda_c := \epsilon^{1/3}$.

Summary of Complexity Bounds for QQR

In iteration i, assuming that the minimum of m_3 has not been found, i.e., $\|g_i\| > \epsilon$, the QQR algorithm satisfies the following improvements per iteration:

1. **Locally strictly convex case ($\lambda_{\min}[H_i] \geq \lambda_c$):** $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \mathcal{O}(\epsilon^{\frac{4}{3}})$ or we have a linear convergence in gradient norm.

2. **Locally nonconvex case ($\lambda_{\min}[H_i] \leq -\lambda_c$):** $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \mathcal{O}(\epsilon)$ or we have a constant error reduction.
3. **Locally nearly convex case ($|\lambda_{\min}[H_i]| < \lambda_c$):** $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \mathcal{O}(\epsilon^{\frac{3}{2}})$ or we have a linear convergence in gradient norm.

Based on these results, we give the first-order complexity for the QQR method.

Theorem 2.3.5. *Let m_3^{low} be the lower bound for $m_3(s)$. Then, there exist positive constants $\hat{c}_1$, $\hat{c}_3$, and $\hat{c}_4$ (see Theorems 2.3.6, 2.3.7, 2.3.8) such that, for any $\epsilon \in (0, 1]$, the QQR method (Algorithm 2.3.1 with M_{α^+} chosen as in Theorems 2.3.3 and 2.3.4 with $a \in (0, 1)$) requires at most*

$$\#_{QQR} = \max\left\{\hat{c}_1, \hat{c}_4, \hat{c}_3\right\} \frac{m_3(s^{(0)}) - m_3^{low}}{\min\{\epsilon^{\frac{4}{3}}, \epsilon, \epsilon^{\frac{3}{2}}\}} = \mathcal{O}(\epsilon^{-\frac{3}{2}})$$

evaluations of m_3 to produce an iterates s_ϵ such that $\|\nabla m_3(s_\epsilon)\| \leq \epsilon$.

Although the overall complexity of the QQR method is $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ evaluations, it only represents the worst-case bound among the three cases. We will prove in Corollary 2.3.5 that the $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ case only occurs in a small region where $-\epsilon^{1/3} \leq \lambda_{\min}(H_i) \leq \epsilon^{1/3}$ and $\epsilon \leq \|g_{i+1}\| \leq \epsilon^{4/5}$. In our numerical implementation, we have also devised refinement steps to accelerate the convergence to the local minimum when we are close to a local minimum with $\epsilon \leq \|g_i\| \leq \epsilon^{4/5}$. In the other cases, we achieve complexity bounds of order $\mathcal{O}(\epsilon^{-1})$ or $\mathcal{O}(\epsilon^{-\frac{4}{3}})$ evaluations, which are better worst-case bounds than those of the cubic regularization method.

2.3.3.1 Complexity Bound for Iterations with $\lambda_{\min}[H_i] \leq -\lambda_c$

In this subsection, we give proof of the complexity results for the locally nonconvex iterations.

Theorem 2.3.6. *Let $s^{(i)} \in \mathbb{R}^n$ be such that $\|g_i\| > \epsilon$ and $\lambda_{\min}[H_i] < -\lambda_c$ with $\lambda_c > 0$. We construct $M_{\alpha^+}(s^{(i)}, s)$ as defined in Theorem 2.3.4. $s^{(i)}$ is updated by the QQR algorithm. Assume that $\|g_{i+1}\| = \|\nabla m_3(s^{(i+1)})\| > \epsilon$. Then, $m_3(s^{(i)})$ is strictly decreasing with i. We have either a constant error reduction $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \check{c}_2$ or*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta \max\left\{\check{c}_1 \epsilon \lambda_c^{3/2}, \hat{c}_1 \lambda_c^3\right\}. \tag{2.37}$$

Here, $\hat{c}_1$, $\check{c}_1$ and $\check{c}_2$ are iteration-independent, λ_c-independent constants and ϵ-independent constants.

Proof. Let $g_v = \frac{g_i}{\|g_i\|}$ be the unit gradient vector. Let u be the unit eigenvector (i.e., $\|u\| = 1$) corresponding to $\lambda_{\min}[H_i]$ with the sign of u chosen such that $u^T g_v \leq 0$. Set $c_1 > 0, c_2 > 0$ as any positive scalars and set s_1 as the linear combination of g_v and u, such that $s_1 := c_2 u - c_1 g_v$. Substituting s_1 into M_{α^+}, we have

$$\begin{aligned} M_{\alpha^+}(s^{(i)}, s_1) &= f_i + g_i{}^T(c_2 u - c_1 g_v) + \frac{\alpha_1^+}{2}(c_2 u - c_1 g_v)^T H_i (c_2 u - c_1 g_v) + \frac{\sigma}{4}\alpha_2^+ \|c_2 u - c_1 g_v\|^4 \\ &\leq f_i + c_2 \underbrace{g_i^T u}_{\leq 0} - c_1 \|g_i\| + \frac{\alpha_1^+}{2}\left[c_1^2 g_v^T H_i g_v + c_2^2 u^T H_i u - 2c_1 c_2 g_v^T H_i u\right] + \frac{\sigma}{4}\alpha_2^+ (c_1 + c_2)^4 \end{aligned}$$

where the last term comes from $\|c_2 u - c_1 g_v\|^4 \leq (c_2\|u\| + c_1\|g_v\|)^4 \leq (c_1 + c_2)^4$. Using $c_1 > 0, c_2 > 0$,

$$u^T H_i u = \lambda_{\min}[H_i] \leq -\lambda_c, \qquad \text{and} \qquad -g_v^T H_i u = -\underbrace{\lambda_{\min}[H_i]}_{\leq 0}\underbrace{g_v^T u}_{\leq 0} \leq 0,$$

we obtain $M_{\alpha^+}(s^{(i)}, s_1) \leq f_i - c_1\|g_i\| + \frac{\alpha_1^+}{2}\left[c_1^2 g_v^T H_i g_v - c_2^2 \lambda_c\right] + \frac{\sigma}{4}\alpha_2^+(c_1 + c_2)^4$. Using $f_i = m_3(s^{(i)})$ and $g_v^T H_i g_v \leq L_g$ in Corollary 2.3.2, we deduce that

$$m_3(s^{(i)}) - M_+(s^{(i)}, s_1) \geq c_1\|g_i\| - \frac{\alpha_1^+}{2}(c_1^2 L_g - c_2^2 \lambda_c) - \frac{\sigma}{4}\alpha_2^+(c_1 + c_2)^4. \tag{2.38}$$

Also, for any c_1, c_2, the inequality $m_3(s^{(i+1)}) = \min_{s\in\mathbb{R}^n} M_{\alpha^+}(s^{(i)}, s) < M_+(s^{(i)}, s_1)$ holds. From Theorem 2.3.4, we deduce that

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta\Big[m_3(s^{(i)}) - M_+(s^{(i)}, s_+^{(i)})\Big] \geq \eta\Big[m_3(s^{(i)}) - M_+(s^{(i)}, s_1)\Big] \tag{2.39}$$

where $s_1 := c_2 u - c_1 g_v$. Substituting (2.38) into (2.39) gives

$$\eta^{-1}\Big[m_3(s^{(i)}) - m_3(s^{(i+1)})\Big] \quad \geq \quad c_1\|g_i\| - \frac{\alpha_1^+}{2}\big(c_1^2 L_g - c_2^2\lambda_c\big) - \frac{\sigma}{4}\alpha_2^+(c_1+c_2)^4.$$

Let $c_1 := \left[\frac{\alpha_1^+ L_g}{\sigma\alpha_2^+\left[1+\sqrt{2L_g{\lambda_c}^{-1}}\right]^4}\right]^{1/2} > 0$ and $c_2 := \sqrt{2L_g{\lambda_c}^{-1}}c_1 > 0$, then

$$\begin{aligned}\eta^{-1}\Big[m_3(s^{(i)}) - m_3(s^{(i+1)})\Big] &\geq c_1\|g_i\| + \frac{\alpha_1^+ L_g}{2}c_1^2 - \frac{\sigma\alpha_2^+}{4}\left[1+\sqrt{2L_g{\lambda_c}^{-1}}\right]^4 c_1^4,\\ &\underset{\|g_i\|>\epsilon}{\geq} \epsilon c_1 + \frac{\alpha_1^+ L_g}{4}c_1^2 \geq \max\left\{\epsilon c_1, \frac{\frac{2}{3}(1-\eta)^{-1}L_g}{4}c_1^2\right\}.\end{aligned} \tag{2.40}$$

The first inequality comes from substituting c_2, the second inequality comes from substituting c_1^2 in the last term and the last inequality comes from the fact that both ϵc_1 and $\frac{\alpha_1^+ L_g}{4}c_1^2$ are positive and $\alpha_1^+ \geq \frac{2}{3}(1-\eta)^{-1}$. Therefore, $\eta^{-1}\big[m_3(s^{(i)}) - m_3(s^{(i+1)})\big]$ only needs to be greater than any one of the terms.

- If $0 < \lambda_c < \min\{\frac{L_H^2}{3\sigma}, 2L_g\} =: \tilde{\lambda}$, then, according to (2.28), we can set $d = \frac{L_H^2}{\sigma\lambda_c}$ and $\alpha_2^+ = \eta^{-1}(1+d) < 2\eta^{-1}d \leq \frac{2L_H^2}{\sigma\lambda_c\eta}$. Also, we have $1+\sqrt{2L_g\lambda_c}^{-1} < 2\sqrt{2L_g\lambda_c}^{-1}$. Consequently,

$$c_1 \geq \left[\frac{\frac{2}{3}(1-\eta)^{-1}L_g}{\sigma(\frac{2L_H^2}{\sigma\lambda_c\eta})\left(2\sqrt{2L_g\lambda_c}^{-1}\right)^4}\right]^{1/2} = \check{c}_1\lambda_c^{3/2}$$

  where $\check{c}_1$ depends in η, L_g, L_H and σ, but is iteration independent and ϵ independent. Substituting into (2.40), we arrive at

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta\max\left\{\epsilon\check{c}_1\lambda_c^{3/2}, \frac{\frac{2}{3}(1-\eta)^{-1}}{4}\check{c}_1^2\lambda_c^3\right\}.$$

- If $\lambda_c > \min\{\frac{L_H^2}{3\sigma}, 2L_g\} =: \tilde{\lambda} > 0$, $c_1 \geq \left[\frac{\frac{2}{3}(1-\eta)^{-1}L_g}{\sigma(\frac{2L_H^2}{\sigma\lambda_c\eta})\left(1+\sqrt{2L_g\tilde{\lambda}^{-1}}\right)^4}\right]^{1/2} =: \tilde{c}_1$ which is a constant independent of iteration and λ_c. We arrive at the constant error reduction $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta\max\left\{\epsilon\tilde{c}_1, \frac{\frac{2}{3}(1-\eta)^{-1}L_g}{4}\tilde{c}_1^2\right\} := \check{c}_2$.

□

Remark 2.3.15. *Using Theorem 2.3.6, if assume that $\lambda_c = \epsilon^\omega$ with $\omega \geq 0$, then either there is constant error reduction or*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \max\left\{\check{c}_1\epsilon^{1+\frac{3}{2}\omega}, \hat{c}_1\epsilon^{3\omega}\right\} \geq \mathcal{O}(\epsilon^{\min\{3\omega, 1+\frac{3}{2}\omega\}}). \tag{2.41}$$

For our choice of $\lambda_c \geq \epsilon^{\frac{1}{3}}$, (2.41) gives $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_1\epsilon$.

2.3.3.2 Complexity Bound for Iterations with $\lambda_{\min}[H_i] \geq \lambda_c$

In this subsection, we give proof of the complexity results for the locally strictly convex iterations. To attain the complexity result stated at the beginning of Section 2.3.3, an additional condition is required when selecting the parameter a. Specifically, we need $\epsilon^{1/2} \leq a\lambda_c \leq \|g_i\|^{1/2}$ (or $\epsilon^{1/2} \leq \tilde{p} \leq \|g^{(i)}\|^{1/2}$).

Theorem 2.3.7. *In iterations with $\|g_i\| > \epsilon$ and $\lambda_{\min}[H_i] \geq \lambda_c$ with $\lambda_c > 0$, choose a such that*

$$\epsilon^{1/2} \leq a\lambda_c \leq \|g_i\|^{1/2} \tag{2.42}$$

and construct $M_{\alpha^+}(s^{(i)}, s)$ as defined in Theorem 2.3.3. The step $s^{(i)}$ is updated by the QQR algorithm. Assume that $\|g_{i+1}\| = \|\nabla m_3(s^{(i+1)})\| > \epsilon$. Then,

1. *Either there is a linear rate of decrease in gradient norm, such that $\|g_{i+1}\| \leq \rho_l \|g_i\|$ where ρ_l is a fixed constant in $(0, 1)$.*
2. *Or the step size has a lower bound, $\|s_+^{(i)}\| \geq c_3\epsilon^{1/2}$. Also, $m_3(s^{(i)})$ is strictly decreasing with i and satisfying*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_3\eta\lambda_c\epsilon. \tag{2.43}$$

Here, c_3 and $\hat{c}_3$ are iteration-independent, λ_c-independent and ϵ-independent constants.

Proof. **1) Linear Convergence in $\|g_i\|$**: If $\|g_{i+1}\| \leq \rho_l\|g_i\|$, we have a linear rate of decrease in $\|g_i\|$. **2) Function Value Decrease**: In the hard case, we have $\|g_{i+1}\| > \rho_l\|g_i\|$. We write

$$g_{i+1} = \nabla M(s^{(i)}, s_+^{(i)}) = \nabla M(s^{(i)}, s_+^{(i)}) - \nabla M_+(s^{(i)}, s_+^{(i)}) + \nabla M_+(s^{(i)}, s_+^{(i)}).$$

By the first-order optimality conditions of M_+, $\nabla M_+(s^{(i)}, s_+^{(i)}) = 0$. Thus,

$$-g_{i+1} = \nabla M_+(s^{(i)}, s_+^{(i)}) - \nabla M(s^{(i)}, s_+^{(i)}) = aH_i s_+^{(i)} - \frac{1}{2}T_i[s_+^{(i)}]^2 + d\sigma\|s_+^{(i)}\|^2 s_+^{(i)}. \tag{2.44}$$

where $d = \frac{L_H^2}{18a\lambda_c}$ (Theorem 2.3.3). Using triangle inequality, we obtain that

$$\epsilon < \|g_{i+1}\| \leq a\lambda_c\|s_+^{(i)}\| + \frac{L_H}{2}\|s_+^{(i)}\|^2 + a^{-1}\lambda_c^{-1}\tilde{d}\|s_+^{(i)}\|^3 \tag{2.45}$$

where $\tilde{d} := \frac{L_H^2}{18}$ is an iteration-independent constant and $d = a^{-1}\tilde{d}$. At least one term in the right hand side of (2.45) needs to be greater than $\frac{1}{3}\|g_{i+1}\|$. Therefore, one of the following inequalities is true

$$\|s_+^{(i)}\| \geq \frac{\|g_{i+1}\|}{3a\lambda_c} \geq \frac{1}{3}\|g_{i+1}\|^{1/2}\frac{\|g_{i+1}\|^{1/2}}{\|g_i\|^{1/2}} \underset{\|g_{i+1}\|>\rho_l\|g_i\|}{\geq} \frac{1}{3}\rho_l^{1/2}\|g_{i+1}\|^{1/2}, \tag{2.46}$$

$$\text{or },\|s_+^{(i)}\| \geq \left[\frac{2\|g_{i+1}\|}{3L_H}\right]^{1/2} \geq \left(\frac{2}{3L_H}\right)^{1/2}\|g_{i+1}\|^{1/2}, \tag{2.47}$$

$$\text{or },\|s_+^{(i)}\| \geq \left[\frac{a\lambda_c\|g_{i+1}\|}{3\tilde{d}}\right]^{1/3} \underset{(2.42)}{\geq} \left[\frac{\epsilon^{1/2}\|g_{i+1}\|}{3\tilde{d}}\right]^{1/3} \geq (3\tilde{d})^{-1/3}\epsilon^{1/6}\|g_{i+1}\|^{1/3}. \tag{2.48}$$

By setting $c_3 := \min\left\{\frac{1}{3}\rho_l^{1/2}, \left(\frac{2}{3L_H}\right)^{1/2}, \left(\frac{L_H^2}{6}\right)^{-1/3}\right\}$, using $\|g_{i+1}\| > \epsilon$, we proved the lower bound for step size $\|s_+^{(i)}\| \geq c_3\epsilon^{1/2}$. Note that c_3 is an iteration independent and λ_c independent constant. Using (2.42) gives $a > 0$, $M(s^{(i)}, s) \leq M_+(s^{(i)}, s)$ in Theorem 2.3.3 and Remark 2.3.9, we deduce that

$$\eta^{-1}\left[m_3(s^{(i)}) - m_3(s^{(i+1)})\right] \geq m_3(s^{(i)}) - M_+(s^{(i)}, s_+^{(i)}) \tag{2.49}$$

$$= -g_i^T s_+^{(i)^T} - \frac{\alpha_1^+}{2}s_+^{(i)^T}H_i s_+^{(i)} - \frac{\alpha_2^+}{4}\|s_+^{(i)}\|^4. \tag{2.50}$$

The first-order optimality conditions of M_+ gives $-g_i = \alpha_1^+ H_i s_+^{(i)} + \alpha_2^+ \sigma \|s_+^{(i)}\|^2 s_+^{(i)}$. Consequently,

$$\eta^{-1}\left[m_3(s^{(i)}) - m_3(s^{(i+1)})\right] \geq \frac{\alpha_1^+}{2} s_+^{(i)T} H_i s_+^{(i)} + \frac{3\alpha_2^+}{4}\sigma\|s_+^{(i)}\|^4$$
$$\geq \frac{\alpha_1^+}{2}\lambda_c\|s_+^{(i)}\|^2 \geq \frac{\lambda_c}{2}\|s_+^{(i)}\|^2 \tag{2.51}$$

where we use $\alpha_1^+ = 1 + a > 1$, $\alpha_2^+ \geq 0$ and $\lambda_{\min}[H_i] \geq \lambda_c > 0$. Finally, using $\|s_+^{(i)}\| \geq c_3\epsilon^{1/2}$, we obtain

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\lambda_c}{2} c_3^2 \epsilon = \lambda_c \hat{c}_3 \epsilon$$

where $\hat{c}_3 := \frac{1}{2}c_3^2 = \frac{1}{2}\min\left\{\frac{\rho_l}{9}, \frac{2}{3L_H}, (\frac{L_H^2}{6})^{-2/3}\right\}$ is an iteration-independent, λ_c iteration-independent and ϵ-independent constant. □

Remark 2.3.16. *Since $\|g_i\| \geq \epsilon$, the bound for a in (2.42) is well-defined.*

Remark 2.3.17. *Using Theorem 2.3.8, if assume that $\lambda_c = \epsilon^\omega$ with $\omega \geq 0$, then either there is a linear rate of decrease in gradient norm or*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_3\epsilon^{1+\omega}. \tag{2.52}$$

For our choice of $\lambda_c \geq \epsilon^{\frac{1}{3}}$, (2.52) gives $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_3\epsilon^{\frac{4}{3}}$.

Remark 2.3.18. *(Extension to Locally Nearly Convex Case) In locally nearly convex iterations, if we impose a similar condition on $\tilde{p}$ as stated in (2.42), specifically, $\epsilon^{1/2} \leq \tilde{p} \leq \|g^{(i)}\|^{1/2}$, then under the iteration where $H_i + \tilde{p}I_n \succ 0$, the analysis presented in (2.44)–(2.51) follows in a similar manner.*

2.3.3.3 Overall Complexity Bound

We have calculated complexity bounds for locally strictly convex and nonconvex iterations in the cases where $\lambda_c > \epsilon^{1/3}$ (as discussed in Remark 2.3.15 and Remark 2.3.17). The only case that is left to address is when $0 \leq \lambda_c \leq \epsilon^{1/3}$. To address this, Theorem 2.3.8 will demonstrate that the function value reduction in this scenario is at least $\mathcal{O}(\epsilon^{\frac{3}{2}})$. Importantly, Theorem 2.3.8 is universally applicable for all $\lambda_c > 0$ and all convexity cases. Theorem 2.3.8 establishes that the worst-case function value decrease achieved by the QQR method is at least as good as that observed in locally convex iterations.

Theorem 2.3.8. *In locally strictly convex or nonconvex iterations with $\|g_i\| > \epsilon$ and $|\lambda_{\min}[H_i]| \geq \lambda_c$, choose a such that*

$$\epsilon^{1/2} \leq |a\lambda_c| \leq \|g_i\|^{1/2}.$$

In locally nearly convex iterations with $\|g_i\| > \epsilon$ and $|\lambda_{\min}[H_i]| \leq \lambda_c$, choose $(a, \tilde{p})$ such that

$$\epsilon^{1/2} \leq \tilde{p} \leq \|g_i\|^{1/2}.$$

We construct $M_{\alpha^+}(s^{(i)}, s)$ as defined in Theorems 2.3.3 and 2.3.4 and update $s^{(i)}$ by the QQR algorithm. Assume that $\|g_{i+1}\| = \|\nabla m_3(s^{(i+1)})\| > \epsilon$; then, either $\|g_{i+1}\| \leq \rho_l\|g_i\|$ where ρ_l is a fixed constant in $(0, 1)$. Or, $\|s_+^{(i)}\| \geq c_3\epsilon^{1/2}$, and $m_3(s^{(i)})$ is strictly decreasing with i and satisfies

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_4\epsilon^{2/3}\|g_{i+1}\|^{5/6} \geq \hat{c}_4\epsilon^{\frac{3}{2}}. \tag{2.53}$$

where c_3 and $\hat{c}_4$ are iteration-independent, λ_c iteration-independent and ϵ-independent constants.

Proof. We give the proof for the case of $\lambda_{\min}[H_i] \geq \lambda_c$, and the proof for locally nearly convex cases follows similarly if we replace $a\lambda_c$ by $\tilde{p}$. This proof is applicable for all $\lambda_c > 0$.

If $\|g_{i+1}\| \leq \rho_l\|g_i\|$, we have a linear rate of decrease in $\|g_i\|$. Otherwise, if $\|g_{i+1}\| \geq \rho_l\|g_i\|$, by a similar deduction as in (2.44)–(2.48), we arrive at the bound for $\|s_+^{(i)}\|$ in (2.46)–(2.48). According to Theorem 8.2.8 [44],

$$-g_i = \alpha_1^+ H_i s_+^{(i)} + \alpha_2^+\sigma\|s_+^{(i)}\|^2 s_+^{(i)}, \qquad \alpha_1^+ H_i \succeq -\alpha_2^+\sigma\|s_+^{(i)}\|^2 I_n.$$

Using these optimality conditions, from (2.50), we deduce that

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha_1^+}{2} {s_+^{(i)}}^T H_i s_+^{(i)} + \frac{3\alpha_2^+}{4}\sigma\|s_+^{(i)}\|^4 \geq \frac{\alpha_2^+}{4}\sigma\|s_+^{(i)}\|^4 \geq \frac{L_H^2}{72}\|g_i\|^{-1/2}\|s_+^{(i)}\|^4 \quad (2.54)$$

where the last inequality comes from $\alpha_2^+ = 1 + d > d = \frac{L_H^2}{18a\lambda_c}$, and $\alpha_2^+\sigma \geq \frac{L_H^2}{18|a|\lambda_c} \geq \frac{L_H^2}{18}\|g_i\|^{-1/2}$. Substituting (2.46)–(2.48) into (2.54),

$$\begin{aligned} m_3(s^{(i)}) - m_3(s^{(i+1)}) &\geq \frac{L_H^2}{72}\min\left\{\frac{\rho_l^2}{3}\frac{\|g_{i+1}\|^2}{\|g_i\|^{1/2}}, \frac{4}{9L_H^2}\frac{\|g_{i+1}\|^2}{\|g_i\|^{1/2}}, \left(\frac{L_H^2}{6}\right)^{-4/3}\epsilon^{2/3}\frac{\|g_{i+1}\|^{4/3}}{\|g_i\|^{1/2}}\right\} \\ &\underset{\|g_{i+1}\|>\rho_l\|g_i\|}{\geq} \min\left\{\frac{L_H^2\rho_l^{5/2}}{216}\|g_{i+1}\|^{3/2}, \frac{\rho_l^{1/2}}{162}\|g_{i+1}\|^{3/2}, \frac{\rho_l^{1/2}}{6L_H^{3/2}}\epsilon^{2/3}\|g_{i+1}\|^{5/6}\right\} \\ &\hat{c}_4\epsilon^{2/3}\|g_{i+1}\|^{5/6} \geq \hat{c}_4\epsilon^{\frac{3}{2}} \end{aligned}$$

where $\hat{c}_4 := \min\left\{\frac{L_H^2\rho_l^{5/2}}{216}, \frac{\rho_l^{1/2}}{162}, \frac{\rho_l^{1/2}}{6L_H^{3/2}}\right\}$ is an iteration independent and λ_c independent constant. □

Our complexity proofs for Theorem 2.3.7 and Theorem 2.3.8 follow a similar structure as the complexity proof for ARC. These results were first obtained by Nesterov and Polyak [146, Thm 3.3.5]. Other works can be found in [23; 53]. To conduct the complexity analysis, we employ a typical construction that involves demonstrating lower bounds on the model decrease and the length of the step at each iteration. The connection between the gradient of model decrease and the gradient of function value decrease is established using (2.51). While our proof assumes exact minimization of M_{α^+} such that $\nabla M_+(s^{(i)}, s_+^{(i)}) = 0$, the same order of complexity bounds as in Theorem 2.3.7 and in Theorem 2.3.8 can be derived when considering approximate minimization with a step-termination condition, such as $\|\nabla M_+(s_+^{(i)})\| \leq \Theta\|s_+^{(i)}\|^3$ for some $\Theta \in (0, 1)$. This step-termination condition is also utilized in [53], and alternative variants exist [46; 88]. The proof incorporates standard techniques found in [23, Lemma 2.3] and in [53, Lemma 3.3], which demonstrate that the step cannot be arbitrarily small compared to the criticality conditions. We delegate the use of approximate termination conditions in the subproblem solution to future work and current numerical results.

Corollary 2.3.5. *Using Theorem 2.3.6–2.3.7, let $\lambda_c = \epsilon^{1/3}$. Then, only in the small region where $-\epsilon^{1/3} \leq \lambda_{\min}(H_i) \leq \epsilon^{1/3}$ **and** $\epsilon \leq \|g_{i+1}\| \leq \epsilon^{4/5}$, there is a linear rate of decrease in gradient norm, or*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_4\epsilon^{\frac{3}{2}}.$$

Otherwise, we have a linear rate of decrease in gradient norm, constant error reduction or $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \min\{\hat{c}_1, \hat{c}_3, \hat{c}_4\}\epsilon^{\frac{4}{3}}$. Constants are defined in Theorem 2.3.6–2.3.7 and are iteration-independent, λ_c iteration-independent and ϵ-independent constants.

Proof. For $\lambda_c = \epsilon^{1/3}$, if $|\lambda_{\min}(H_i)| > \lambda_c \geq \epsilon^{1/3}$ according to Remark 2.3.15 and Remark 2.3.17, we have error reduction of at least $\mathcal{O}(\epsilon)$ (or constant decrease) for the locally nonconvex case and $\mathcal{O}(\epsilon^{\frac{4}{3}})$ (or linear decrease in gradient) for locally strictly convex case. If $\|g_{i+1}\| \geq \epsilon^{4/5}$, then according to Theorem 2.3.8, we have either linear rate linear decrease in gradient or $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \hat{c}_4\epsilon^{\frac{2}{3}}\epsilon^{\frac{4}{5}\frac{5}{6}} \geq \hat{c}_4\epsilon^{\frac{4}{3}}$. □

In the implementation of the QQR algorithm, when we are close to a local minimum with $\epsilon \leq \|g_i\| \leq \epsilon^{4/5}$, we often employ a refinement step to accelerate the convergence to the local minimum. Specifically, we set a small value for the quartically regularization parameter and relax the control over the regularization to enhance asymptotic convergence. As a result, in our numerical experiments, we rarely encounter the worst-case scenario where the convergence rate is $\mathcal{O}(\epsilon^{\frac{3}{2}})$. A more comprehensive discussion on the implementation of QQR and the refinement step can be found in Section 2.5.

To conclude this section, it is important to highlight that there may be several alternative techniques for constructing the quadratic quartic local models within the context of the QQR method. For instance, as illustrated in [142], we can use the gradient information, i.e., the (local) gradient

Lipschitz constant L_g, to obtain the local upper/lower bounds for $m_3(s)$ at $s = s^{(i)}$. Let $\tau_\pm > 0$, then the gradient-based bounds are

$$M_{\alpha^\pm}(s^{(i)}, s) = m_3(s^{(i)}) + g_i^T s + s^T \left[\left(\frac{1}{2} - \frac{1}{3\tau_\pm} \right) \nabla^2 m_3(s^{(i)}) \pm \frac{L_g}{3\tau_\pm} I_n \right] s + (1 \pm 2\tau_\pm) \frac{\sigma}{4} \|s\|^4$$

where $\tau_+ > 1$ and $0 < \tau_- < \frac{1}{2}$. The gradient-based upper bounds also have a comparable structure to Nesterov's method. By selecting specific values for $\tau_\pm$, the gradient-based upper bounds can be simplified to match the expression in Nesterov's method. To avoid complicating the presentation of our general QQR method, we will not delve into the specifics of a gradient-bound-based QQR method. Instead, we focus on creating a QQR method that exploits (local) second and third-order smoothness, adapting the parameters automatically during the iterations.

2.4 Bounded Minimizer and Lipschitz Constants

In this section, we prove the existence of a uniform upper bound, r_c, such that for $s_* = \operatorname{argmin}_{s \in \mathbb{R}^n} m_3(s)$, we have $\|s_*\| \leq r_c$. We discuss local Lipschitz constants and provide technical lemmas to prove Corollary 2.3.2.

2.4.1 Proof of Lemma 2.3.3

Proof. We define a function $\hat{K}(r) : \mathbb{R}^+ \to \mathbb{R}$, $\hat{K}(r) := \frac{\|g\|}{r^3} + \frac{\lambda_0}{2r^2} + \frac{\Lambda_0}{6r}$. $\hat{K}(r)$ is a monotonically decreasing function with $r > 0$. $\hat{K}(r) \to +\infty$ as $r \to 0$ and $\hat{K}(r) \to 0$ as $r \to +\infty$. Thus, we can find a unique constant $r_c > 0$ that satisfies $\frac{\sigma}{4} = \frac{\|g\|}{r_c^3} + \frac{\lambda_0}{2r_c^2} + \frac{\Lambda_0}{6r_c}$. Therefore, we re-write $m_3(s)$ as, $m_3(s) = f_0 + g^T s + \frac{1}{2} s^T H s + \frac{1}{6} T[s]^3 + \left[\frac{\|g\|}{r_c^3} + \frac{\lambda_0}{2r_c^2} + \frac{\Lambda_0}{6r_c} \right] \|s\|^4$. Let $\xi_s := \frac{\|s_*\|}{r_c}$. Assume for contradiction the global minimum of $m_3(s)$ happens at $\|s_*\| > r_c$ (namely $\xi_{s_*} > 1$). Then,

$$m_3(s_*) = f_0 + \underbrace{\left[g^T s_* + \xi_s^3 \|g\| \|s_*\| \right]}_{>0} + \frac{1}{2} \underbrace{s_*{}^T \left[H + \lambda_0 \xi_s^3 I_n \right] s_*}_{>0} + \frac{1}{6} \underbrace{\left[T[s_*]^3 + \Lambda_0 \xi_s^3 \|s_*\|^3 \right]}_{>0}. \tag{2.55}$$

Thus, $m_3(s_*) > f_0$. This contradicts the hypothesis that s_* is the global minimum. □

2.4.2 Local Lipschitz Constants

We prove the technical Lemma for Corollary 2.3.2.

Lemma 2.4.1. *For any $s \in \mathbb{R}^n$, $L_H(s)$ satisfies*

$$L_H(s) := \max_{u \in \mathbb{R}^n, \|u\|=1} \nabla^3 m_3(s)[u]^3 \leq \Lambda_0 + 6\sigma \|s\| \tag{2.56}$$

Similarly,

$$L_g(s) := \max_{u \in \mathbb{R}^n, \|u\|=1} u^T \nabla^2 m_3(s) u \leq \max \left\{ \lambda_{\max}(H) + \frac{1}{2} \Lambda_0 \|s\| + 3\sigma \|s\|^2, 0 \right\} \tag{2.57}$$

where Λ_0 given in Definition 2.3.2.

Proof. For any $s = [s_1, \ldots, s_n]^T \in \mathbb{R}^n$, the third order derivative is $\nabla^3 m_3(s) = T + \frac{\sigma}{4} \nabla^3(\|s\|^4)$ where $\nabla^3(\|s\|^4) \in \mathbb{R}^{n \times n \times n}$ is a symmetric tensor with

$$\text{the } j\text{th face of } \nabla^3(\|s\|^4) = 8(s e_j^T + e_j s^T + s_j I) \tag{2.58}$$

for $1 \leq j \leq n$. $e_j \in \mathbb{R}^n$ is the unit vector with one on the jth entry and zero otherwise and $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. We first prove the bound for $L_H(s)$. For all $u \in \mathbb{R}^n$ with $\|u\| = 1$,

$$\nabla^3(\|s\|^4)[u]^3 = 8 \sum_{j=1}^n u_j u^T \left[s e_j^T + e_j s^T + s_j I_n \right] u = 8 \sum_{j=1}^n (2u_j^2 u^T s + u_j s_j \|u\|^2) = 24 u^T s.$$

Thus, $\nabla^3 m_3(s)[u^3] = T[u^3] + \frac{\sigma}{4}\nabla^3(\|s\|^4) \leq \Lambda_0 + 6\sigma u^T s$. Using Cauchy-Schwarz inequality $u^T s \leq \|s\|\|u\| \leq \|s\|$ gives (2.56). Similarly for $L_g(s)$, we have

$$u^T \nabla^2 m_3(s) u = u^T H u + \frac{1}{2} T[s][u]^2 + \sigma \left[\|s\|^2 + 2(u^T s)^2 \right].$$

Therefore, by Cauchy-Schwarz inequality and triangle inequality $\max_{u \in \mathbb{R}^n, \|u\|=1} \left| u^T \nabla^2 m_3(s) u \right| \leq \lambda_{\max}(H) + \frac{1}{2}\Lambda_0 \|s\| + 3\sigma \|s\|^2$. □

Remark 2.4.1. *(Alternative Definitions of Lipschitz Constant) The third order Taylor model of $m_3(s)$ at $s = s^{(i)}$ is $T_3(s^{(i)}, s) = f_i + g_i^T s + \frac{1}{2} s^T H_i s + \frac{1}{6} T_i[s]^3$. Note that $T_3(0, s) = f_0 + g^T s + \frac{1}{2} s^T H s + \frac{1}{6} T[s]^3$. Let $\hat{L}_H(s^{(i)})$ be the Lipschitz constant of $\nabla^2 T_3(s)$, such that*

$$\left\| \nabla^2 T_3(s^{(i)}, s_1) - \nabla^2 T_3(s^{(i)}, s_2) \right\|_{[2]} = \left\| T_i(s_1 - s_2) \right\|_{[2]} \leq \hat{L}_H(s^{(i)}) \|s_1 - s_2\|$$

for all $s_1, s_2 \in \mathbb{R}^n$. $\|\cdot\|_{[p]}$ is the tensor norm recursively induced by the Euclidean norm on the space of pth-order tensors $\|\mathcal{T}\|_{[p]} := \max_{\|v_1\|=\ldots=\|v_p\|=1} \left| \mathcal{T}[v_1, \ldots, v_p] \right|$. Then,

$$\hat{L}_H(s^{(i)}) = \left\| T_i \right\|_{[3]} = \max_{v \in \mathbb{R}^n, \|v_1\|=\ldots=\|v_3\|=1} \left| T_i[v_1][v_2][v_3] \right|.$$

$L_H(s^{(i)})$ is the maximum absolute value of $T_i[v_1][v_2][v_3]$ by applying any three unit vectors. But in the complexity and convergence proof, we can the tensor norm for T_i which is the maximum absolute value of $T_i[u]^3$ by applying any same unit vectors three times. For any $s^{(i)} \in \mathbb{R}^n$, we define

$$L_H(s^{(i)}) := \left\| T_i \right\|_{[3]^*} = \max_{u \in \mathbb{R}^n, \|u\|=1} T_i[u]^3.$$

Note that $L_H(s^{(i)}) \leq \hat{L}_H(s^{(i)})$ for all $s^{(i)} \in \mathbb{R}^n$. Note that we can choose the sign of $\pm u \in \mathbb{R}^n$ such that $T_i[u]^3 > 0$. Therefore, by definition, $L_H(s^{(i)}) \geq 0$ for all $s^{(i)} \in \mathbb{R}^n$. Since T_i is symmetric, the symmetric multilinear forms gives the equivalent diagonal characterization $\|T_i\|_{[3]} = \max_{\|u\|=1} |T_i[u, u, u]|$. Thus we may equivalently define $L_H(s^{(i)}) := \max_{\|u\|=1} |T_i[u]^3|$. Moreover, since the cubic form is odd, replacing u by $-u$ changes the sign of $T_i[u]^3$. Hence $L_H(s^{(i)}) = \max_{\|u\|=1} T_i[u]^3 \geq 0$. Consequently, $L_H(s^{(i)}) = \hat{L}_H(s^{(i)})$, so the two definitions give the same Lipschitz constant for the symmetric tensor T_i.

2.5 Implementations of the QQR Algorithmic Framework

In this section, we introduce two implementations of the QQR algorithmic framework, leveraging the theoretical results established in the previous section. The first implementation, which is built on `QQR-v1` in Section 2.3.1, involves the adaptive adjustment of a single parameter. The second implementation, built on `QQR-v2` in Section 2.3.2, incorporates the adaptive adjustment of two parameters. The numerical set-up and examples are provided in Section 2.5.3. It is important to note that we do not explicitly compute the (local) Lipschitz constants L_g or L_H that appear in the theoretical analysis. Instead, we employ adaptive parameter adjustment(s) strategies to achieve the desired theoretical bounds. Note that `QQR-v1` and `QQR-v2` (Algorithms 2.5.1–2.5.2) are constructed within the framework of Algorithm 2.3.1, with more specific parameter selections.

2.5.1 Implementation of Variant 1

Choice of $\hat{q}$: For each locally strictly convex iteration i, the set $\mathcal{D}^{(i)}(\hat{q})$ is defined relative to the current iterate $s^{(i)}$ and is therefore iteration-dependent. Theorem 2.3.2 shows that, provided the trial step is computed within the current region $\mathcal{D}^{(i)}(\hat{q})$ and the parameter μ satisfies (2.11), the decrease is locally linear with contraction factor $(1 - \mu)$. The parameter $\hat{q}$ controls the trade-off between the size of the certified region $\mathcal{D}^{(i)}(\hat{q})$ and the amount of local curvature information available from the cubic term. Smaller values of $\hat{q}$ impose a weaker curvature requirement and hence lead to a

larger admissible region. In particular, $\hat{q} = 0$ gives the largest region, but it does not provide any positive lower bound from the cubic curvature term. Consequently, the resulting upper model may require a more conservative regularization, and the corresponding contraction constant may be weaker. Conversely, increasing $\hat{q}$ strengthens the required local curvature condition. This gives more curvature information and can reduce the amount of regularization needed in M_μ, potentially allowing larger trial steps. However, the certified region $\mathcal{D}^{(i)}(\hat{q})$ then becomes smaller and may even fail to contain the relevant trial step if $\hat{q}$ is chosen too large. Thus, $\hat{q}$ should be chosen as an intermediate value balancing the size of the certified region and the strength of the curvature information. In our implementation and complexity discussion, we use the representative choice $\hat{q} = 2$. This choice is consistent with the positive-semidefinite condition appearing in the global optimality analysis, namely

$$H_i + T_i[s] + \sigma\|s\|^2 I_n \succeq 0,$$

while avoiding the overly restrictive choice $\hat{q}$ close to 3.

Choice of $\bar{\iota}$ and $\bar{\kappa}$: In the case of locally convex iterations, we set $\bar{\iota} = \bar{\kappa} = 1$, which ensures that M_μ represents a linear combination of the upper bound and the lower bound.

Rate of Linear Convergence τ_* under Convexity: The optimal linear rate is determined by the largest possible μ that satisfies the condition in (2.11). Substituting $\bar{\iota} = \bar{\kappa} = 1$ and $\hat{q} = 2$, we have $\mu_c = \max_{\tau\in(\frac{1}{3},1)} \min\left\{\frac{3\tau-1}{3\tau+1}, \frac{1-\frac{2}{3}\tau}{1+\frac{2}{3}\tau}\right\}$. This gives a unique optimal root, $\mu_c = \frac{1}{7}(11 - 6\sqrt{2}) \approx 0.359$ and consequently, the optimal (local) linear rate $\tau_* = \frac{\sqrt{2}}{2}$.

Remark 2.5.1. *Compared to Nesterov's methods,* `QQR-v1` *achieves a faster (local) linear rate of convergence. The contraction factor for Nesterov's methods is* $1-\mu_{\mathcal{N}} \approx 0.807$*, whereas the contraction factor for* `QQR-v1` *is* $1 - \mu_c \approx 0.641$.

Number of Unsuccessful Iterations: We typically initialise μ with $\mu = \mu_0$ and then adaptively decrease μ by updating $\mu := \eta_0\mu$, where $0 < \eta_0 < 1$. If the iteration is locally strictly convex, we set $\bar{\iota} = 1$. We have proven that choosing $\mu < \mu_c \approx 0.359$ ensures (local) linear convergence. Therefore, it takes at most $\lceil \log(\mu_c/\mu_0)/\log(\eta_0)\rceil + 1$ unsuccessful iterations to decrease μ below μ_c before we can accept the step. If the iteration is locally nonconvex, according to Corollary 2.3.4, we require $0 < \mu < \mu_*$. Therefore, it takes at most $\lceil \log(\mu_*/\mu_0)/\log(\eta_0)\rceil + 1$ unsuccessful iterations to decrease μ below μ_* before we can accept the step. If the iteration is nearly convex, then we add a perturbation to the second-order term by setting $\tilde{p} > 0$.

Algorithm 2.5.1: `QQR-v1` Algorithm

Initialization: $s^{(0)} = 0$, $i = 0$, $0 < \rho_1 < \frac{1}{3} < \rho_2$, $0 < \gamma_0 < 1 < \gamma_1$.
Set $\epsilon > 0$, usually $\epsilon = 10^{-5} \sim 10^{-8}$, $\lambda_c = \epsilon^{1/3}$.
Use the model in (2.6) (with initialization $\bar{\kappa} = \bar{\iota} = 1$, $\hat{q} = 2$, $\mu = \frac{1}{2}$, $\tau_* = \frac{\sqrt{2}}{2}$ and $\tilde{p} = 0$).
while $\|g_i\| > \epsilon$ **do**
  Compute $s_+^{(i)} := \operatorname{argmin}_{s\in\mathbb{R}^n} M_\mu(s^{(i)}, s)$ and $\rho(s_+^{(i)})$ as in (QQR Ratio Test).
  if $\rho(s_+^{(i)}) \geq \rho_1$, *Successful* **then**
    Accept the iterate. $s^{(i+1)} := s^{(i)} + s_+^{(i)}, i = i + 1, \tilde{p} = 0$.
    if $\rho(s_+^{(i)}) \geq \rho_2$ *Very Successful* **then**
      Increase μ by $\mu = \gamma_1\mu$
  if $\rho(s_+^{(i)}) < \rho_1$, *Unsuccessful* **then**
    Reject the iterate. Evaluate $\lambda^{(i)} = \lambda_{\min}[H_i]$.
  if $|\lambda^{(i)}| \leq \lambda_c$ **then**
    Add perturbation $\tilde{p} > 0$.
  else
    Decrease μ by $\mu = \gamma_0\mu$. If $\lambda^{(i)} < -\lambda_c$, set $\alpha_1^\mu := 1$.

2.5.2 Implementation of Variant 2

In the implementation of `QQR-v2`, we initialise M_α with $\alpha_1 = \alpha_2 = 1$ and $\tilde{p} = 0$. The complete algorithm for `QQR-v2` is provided in Algorithm 2.5.2. We note that both the smallest eigenvalue $\lambda_{\min}[H_i]$ and largest one $\lambda_{\max}[H_i]$ are needed in the running of the algorithm.

Algorithm 2.5.2: `QQR-v2` Algorithm

Initialization: $s^{(0)} = 0$, $i = 0$, $0 < \rho_1 < \frac{1}{3} < \rho_2$, , $0 < \eta_0 < 1 < \eta_1$, $\gamma_2 > 1$.
Set $\epsilon > 0$, usually $\epsilon = 10^{-5} \sim 10^{-8}$, $\lambda_c = \epsilon^{1/3}$.
Use the model in (2.18) (with initialization $\alpha_1 = \alpha_2 = 1$ and $\tilde{p} = 0$):
while $\|g_i\| > \epsilon$ **do**
  Compute $s_+^{(i)} := \operatorname{argmin}_{s \in \mathbb{R}^n} M_\alpha(s^{(i)}, s)$ and $\rho(s_+^{(i)})$ as in (QQR Ratio Test).
  if $\rho(s_+^{(i)}) \geq \rho_1$, *Successful* **then**
    Accept the iterate. $s^{(i+1)} := s^{(i)} + s_+^{(i)}, i = i + 1, \tilde{p} = 0$.
    if $\rho(s_+^{(i)}) \geq \rho_2$, *Very Successful* **then**
      Decrease α_2 by $\mu = \eta_0 \alpha_2$
  if $\rho(s_+^{(i)}) < \rho_1$, *Unsuccessful* **then**
    Reject the iterate. Evaluate $\lambda^{(i)} = \lambda_{\min}[H_i]$.
    if $|\lambda^{(i)}| \leq \lambda_c$ **then**
      Add perturbation $\tilde{p} > 0$.
    else
      Increase α_2 by $\alpha_2 := \eta_1 \alpha_2$, and
      If $\lambda^{(i)} < -\lambda_c$, set $\alpha_1 := \max\left\{\frac{2}{3}(1-\eta)^{-1}, 1 - \frac{|\lambda_{\min}[H_i]|}{2|\lambda_{\max}[H_i]}\right\}$,
      If $\lambda^{(i)} > \lambda_c$, increase $\alpha_1 := \gamma_2 \alpha_1$.

Remark 2.5.2. *(Other Variations of the Algorithm) Note that when $s^{(i)}$ is close to a local minimum of $m_3(s)$, setting the regularization parameter to $\min\left\{\frac{\sigma}{4}, \|g_i\|\right\}$ is a standard numerical technique [45] to speed up the asymptotic convergence in the convex region. As the iterates approach the local minimum, the coefficient of the quartic-order term tends to ϵ, allowing for more relaxed control over the regularization. Other numerical techniques that speed up asymptotic convergence, such as the Newton method, can also be applied.*

2.5.3 Numerical Set-up and Illustrations

We construct quartically-regularised cubic polynomials to test the QQR algorithm's performance. Specifically, we define $m_3(s) = f_0 + g^T s + \frac{1}{2} s^T H s + \frac{1}{6} T[s]^3 + \frac{\sigma}{4}\|s\|^4$, where $f_0 = 0$, and the coefficients g, H, and T are generated as follows:

$$g = \texttt{a*randn(n, 1)}, \quad H = \texttt{b*symm(randn(n, n))}, \quad T = \texttt{c*symm(randn(n, n, n))}. \tag{2.59}$$

g is a random n-dimensional vector multiplied by a scaling factor `a`; H is a random symmetric $n \times n$ matrix multiplied by `b`; and T is a random symmetric $n \times n \times n$ tensor multiplied by `c`. Here, `n` represents the dimension of the problem ($s \in \mathbb{R}^n$), and `symm(randn())` denotes a symmetric matrix or tensor whose entries follow a normal distribution with mean zero and variance one. The parameters `a`, `b`, `c`, and σ are chosen differently to test the algorithm's performance under various scenarios. We set the stopping criteria to be $\|g_i\| < \epsilon$ where $\epsilon = 10^{-5}$ unless otherwise specified. The *iteration counts* record the number of times we minimize the quadratic model with quartic regularization. The iteration counts include both successful and unsuccessful iterations. The *functions'/derivatives' evaluation counts* record the number of times we evaluate the function and derivative values. The evaluation counts only include the successful iterations.

According to Figures 2.2, `QQR-v1` performs competitively with Nesterov's method in terms of both evaluation counts. In locally strictly convex iterations (which start from the third evaluation count), there are two main factors contributing to the rate of `QQR-v1`. Firstly, as illustrated in Remark 2.5.1,

QQR-v1 has a contraction factor of 0.641, while Nesterov's method has a contraction factor of 0.807. In addition to QQR-v1 having a faster theoretical linear rate, the model we minimize in each step of QQR-v1 also incorporates a linear combination of (local) upper bound M_{α^+} and (local) lower bound M_-. In contrast, in Nesterov's method, the model we minimize in each step is only the (local) upper bound. In locally nonconvex iterations, neither method guarantees a linear rate of convergence. Among these methods, QQR-v2 requires the least number of function/derivative evaluation counts to find the local minimum of $m_3(s)$. The ARC method only adjusts the parameter in the regularization term, whereas QQR-v2 adjusts both the coefficients in the second-order term and the regularization term.

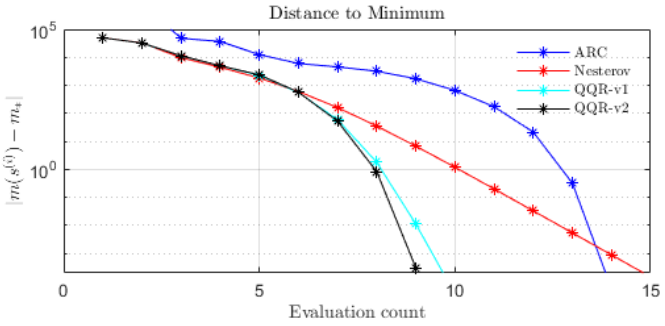


Figure 2.2: The parameters are a = 80, b = 80, c = 80, and $\sigma = 80$, $n = 100$.

2.6 Numerical Results: Comparison Across Methods

This section provides a comprehensive comparison of minimization algorithms for the AR3 sub-problem discussed in this chapter. These algorithms include the ARC method, Nesterov's method (for convex m_3 only), QQR-v1 method, and QQR-v2 method. The ARC method was proposed by Cartis, Gould, and Toint in [46], while Nesterov's method was introduced in [142]. QQR-v1 and QQR-v2 were proposed in this chapter.

To assess the robustness of the minimization solvers, we devised multiple sets of AR3 sub-problems across different dimensions, n, ranging from 5 to 100. These test sets comprise AR3 sub-problems that feature varying gradients, Hessians, tensors, and regularization terms. Specifically, we consider singular, ill-conditioned, concave or convex, diagonal, and indefinite Hessians, as well as large and small tensor terms, dense tensor terms, and ill-conditioned tensor terms.

Table 2.3 to Table 2.7 report the numerical results for these test sets. The coefficients in the test sets are generated as described in Section 2.5.3. The results presented in each table are obtained over 10 random problems randomized using MATLAB functions `rand()` or `randn()`. The *relative minimum values* of a method is obtained by dividing the minimum value achieved by this algorithm by the mean of all the minimum values obtained by all algorithms. In all of the test cases, none of the algorithms resulted in unsuccessful cases, as they all converged to a local minimum that satisfies $\|g_i\| < \epsilon$ and $\lambda_{\min}[H_i] > -\epsilon^{1/2}$ with $\epsilon = 10^{-5}$. The best-performing results for each problem set are highlighted.

The numerical results show that overall, our proposed methods, including QQR-v1 and QQR-v2, consistently perform competitively with the ARC method and Nesterov's method in terms of the number of evaluations of functions/derivatives or iterations required. Our proposed methods are robust enough to handle ill-conditioning in the Hessian term (Table 2.4) and perform well for the minimization of m_3 with singular, concave, diagonal, or indefinite Hessians (Table 2.3, 2.4). Additionally, QQR is effective in finding local minima for quartic polynomials with ill-conditioned tensor terms, diagonal tensor terms, and large and directional tensors Table 2.7. Last but not least, our proposed algorithm can locate the local minimum for a wide range of positive σ values (Table 2.5). This enables us to dynamically adjust σ while utilizing QQR for the AR3 method in minimizing the objective function $f(x)$. We provide some highlights specific to each method and each problem sets.

- Our proposed QQR-v1 method is a generalization of Nesterov's method and performs competitively with it in terms of function/derivative evaluations and iterations for all dimensions. A

notable finding is that `QQR-v1` performs competitively with Nesterov's method even for convex quartically-regularised cubic polynomials (Table 2.3). The reason behind this is that our implementation of `QQR-v1` uses a model that is a linear combination of both the local upper bound and lower bound, while Nesterov's method only utilizes the upper bound. When dealing with convex and locally strictly convex m_3, our method can converge to the local minimum with just a few iterations and function evaluations.

- Our proposed `QQR-v2` method generally requires the least function/derivative to locate the local minimum of m_3 among all methods. All methods have comparable CPU times, with `QQR-v2` requiring slightly less time than the other methods.
- QQR can minimize ill-conditioned problems (Table 2.4). Interestingly, in the ill-conditioned test case in Table 2.4, QQR locates the local minimum of m_3 with very few function/derivative evaluations. This can be attributed to the Hessian being diagonal and positive semi-definite, which results in faster convergence, as seen in the convex quartically-regularised cubic polynomial minimization (Table 2.3).
- Based on our numerical experiments, the nearly convex case happens very rarely in the QQR iterates, even for ill-conditioned Hessian terms or singular Hessian terms. This size of the perturbation term $\tilde{p}$ does not significantly affect the convergence behaviour in practice.

The selection of which method to use for minimizing the quartically regularised cubic polynomial is contingent on the intended objective of the optimization. If the goal is to satisfy the complexity requirements outlined in [53] by attaining a local minimum for the AR3 sub-problem, then `QQR-v2` is generally favored.

Table 2.3: **Convex m_3, Convex H or Concave H:** `a` $= 80$, and $\sigma = 80$. In the **first table**, m_3 is convex, $H =$ `symm(30*(randn(n)+neye(n)))` and `c` $= 1$; In the **second table**, m_3 is nonconvex but locally convex at $s = 0$, $H =$ `symm(30*(randn(n)+neye(n)))` and `c` $= 80$. In the **third table**, m_3 is concave, $H =$ `symm(30(randn(n)-n*eye(n)))` and `c` $= 80$. The CPU time is calculated as the average CPU time of the solver for solving all problems across all dimensions. $\rho_1 = 0.3, \rho_2 = 3, \eta_0 = 0.5, \eta_1 = 2, \gamma_0 = 0.8, \gamma_1 = 1.2, \gamma_2 = 1.1$ for Tables 2.3 to 2.7.

	Relative Min.			Iterations.			Evaluations.			CPU time
Method	$n = 5$	50	100	5	50	100	5	50	100	(s)
ARC	1	1	1	5	4	3	5	4	3	0.024
Nesterov	1	1	1	12.9	14	15	12.9	14	15	0.077
QQR-v1	1	1	1	6	7.3	7	6	7	7	0.049
QQR-v2	1	1	1	2.9	2	2	4	3	3	0.018
ARC	1	1	1	5	4	4	5	4	4	0.032
QQR-v1	1	1	1	6.5	7	7	6.5	7	7	0.054
QQR-v2	1	1	1	4	3	3	4	3	3	0.029
ARC	0.998	0.995	0.997	11.7	24.1	27.6	8	14.4	21.4	0.129
QQR-v1	1	1	1	9.7	22	23.4	9.6	20.7	22.6	0.145
QQR-v2	0.998	0.995	0.997	7.4	12.2	17	7.3	11.8	17	0.106

Table 2.4: **Ill-Conditioned H, Diagonal H or Singular H:** `a` $= 80$, `c` $= 80$ and $\sigma = 80$. H is a diagonal ill-conditioned matrix with diagonal entries uniformly distributed in $[0, 10^{10}]$.

	Relative Min.			Iterations.			Evaluations.			CPU time
Method	$n = 5$	50	100	5	50	100	5	50	100	(s)
ARC	1	1	1	5.3	6.5	5.3	5.3	6.2	5.3	0.042
QQR-v1	1	1	1	7.3	7.5	9.7	7	7.4	9.2	0.062
QQR-v2	1	1	1	4.1	4.8	4.7	4.1	4.8	4.5	0.037

Table 2.5: **Changing σ:** From top to bottom, a $= 80$, b $= 80$, c $= 80$ and $\sigma = 5, 300$, respectively. The experiments show the algorithm's ability to handle a range of σ values. Generally, smaller values of σ for $m_3(s)$ require more function/derivative evaluations for convergence across all algorithms.

	Relative Min.			Iterations.			Evaluations.			CPU time
Method	$n = 5$	50	100	5	50	100	5	50	100	(s)
ARC	1	1	1	11.8	24.8	28	8.1	15.2	20.5	0.12
QQR-v1	0.84	0.9958	0.897	31.9	19.2	37.9	11.3	18.5	26.7	0.19
QQR-v2	1	0.9958	0.9978	16.1	12.9	18.1	14.5	12.8	17.6	0.10
ARC	1	1	1	7	9.8	20.4	6.7	8.8	12.4	0.09
QQR-v1	1	1	1	7.6	11.3	14	7.6	10.6	13.4	0.10
QQR-v2	1	1	1	4.8	7	14	4.8	7	10.2	0.08

Table 2.6: **Small Tensor Term v.s. Large Tensor Term:** From top to bottom, a $= 80$, b $= 80$, $\sigma = 80$, and c $= 300, 10$ respectively.

	Relative Min.			Iterations.			Evaluations.			CPU time
Method	$n = 5$	50	100	5	50	100	5	50	100	(s)
ARC	1	1	0.979	9.5	19.3	26.9	8	15.7	19	0.128
QQR-v1	0.84	1	1	39.5	18.5	20.7	16.9	17.8	20.4	0.163
QQR-v2	1	1	1	12.8	12.5	16.9	11.8	12.5	16.7	0.123
ARC	1	1	1	7.1	11.4	15.1	6.3	7.9	9.9	0.0695
QQR-v1	1	1	1	6.8	8.6	10	6.7	8.6	9.8	0.0691
QQR-v2	1	1	1	4.2	5.7	6.6	4.2	5.7	6.6	0.0549

Table 2.7: **Ill-Conditioned Tensor or Positive and Directional Tensors:** a $= 80$, b $= 80$, and $\sigma = 80$. In the **first table**, T is a diagonal ill-conditioned tensor with diagonal entries uniformly distributed in $[10^{-10}, 10^{3}]$. In the **second table**, we consider positive and directional tensors, where T is a diagonal tensor with entries that are uniformly distributed in the range of $[0, 40]$.

	Relative Min.			Iterations.			Evaluations.			CPU time
Method	$n = 5$	50	100	5	50	100	5	50	100	(s)
ARC	1	1	1	9	11.7	15.1	7.9	9.6	11.6	0.077
QQR-v1	1	1	1	13.3	11.8	11.6	12.5	11.5	11.5	0.090
QQR-v2	1	1	1	7.6	8.9	8.4	7.3	8.5	8.4	0.071
ARC	1	1	1	7.1	8.5	14	6.2	6.5	8	0.0501
QQR-v1	1	1	1	6.9	8	8.1	6.8	7.5	7.5	0.0509
QQR-v2	1	1	1	4.1	4.4	4	4.1	4.4	4	0.0293

2.6.1 An Illustration of Minimizing a General Objective using the AR3 algorithm

We have conducted small-scale numerical testing to minimize a general objective function $f(x)$ using the AR3 algorithm with the subproblem as in (ARp Subproblem). Here, we present a few illustrative examples. After introducing the Cubic–Quadratic Regularization (CQR) model in Chapter 3 and the Diagonal Tensor Model (DTM) in in Chapter 4, we will provide a more comprehensive benchmark performance study on the Moré–Garbow–Hillstrom (MGH) test set, which is a classical collection of nonlinear optimization problems for benchmarking iterative algorithms. More numerical experiments—comparing QQR, DTM, and CQR against ARC and other state-of-the-art adaptive regularization methods—are presented in Section 3.4 and Section 4.5. The experiments in this section are therefore intended only as preliminary demonstrations.

In this subsection, we implement the `QQR-v2` methods for the minimization problem within the AR3 sub-problem at each step of the AR3 algorithm. We compare the performance of the ARC method, the AR3 method with the ARC inner sub-problem solver (referred to as `AR3 with ARC`), and the AR3 method with the QQR inner sub-problem solver (referred to as `AR3 with QQR`). The test

problems are selected from Moré, Garbow, and Hillstrom's Test Problem Set [129]. In Table 2.8, we provide preliminary numerical results where

- `ev.` and `it.` denote the number of successful outer iterations and the number of total outer iterations, respectively.
- `inner ev.` denotes the average number of successful inner iterations required to solve each sub-problem, while `inner it.` denotes the average number of total inner iterations required to solve each sub-problem.

In these examples, the `AR3` method requires slightly fewer outer functions and derivative evaluations. For inner iterations, `AR3 with QQR` usually takes fewer inner iterations and evaluations than `AR3 with ARC`, indicating that `QQR-v2` is a more efficient solver than ARC for the sub-problem. Note that the AR3 method generally requires more CPU time than the ARC method. The key reason for this is that for the problems we tested, the CPU time is dominated by numerical linear algebra costs rather than by the costs of function and derivative evaluations. AR3 methods may be more suitable for functions that are expensive to evaluate or are dominated by tensor directions.

Table 2.8: The first test function is adapted from [124], $f(x) = \frac{1}{2}x^T Ax - 5\sum_{i=1}^{n} i\sin(x_i) + \|x\|^4$, where $x = [x_1, \ldots, x_n] \in \mathbb{R}^n$ and $n = 5$. The matrix $A \in \mathbb{R}^{n\times n}$ and the vector $a \in \mathbb{R}^n$ are randomly generated with entries following a normal distribution $\mathcal{N}(0, 1)$. The next five test functions are taken from Moré, Garbow, and Hillstrom's Test Problem Set [129]. $\rho_1 = 0.3, \rho_2 = 0.9, \eta_0 = 0.5, \eta_1 = 2$ for both inner and outer solver. The tolerance for the outer and inner layers are 10^{-2} and 10^{-3} respectively.

Method	**ARC**		**AR3 + ARC**				**AR3 + QQR**			
Prob	# it.	# ev.	# it.	# ev.	# inner it.	# inner ev.	# it.	# ev.	# inner it.	# inner ev.
SIN	25	15	26	12	4.6	4.6	31	13	1.9	1.9
GAU	42	37	41	32	3.0	3.0	41	32	1.2	1.2
FRF	50	49	47	46	4.6	5.6	55	44	3.2	2.3
BES	38	38	37	36	10.1	2.8	41	39	10.8	3.0
HVF	527	527	526	525	2.3	2.0	483	482	1.2	1.1
BAR	9	9	10	9	31.4	2.8	10	9	30.8	10

In conclusion, this chapter presents the QQR method, a second-order method that aims to achieve high-accuracy minimization of nonconvex quartically regularized cubic polynomials, specifically the AR3 sub-problem. The QQR method is an iterative algorithm that operates by minimizing a sequence of quadratic polynomials with quartic regularization at the set of points $\{s^{(i)}\}_{i\geq 0}$. This method can be viewed as a nonconvex extension of Nesterov's method proposed in [142], which was originally designed for convex m_3 functions. Notably, when applied to convex m_3 functions, the QQR method exhibits the same linear convergence profile as Nesterov's method. For locally convex iterations, the QQR method achieves either a linear rate of convergence in gradient norm or an error reduction of at least $\mathcal{O}(\epsilon^{\frac{4}{3}})$ per iteration. In the case of locally nonconvex iterations, the QQR method provides an error reduction of at least $\mathcal{O}(\epsilon)$. The QQR method demonstrates superior complexity performance compared to the cubic regularization method in certain scenarios. The conducted numerical experiments validate the effectiveness of the QQR methods, as they perform competitively with state-of-the-art approaches such as ARC. In many instances, the QQR methods require fewer iterations or function evaluations and can find lower minima. Additionally, the QQR methods can handle both convex and nonconvex quartically-regularised cubic polynomials while effectively addressing challenges related to ill-conditioning in the Hessian term and the tensor term.

Chapter 3

Local Methods for the Regularized Third-order Subproblem II

3.1 Introduction

In the previous chapter, we introduced the Quadratic Quartic Regularisation (QQR) method by approximating the third-order tensor term of the AR3 model as a linear combination of quadratic and quartic terms, yielding (possibly nonconvex) local models that can be solved to global optimality. In this chapter, we aim to enhance the previous approach by explicitly incorporating (simple) third-order terms. Similarly, our target is to minimize (Quartically Regularized Cubic Model) iteratively. We propose the CQR algorithmic framework for minimizing a nonconvex Cubic multivariate polynomial with Quartic Regularisation, by sequentially minimizing a sequence of local quadratic models that also incorporate both simple cubic and quartic terms. The role of the cubic term is to crudely approximate local tensor information, while the quartic one provides model regularization and controls progress. We provide necessary and sufficient optimality conditions that fully characterise the global minimizers of these cubic-quartic models. We then turn these conditions into secular equations that can be solved efficiently using root-finding procedures. We show, using our optimality characterisations, that a CQR algorithmic variant has the optimal-order evaluation complexity of $\mathcal{O}(\epsilon^{-3/2})$ when applied to minimizing our quartically-regularized cubic subproblem, which can be further improved in special cases. We propose practical CQR variants that judiciously use local tensor information to construct the local cubic-quartic models. We test these variants numerically and observe them to be competitive with cubic regularization and other subproblem solvers on typical instances and even show superior performance on ill-conditioned subproblems with special structure.

The fourth-order Taylor expansion of $m_3(s^{(i)}+s)$ at $s^{(i)}$ is given in (2.1), denoted as

$$M(s^{(i)},s) := \underbrace{f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{1}{6}T_i[s]^3}_{:=T_3(s^{(i)},s)} + \frac{\sigma}{4}\|s\|^4,$$

where $T_3(s^{(i)},s)$ represents the third-order Taylor expansion of $m_3(s)$ at $s=s^{(i)}$. We have $M(s^{(i)},s) = m_3(s+s^{(i)})$, $\min_{s\in\mathbb{R}^n} m_3(s) = \min_{s\in\mathbb{R}^n} M(s^{(i)},s)$, and

$$s_* := \operatorname*{argmin}_{s\in\mathbb{R}^n} m_3(s) = s^{(i)} + \operatorname*{argmin}_{s\in\mathbb{R}^n} M(s^{(i)},s).$$

We employ a quadratic model with a cubic degree term and quartic regularization to provide an approximation for $M(s^{(i)},s)$. This model, which we refer to as the **Cubic Quartic Regularization (i.e., CQR) polynomial/model** can be expressed as follows

$$M_c(s^{(i)},s) = f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{\beta_i}{6}\|s\|_W^3 + \frac{\sigma_c^{(i)}}{4}\|s\|_W^4 \approx M(s^{(i)},s), \tag{3.1}$$

where $\sigma_c^{(i)} > 0$, $\beta^i \in \mathbb{R}$, W is a symmetric, positive-definite matrix[1]. In the CQR polynomial, β_i typically provides a simple approximation of the tensor term $T_i[s]^3$, while σ_c is adjusted to control

[1] Note that $\|v\|_W := \sqrt{v^T W v}$.

regularization and algorithmic progress. It is worth noting that our model allows negative values of β_i, which is particularly useful as it can yield information about negative tensor directions. When $\beta_i = 0$ and $W = I_n$, $M_c(s^{(i)}, s)$ simplifies to the type of quadratic quartically-regularized (QQR) polynomials in Chapter 2. On the other hand, when $\beta_i > 0$ and $\sigma_c^{(i)} = 0$, $M_c(s^{(i)}, s)$ reduces to the quadratic model with cubic regularization used in the ARC/AR2 algorithm [46; 70; 101; 125; 146]. However, note that in this chapter, we require $\sigma_c^{(i)} > 0$ but allow $\beta_i \leq 0$.

The reason for choosing the CQR polynomial $M_c(s^{(i)}, s)$ to approximate $M(s^{(i)}, s)$ is that in this chapter, we characterise the global minimizers of such CQR polynomials and devise a procedure to locate an approximate global minimizer of $M_c(s^{(i)}, s)$. Then, we minimize the CQR polynomial $M_c(s^{(i)}, s)$ iteratively and find a sequence $\{s^{(i)}\}_{i\geq 0}$ such that $s^{(i)}$ converges to a second-order local minimizer of $M(s^{(i)}, s)$ (or equivalently m_3). We refer to this iterative minimization method as **the CQR algorithmic framework/method**. The updates for the CQR method are characterised by the following iterations

$$s_c^{(i)} = \operatorname*{argmin}_{s\in\mathbb{R}^n} M_c(s^{(i)}, s), \qquad s^{(i+1)} = s^{(i)} + s_c^{(i)}, \qquad i := i + 1$$

where $\beta_i, \sigma_c^{(i)}$ are adjusted adaptively in each iteration.

The chapter presents the following key contributions:

- We develop necessary and sufficient optimality conditions for the global minimizers of $M_c(s^{(i)}, s)$. These conditions are general and applicable to both convex and nonconvex M_c and for both $\beta_i \geq 0$ and $\beta_i \leq 0$. Leveraging these optimality conditions, we can transform the problem of locating a global minimizer of $M_c(s^{(i)}, s)$ into a problem of solving a univariate nonlinear equation coupled with a matrix system that is similar to the trust region and ARC secular equations (Theorem 8.2.8 [44]). This opens up opportunities for developing efficient algorithms for minimizing $M_c(s^{(i)}, s)$, such as those based on Cholesky decompositions and Newton's method for root finding.

  In the literature, there are related works on cubic polynomial multivariate models. For instance, Martinez and Raydan [124; 125] investigate the problem of the separable cubic model, where the cubic power term is expressed as $\sum_{i=1}^n \beta\bar{a}_j s_j^3$, with s_j being the jth entry of $s \in \mathbb{R}^n$ and $\beta\bar{a}_j$ being scalar constants. However, these studies are conducted within the framework of cubic regularization or trust region methods [46; 47; 80; 83]. They do not incorporate third-order terms of the Taylor series in the objective function or its cubic approximations. The minimization of quartic regularized polynomials with specific cubic terms represents a relatively new research area. Nesterov [142] recently introduced a linearly convergent second-order method for minimizing convex quartically-regularized cubic polynomials. Also, Bullins [34] provides a descent method for efficiently minimizing similar polynomials, allowing a more general structure in the quartic regularization term. Yet, as far as we are aware, there are no algorithms explicitly designed for the minimization of nonconvex m_3 that use local models with simple cubic and quartic terms, whose global minimizers can be characterised and calculated.

- We then show global convergence properties of CQR algorithmic variants, analyzing the quality of the approximation provided by $M_c(s^{(i)}, s)$ to $m_3(s)$. Our theoretical analysis shows that CQR exhibits at least the same complexity as ARC. In specific cases, such as in the presence of separability, a variant of the CQR algorithm offers advantageous complexity and convergence properties compared to ARC. Practical CQR variants are proposed, where particular attention is given to the tensor approximation terms. Preliminary numerical experiments indicate that the CQR algorithm performs competitively compared to state-of-the-art methods such as the ARC method in terms of iterations or function evaluations. Specifically, for badly scaled problems and for problems where the gradient and tensor terms have similar orientation, the CQR algorithm requires far fewer iterations and evaluations than ARC or QQR to find an approximate stationary point for this set of subproblems.

The chapter is structured as follows: In Section 3.2, we explore necessary and sufficient optimality conditions that hold at a global minimizer of $M_c(s^{(i)}, s)$ and present an efficient algorithm in minimizing

the CQR polynomial. Section 3.3 contains the convergence proof and complexity analysis of the CQR method. In addition, Section A.1.1 proves that the iterates of the CQR method are uniformly bounded, a technical theorem crucial for analyzing the convergence and complexity of the CQR method. Finally, we provide algorithmic implementations and numerical examples in Section 3.4. Note that during the preparation of this dissertation, follow-up work has already emerged on the CQR model, most notably the recent work [190], which provides a tight SDP-based global solution framework that complements the theoretical optimality conditions developed for CQR.

Remark 3.1.1. *(On the distinction between $\beta\|s\|^3$ and $T[s]^3$) The radial cubic term $\beta\|s\|^3$ used in the CQR model and the genuine third-order Taylor term $T[s]^3$ are different. The Taylor term is odd along rays, $T[-s]^3 = -T[s]^3$, and hence asymmetric about the origin. By contrast, $\beta\|s\|^3$ is radially symmetric, $\beta\|-s\|^3 = \beta\|s\|^3$, and therefore cannot reproduce this directional asymmetry; it retains only a single scalar magnitude of third-order information. This is a trade-off made for tractability: the full tensor subproblem is hard to minimize globally, whereas the radial cubic term renders the subproblem amenable to a secular-equation solution and, as shown in subsequent work, to a* tight *semidefinite programming relaxation. Indeed, the recent follow-up of Zhou, Liu, Nie, and Tang [190] establishes an SDP relaxation for the CQR problem that is provably tight whenever $\|s^*\|\big(\beta+3\sigma\|s^*\|\big) \geq 0$ is satisfied, admits a globally tractable convex reformulation.*

3.2 Characterising Global Minima of the Cubic Quartic Regularization (CQR) Polynomial

In this section, we develop necessary and sufficient conditions that hold at the global minimizers of $M_c(s^{(i)}, s)$ in 3.1. We convert the problem of finding a global minimizer of $M_c(s^{(i)}, s)$ into a problem of solving a nonlinear eigenvalue problem, which we can then solve using root-finding techniques. For the sake of notational simplicity, in this section, we write $M_c(s^{(i)}, s)$ as $M_c(s)$,

$$M_c(s) = f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{\beta}{6}\|s\|_W^3 + \frac{\sigma_c}{4}\|s\|_W^4, \tag{3.2}$$

where σ_c represents $\sigma_c^{(i)}$ and β as β_i. The derivatives of $M_c(s)$ has the expression

$$\nabla M_c(s) = g_i + H_i s + \frac{\beta}{2}\|s\|_W(Ws) + \sigma_c\|s\|_W^2(Ws), \tag{3.3}$$

$$\nabla^2 M_c(s) = H_i + \frac{\beta}{2}\left(W\|s\|_W + \frac{(Ws)(Ws)^T}{\|s\|_W}\right) + \sigma_c\left(W\|s\|_W^2 + 2(Ws)(Ws)^T\right) \tag{3.4}$$

where ∇ is the derivative with respect to s and the matrix $(Ws)(Ws)^T$ is a rank-one matrix.

The section is organized as follows: In Section 3.2.1, we present global necessary optimality conditions for the CQR Polynomial in (3.2). In Section 3.2.2, we provide global sufficient optimality conditions. Finally, employing these conditions, in Section 3.2.3, we introduce an algorithm for locating an approximate minimizer of the CQR polynomial using nonlinear root-finding techniques.

3.2.1 Necessary Optimality Conditions for the CQR Polynomial

Theorem 3.2.1 provides a necessary condition for the global minimizers of $M_c(s)$, and we prove this theorem using a methodology similar to that of Theorem 8.2.7 in [44].

Theorem 3.2.1. ***(Necessary optimality conditions)*** *Let s_c be a global minimizer of $M_c(s)$ in (3.2) over $\mathbb{R}^n$ and let*

$$B_{\boldsymbol{CQR}}(s_c) := H_i + \frac{\beta}{2}W\|s_c\|_W + \sigma_c W\|s_c\|_W^2. \tag{3.5}$$

Then, s_c satisfies the system of equations

$$B_{\boldsymbol{CQR}}(s_c)s_c = -g_i \tag{3.6}$$

and

$$B_{\boldsymbol{CQR}}(s_c) \succeq 0. \tag{3.7}$$

In addition, if $B_{\boldsymbol{CQR}}(s_c) \succ 0$, then s_c is the unique global minimizer of M_c.

Proof. The first-order necessary local optimality conditions at s_c give $g_i + H_i s_c + \frac{\beta}{2}\|s_c\|_W(Ws_c) + \sigma_c\|s_c\|_W^2(Ws_c) = 0$ and hence that (3.6) holds. The second-order necessary optimality gives

$$w^T\left[H_i + \frac{\beta}{2}\left(W\|s_c\|_W + \frac{(Ws_c)(Ws_c)^T}{\|s_c\|_W}\right) + \sigma_c\left(W\|s_c\|_W^2 + 2(Ws_c)(Ws_c)^T\right)\right]w \geq 0 \tag{3.8}$$

for all vectors $w \in \mathbb{R}^n$. If $s_c = 0$, (3.8) is equivalent to H_i being positive semi-definite, which immediately gives the result in (3.7). Thus, we only need to consider $s_c \neq 0$. There are two cases to consider. Firstly, suppose that $w^T W s_c = 0$. In this case, it immediately follows from (3.8) that

$$w^T\Big[\underbrace{H_i + \frac{\beta}{2}\|s_c\|_W W + \sigma_c\|s_c\|_W^2 W}_{=B_{\mathbf{CQR}}(s_c)}\Big]w \geq 0 \text{ for all } w \text{ for which } w^T W s_c = 0. \tag{3.9}$$

It thus remains to consider vectors w for which $w^T W s_c \neq 0$. Since w and s_c are not orthogonal with respect to the W-norm, the line $s_c + \tilde{k}w$ intersects the ball centred at the origin of radius $\|s_c\|_W$ at two points, s_c and u_*, with

$$\|s_c\|_W = \|u_*\|_W. \tag{3.10}$$

We set $w_* = u_* - s_c$, and note that w_* is parallel to w. Since s_c is a global minimizer, we immediately have that

$$\begin{aligned} 0 &\leq M_c(u_*) - M_c(s_c) \\ &= g_i^T(u_* - s_c) + \frac{1}{2}{u_*}^T H_i u_* - \frac{1}{2}{s_c}^T H_i s_c + \frac{\beta}{6}(\|u_*\|_W^3 - \|s_c\|_W^3) + \frac{\sigma_c}{4}(\|u_*\|_W^4 - \|s_c\|_W^4) \\ &= g_i^T(u_* - s_c) + \frac{1}{2}{u_*}^T H_i u_* - \frac{1}{2}{s_c}^T H_i s_c \end{aligned}$$

where the last equality follows from (3.10). But (3.6) gives

$$(u_* - s_c)^T g_i = (s_c - u_*)^T H_i s_c + \frac{\beta}{2}\|s_c\|_W(s_c - u_*)^T W s_c + \sigma_c\|s_c\|_W^2(s_c - u_*)^T W s_c. \tag{3.11}$$

In addition, (3.10) gives

$$(s_c - u_*)^T W s_c = \frac{1}{2}{s_c}^T W s_c + \frac{1}{2}{u_*}^T W u_* - {u_*}^T W s_c = \frac{1}{2}{w_*}^T W w_* = \frac{1}{2}\|w_*\|_W^2. \tag{3.12}$$

Thus we have

$$\begin{aligned} 0 &\leq M_c(u_*) - M_c(s_c) \\ &\overset{(3.11)}{=} (s_c - u_*)^T H_i s_c + \frac{\beta}{2}\|s_c\|_W(s_c - u_*)^T W s_c + \sigma_c\|s_c\|_W^2(s_c - u_*)^T W s_c + \frac{1}{2}{u_*}^T H_i u_* - \frac{1}{2}{s_c}^T H_i s_c \\ &\overset{(3.12)}{=} (s_c - u_*)^T H_i s_c + \frac{\beta}{4}\|s_c\|_W\|w_*\|_W^2 + \frac{\sigma_c}{2}\|s_c\|_W^2\|w_*\|_W^2 + \frac{1}{2}{u_*}^T H_i u_* - \frac{1}{2}{s_c}^T H_i s_c \\ &= \frac{1}{2}{w_*}^T H_i w_* + \frac{\beta}{4}\|s_c\|_W\|w_*\|_W^2 + \frac{\sigma_c}{2}\|s_c\|_W^2\|w_*\|_W^2 = \frac{1}{2}{w_*}^T\left[H_i + \frac{\beta}{2}\|s_c\|W + \sigma_c\|s_c\|_W^2 W\right]w_*. \end{aligned}$$

We deduce that

$$w^T\left[H_i + \frac{\beta}{2}\|s_c\|_W W + \sigma_c\|s_c\|_W^2 W\right]w \geq 0 \text{ for all } w \text{ such that } w^T W s_c \neq 0. \tag{3.13}$$

Hence, (3.9) and (3.13) together show that $H_i + \frac{\beta}{2}\|s_c\|_W W + \sigma_c\|s_c\|_W^2 W \succeq 0$. The uniqueness of s_c when $H_i + \frac{\beta}{2}\|s_c\|_W W + \sigma_c\|s_c\|_W^2 W$ is positive definite follows immediately from (3.6).

□

3.2.2 Sufficient Optimality Conditions for the CQR Polynomial

In this section, we derive sufficient optimality conditions using two approaches. Specifically, in Section 3.2.2.1, we establish a sufficient optimality condition inspired by Nocedal and Wright [150, Thm 4.1], while in Section 3.2.2.2, inspired by Cartis, Gould, and Toint [44, Sec 8.2.1], we deduce another set of sufficient optimality conditions.

The former approach crucially applies to a range of β values that sometimes depend on the length of a global minimizer (which is typically not known a priori). The second approach relies on a secular equation, and hence, connects β to the left-most eigenvalue of the Hessian, which can often be computed numerically [62; 44]. Finally, in Section 3.2.2.3, we combine both theorems to provide a general sufficient optimality result (Theorem 3.2.5) that applies to all CQR polynomials.

3.2.2.1 Sufficient Optimality Conditions using Quadratic Functional Reformulations

In this section, we investigate sufficient conditions that hold at a global minimizer of $M_c(s)$, following a methodology inspired by Nocedal and Wright [150, Thm 4.1].

Theorem 3.2.2. ***(Sufficient optimality conditions using quadratic reformulations)*** *Let $B_{\boldsymbol{CQR}}(s_c)$ be defined as in (3.5). Then, s_c is a global minimizer of $M_c(s)$ in (3.2) over $\mathbb{R}^n$ if the following conditions are satisfied:*

1. *g is in the range of $B_{\boldsymbol{CQR}}(s_c)$, such that*

$$B_{\boldsymbol{CQR}}(s_c)s_c = \left(H_i + \frac{\beta}{2}W\|s_c\|_W I_n + \sigma_c W\|s_c\|_W^2 I_n\right)s_c = -g_i. \tag{3.14}$$

2. *$B_{\boldsymbol{CQR}}(s_c)$ defined in (3.5) is positive semidefinite*

$$B_{\boldsymbol{CQR}}(s_c) \succeq 0. \tag{3.15}$$

3. *β satisfies*

$$\beta \geq -3\sigma_c\|s_c\|_W, \qquad \textit{or equivalently} \qquad \|s_c\|_W \geq -\frac{\beta}{3\sigma_c}. \tag{3.16}$$

If, in addition to conditions (3.14)–(3.16), either $B_{\boldsymbol{CQR}}(s_c)$ is positive definite or $\beta > -3\sigma_c\|s_c\|_W$, then s_c is the unique global minimizer of M_c.

Proof. We assume wlog $f_i = 0$. For all vectors $w \in \mathbb{R}^n$, we have

$$\begin{aligned} M_c(s_c+w) &= g_i^T(s_c+w) + \frac{1}{2}H_i[s_c+w]^2 + \frac{\beta}{6}\|s_c+w\|_W^3 + \frac{\sigma_c}{4}\|s_c+w\|_W^4, \\ &= \underbrace{g_i^T(s_c+w) + \frac{1}{2}B_{\mathbf{CQR}}(s_c)[s_c+w]^2 - \frac{\beta}{12}\|s_c\|_W^3 - \frac{\sigma_c}{4}\|s_c\|_W^4}_{\mathcal{F}_1} \\ &\quad \underbrace{-\frac{1}{2}\left[\frac{\beta}{2}\|s_c\|_W + \sigma_c\|s_c\|_W^2\right]\|s_c+w\|_W^2 + \frac{\beta}{6}\|s_c+w\|_W^3 + \frac{\sigma_c}{4}\|s_c+w\|_W^4 + \frac{\beta}{12}\|s_c\|_W^3 + \frac{\sigma_c}{4}\|s_c\|_W^4}_{\mathcal{F}_2}. \end{aligned}$$

After rearranging,

$$\mathcal{F}_1 = \underbrace{\left[g_i^T s_c + \frac{1}{2}B_{\mathbf{CQR}}(s_c)[s_c]^2 - \frac{\beta}{12}\|s_c\|_W^3 - \frac{\sigma_c}{4}\|s_c\|_W^2\right]}_{=M_c(s_c)} + \underbrace{\left[g_i + B_{\mathbf{CQR}}(s_c)s_c\right]}_{=0 \text{ by } (3.14)}^T w + \underbrace{\frac{1}{2}B_{\mathbf{CQR}}(s_c)[w]^2}_{\geq 0 \text{ by } (3.15)}.$$

Therefore, we deduce that $\mathcal{F}_1 \geq M_c(s_c)$ for all $w \in \mathbb{R}^n$. Also, if $B_{\mathbf{CQR}}(s_c)$ is positive definite, then $\mathcal{F}_1 > M_c(s_c)$ for all $w \in \mathbb{R}^n \setminus \{0\}$.

On the other hand, we denote $E = \|s_c\|_W$ and $F = \|s_c + w\|_W$, clearly $E \geq 0$ and $F \geq 0$. We can simplify and rearrange $\mathcal{F}_2$ as

$$\mathcal{F}_2 = \frac{1}{2}(E - F)^2 \left[\frac{\beta}{6}(E + 2F) + \frac{\sigma_c}{2}(E + F)^2\right]. \tag{3.17}$$

Clearly, when $\beta \geq 0$, we have $\mathcal{F}_2 \geq 0$. Else, when $\beta \leq 0$, we let $\tilde{f}(F) := \frac{(E+F)^2}{(E+2F)}$ for $F \geq 0$; then $F_* := \operatorname{argmin}_{F \geq 0} \tilde{f}(F) = 0$ and $\tilde{f}(F_*) = E$. This implies that E is the best lower bound for $\tilde{f}(F)$ with $F \geq 0$. Together with condition (3.16), we have $0 \geq \beta \geq -3\sigma_c E \geq -3\sigma_c \frac{(E+F)^2}{(E+2F)}$. This immediately gives $\frac{\beta}{6}(E + 2F) + \frac{\sigma_c}{2}(E + F)^2 \geq 0$ and, consequently, $\mathcal{F}_2 \geq 0$. Note that if $\beta > -3\sigma_c\|s_c\|_W$, equality is attained only at $w = 0$.

Combining $\mathcal{F}_1 \geq 0$ and $\mathcal{F}_2 \geq 0$, we have $M_c(s_c + w) \geq M_c(s_c)$ for all $w \in \mathbb{R}^n$ and s_c is a global minimizer of M_c. If $B_{\mathbf{CQR}}(s_c)$ is positive definite or $\beta > -3\sigma_c\|s_c\|_W$, we have $M_c(s_c + w) > M_c(s_c)$ for all $w \in \mathbb{R}^n/\{0\}$.

□

The sufficiency conditions (3.14)–(3.15) coincide with the necessary conditions (3.6)–(3.7). The distinction between sufficiency and necessity arises from the inclusion of (3.16), which imposes certain constraints on the choice of β. Namely, β should not be selected to have an excessively negative value.

Theorem 3.2.2 can be considered a generalization of sufficient conditions for the minimization of a cubically regularized quadratic polynomial [46; 146]. If $\sigma_c = 0$ and $\beta = 2\sigma_q > 0$, we recover the sufficient condition for the cubically regularized quadratic polynomial, $f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{\sigma_q}{3}\|s\|_W^3$. The sufficiency condition (3.16) is naturally satisfied since $\beta > 0$. The sufficiency conditions (3.14) and (3.15) for the CQR polynomial reduce to the sufficiency conditions of the cubic regularized polynomial in [44, Thm 8.2.8].

3.2.2.2 Sufficient Optimality Conditions using a Secular Equation Approach

Theorem 3.2.3. ***(Sufficient optimality conditions using a secular equation approach)*** *Assume that H_i and β satisfy any one of the following cases,*

- ***Case 1:*** *H_i is indefinite.*
- ***Case 2:*** *H_i is positive definite and $\beta \geq 0$.*
- ***Case 3:*** *H_i is positive definite, $-4\sqrt{\sigma_c\lambda_1} < \beta \leq 0$, and $\left\|\left[H_i - \frac{\beta^2}{16\sigma_c}W\right]^{-1}g\right\|_W \leq -\frac{\beta}{4\sigma_c}$ where λ_1 is the leftmost eigenvalue of the pencil $(H_i; W)$.*

Let $s_c \in \mathbb{R}^n$ satisfy (3.6)–(3.7). Then, s_c is a global minimizer of $M_c(s)$ in (3.2) over $\mathbb{R}^n$.

Theorem 3.2.3 shares some similarities with Theorem 3.2.2. In particular, when $\beta \geq 0$, both proofs indicate that conditions (3.6)–(3.7) represent the necessary and sufficient condition for global optimality of s_c for M_c. When $\beta < 0$, (3.16) in Theorem 3.2.2 establishes the relationship between β and $\|s_c\|_W$ required to satisfy the sufficient optimality conditions. On the other hand, Case 3 in Theorem 3.2.3 provides the relationship between β and the eigenvalues of H_i to identify the global minimizer before computing $\|s_c\|_W$. A combined result of the two sufficiency theorems can be found in Section 3.2.2.3. In this subsection, we focus on proving Theorem 3.2.3.

To prove Theorem 3.2.3, we first need to introduce the **secular equation**. To satisfy the conditions (3.6)–(3.7), we seek a vector $s(\lambda)$ which satisfies the system:

$$B_{\mathbf{CQR}}(s_c)s(\lambda) = (H_i + \lambda W)s(\lambda) = -g_i, \tag{3.18}$$

$$B_{\mathbf{CQR}}(s_c) = H_i + \lambda W \succeq 0, \tag{3.19}$$

$$\lambda = \frac{\beta}{2}\|s(\lambda)\|_W + \sigma_c\|s(\lambda)\|_W^2. \tag{3.20}$$

where $B_{\mathbf{CQR}}(s_c)$ is defined in (3.5).

To rearrange (3.19), let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of the pencil $(H_i; W)$ and let $u_1, u_2, \ldots, u_n$ be the corresponding generalised eigenvectors where each $u_i \in \mathbb{R}^n$; see Appendix A.5 in [44]. Then,

$$H_i U = WUD, \text{ where } U^T W U = I_n, \tag{3.21}$$

where $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix with entries λ_i for $1 \leq i \leq n$, and $U \in \mathbb{R}^{n \times n}$ is the matrix whose columns are the u_i vectors for $1 \leq i \leq n$. Define $g_u := U^T g_i$ and $s_u(\lambda) = WUs(\lambda)$. From this, we may deduce that $s(\lambda) = Us_u(\lambda)$ and $g = WUg_u$ using (3.21). It then follows from (3.19) that

$$(H_i + \lambda W)s(\lambda) = (H_i + \lambda W)Us_u(\lambda) \underset{(3.21)}{=} WU(D + \lambda I)s_u(\lambda) = -g_i = -WUg_u.$$

Since W and U are nonsingular, therefore $s_u(\lambda) := (D + \lambda I)^{-1} g_u$. For $\lambda \neq -\lambda_i$ for any $1 \leq i \leq n$,

$$\Psi_c(\lambda) := \|s(\lambda)\|_W^2 = \|s_u\|_2^2 = \|(D + \lambda I)^{-1} g_u\|^2 = \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \tag{3.22}$$

where γ_i is the ith component of $U^T g_i$.

We also rearrange (3.20) to a nonlinear function of λ, such that

$$\mathcal{L}_{\pm}^c(\lambda) := \frac{1}{4\sigma_c}(-\beta \pm \sqrt{\beta^2 + 16\lambda\sigma_c}) = \|s(\lambda)\|_W. \tag{3.23}$$

We say (3.23) is well defined if $-\beta \pm \sqrt{\beta^2 + 16\lambda\sigma_c} \geq 0$ (or equivalently, $\mathcal{L}_{\pm}^c(\lambda) \geq 0$); we note that in all cases detailed below, $\beta^2 + 16\lambda\sigma_c > 0$, and so $\mathcal{L}_{\pm}^c(\lambda)$ are real-valued. Using (3.22) and (3.23), we have the following nonlinear equations that need to be satisfied for λ.

Let $\psi_c(\lambda) = \sqrt{\Psi(\lambda)}$, where $\Psi(\lambda)$ is defined in (3.22); then, solving the secular equations (3.18)–(3.20) is equivalent to finding roots that satisfy

$$\psi_c(\lambda) = \mathcal{L}_{\pm}^c(\lambda) \quad \text{for} \quad \lambda \geq -\lambda_1, \tag{3.24}$$

where λ_1 is the leftmost eigenvalue of the pencil $(H_i; W)$.

The sufficiency implication states that if we can prove the existence of a solution λ^* to (3.24), then it follows that the solution to the system (3.18)–(3.20) exists.

- If $H_i + \lambda^* W \succ 0$, (3.18) yields a unique s_c, indicating a unique global minimizer for $M_c(s)$.
- The hard case, similar to the trust-region subproblem [44, Sec 8.3.1], may occur when $\tau_1 := u_1^T g_i = 0$, leading to non-unique global minimizers for $M_c(s)$.

Case-by-case proof: global minimizer of $M_c(s)$ for $\tau_1 = u_1^T g_i \neq 0$.

We start with the standard cases when g_i is not orthogonal to the first generalised eigenvector of the pencil $(H_i; W)$ (i.e., $\tau_1 \neq 0$). We prove that, in these cases, there exist λ^* satisfying (3.24), and $H_i + \lambda^* W$ is positive definite. Thus, $M_c(s)$ has a global minimizer, which is unique in Cases 1, 2 and 4. To aid the proof, we define point $\mathcal{A}$ as the intersection point for $\mathcal{L}_+^c$ and $\mathcal{L}_-^c$ which have coordinates $(\lambda_A, s_A) := \big(-\frac{\beta^2}{16\sigma_c}, -\frac{\beta}{4\sigma_c}\big)$.

Case 1: If H_i is indefinite and $\tau_1 \neq 0$, we seek a solution satisfying $\lambda > -\lambda_1 \geq 0$. As illustrated by the left two plots in Figure 3.1, for all $\beta \in \mathbb{R}$ and $\lambda > -\lambda_1 \geq 0$, only $\mathcal{L}_+^c(\lambda)$ is well defined (i.e. $\mathcal{L}_+^c(\lambda) \geq 0$). Since $\psi_c(\lambda) = \sqrt{\Psi(\lambda)}$ is monotonically decreasing with $\lim_{\lambda \to -\lambda_1} \psi_c(\lambda) \to \infty$ and $\lim_{\lambda \to \infty} \psi_c(\lambda) \to 0$ and $\mathcal{L}_+^c(\lambda)$ is monotonically increasing with $\mathcal{L}_+^c(-\lambda_1) \geq 0$, there is a unique solution λ^* in the $\mathcal{L}_+^c(\lambda)$ branch of (3.24). Note that $\lambda^* > -\lambda_1$ and $H_i + \lambda^* W \succ 0$, ensuring the existence of a unique global minimizer.

Case 2: If $\beta \geq 0$, $H \succ 0$ and $\tau_1 \neq 0$, the intersection point $\mathcal{A} := (\lambda_A, s_A) = \big(-\frac{\beta^2}{16\sigma_c}, -\frac{\beta}{4\sigma_c}\big)$ must be in the third quadrant and the only branch that is well defined is $\mathcal{L}_+^c(\lambda)$ for $\lambda > 0$ (an example illustrating this is given in the right plot of Figure 3.1). Moreover, $\mathcal{L}_+^c(\lambda)$ is monotonically increasing for $\lambda > 0$ with $\mathcal{L}_+^c(0) = 0$ and $\lim_{\lambda \to \infty} \mathcal{L}_+^c(\lambda) \to \infty$. $\psi_c(\lambda)$ is monotonically decreasing with $\psi_c(0) > 0$

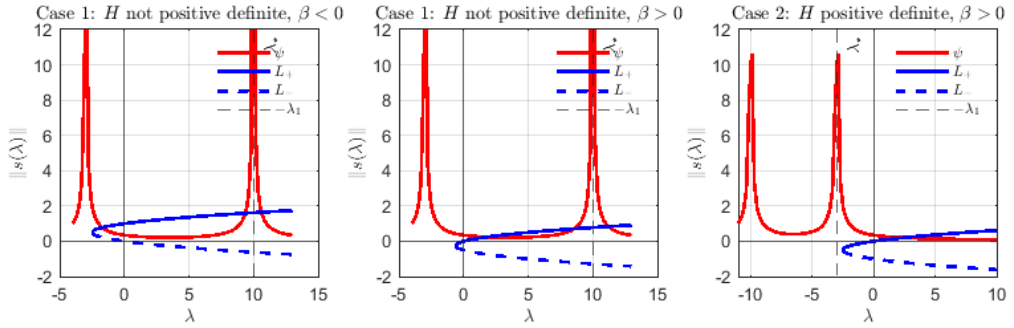


Figure 3.1: $\sigma_c = 10$, $W = I_2$, $g_i = [1, -1]^T$. **Left plot (Case 1):** $\beta = -20$, $H_i = [-10, 0; 0, 3]$. **Mid plot (Case 1):** $\beta = 20$, $H_i = [-10, 0; 0, 3]$; **Right plot (Case 2):** $\beta = 20$, $H_i = [10, 0; 0, 3]$;

and $\lim_{\lambda\to\infty} \psi_c(\lambda) \to 0$. Thus, there is a unique solution λ^* lying in the $\mathcal{L}_+^c(\lambda)$ branch for (3.24) with $\lambda^* > -\lambda_1$. The uniqueness of the global minimizer follows similarly to Case 1.

Case 3: If $\beta \leq 0$, $H \succ 0$, and $\tau_1 \neq 0$, as shown in the left plot of Figure 3.2, (3.24) may have multiple solutions, so we need to impose extra conditions on β. At least one solution lies in the $\mathcal{L}_-^c$ trench[2] if point $\mathcal{A}$ stays at the right of $\lambda = -\lambda_1$ and point $\mathcal{A}$ is above ψ_c (the left plot of Figure 3.2). In other words,

$$\lambda_A = -\frac{\beta^2}{16\sigma_c} > -\lambda_1, \qquad \text{and} \qquad s_A = -\frac{\beta}{4\sigma_c} > \psi_c\Big(-\frac{\beta^2}{16\sigma_c}\Big).$$

Note that we have $\lambda^* \geq \lambda_A > -\lambda_1$, therefore $H_i + \lambda^* W \succ 0$, so this root yields a valid minimizer; as in Case 5, the global minimizer is the root that minimizes M_c.

Lemma 3.2.1. ***(Case 3)*** *Assume $H \succ 0$, and $\tau_1 \neq 0$, if $\beta \leq 0$ and satisfies $\frac{\beta^2}{16\sigma_c} < \lambda_1$ and $\psi_c(-\frac{\beta^2}{16\sigma_c}) = \big\|\big[H_i - \frac{\beta^2}{16\sigma_c} W\big]^{-1} g\big\|_W < -\frac{\beta}{4\sigma_c}$, then there exists at least one solution λ^* to (3.24) that lies in the $\mathcal{L}_-^c(\lambda)$ branch.*

Proof. Since $\beta \leq 0$, we have $\lambda_A = -\frac{\beta^2}{16\sigma_c} < 0$ and $s_A = \frac{\beta}{4\sigma_c} > 0$. Point $\mathcal{A}$ which have coordinates $(\lambda_A, s_A) := (-\frac{\beta^2}{16\sigma_c}, -\frac{\beta}{4\sigma_c})$ is in the second quadrant. Since $\mathcal{L}_-^c$ is only well-defined for $\lambda > \lambda_A$, the range of interest for finding a solution is $\lambda \in (\lambda_A, 0)$. Both ψ_c and $\mathcal{L}_-^c$ are continuous and monotonic in the interval $(\lambda_A, 0)$. Since $\psi_c(\lambda_A) \leq -\frac{\beta}{4\sigma_c} = \mathcal{L}_-^c(\lambda_A)$ and $\psi_c(0) > 0 = \mathcal{L}_-^c(0)$, there must be at least one intersection between $\mathcal{L}_-^c$ and ψ_c within this interval. Lastly, we can clarify that the feasible root cannot occur in the $\mathcal{L}_+^c$ branch. This is because $\mathcal{L}_+^c$ is monotonically increasing and ψ_c is monotonically decreasing for all $\lambda > \lambda_A$ and $\mathcal{L}_+^c(\lambda_A) = \mathcal{L}_-^c(\lambda_A) > \psi_c(\lambda)$. □

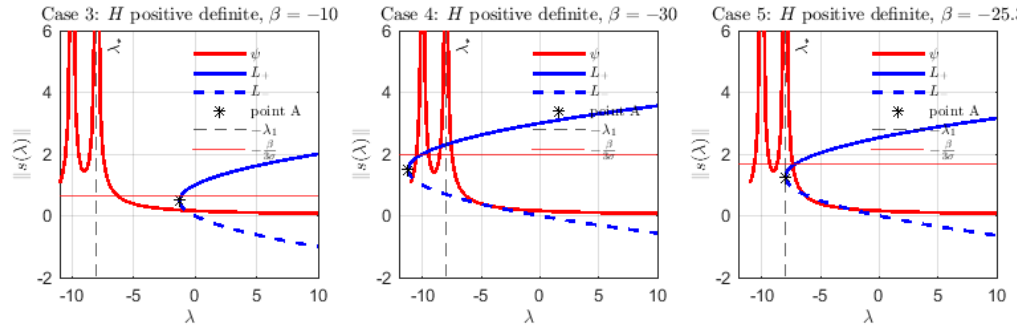


Figure 3.2: $\sigma_c = 5$, $W = I_2$, $g_i = [1, -1]^T$, $H_i = [10, 0; 0, 8]$. **Left plot (Case 3):** $\beta = -10$. **Mid plot (Case 4):** $\beta = -30$; **Right plot (Case 5):** $\beta = -25.3$;

Case-by-case proof: non-unique global minimizer of $M_c(s)$ for $\tau_1 = u_1^T g_i = 0$.

[2]We are grateful to Mr Charlie Offen, who observed during his MSc dissertation project that the secular equation may admit more than one root in Case 3.

Now we consider the case where g_i is orthogonal to the generalised eigenvector of the pencil $(H_i; W)$ (i.e., $\tau_1 = u_1^T g_i = 0$). If $\tau_1 = 0$, we may have difficulties similar to the hard case of the trust-region subproblem [44, Sec 8.3.1].

Under these circumstances, (3.18) will be consistent if we set $\lambda = \lambda_s := -\lambda_1$, such that (3.18) becomes $(H + \lambda_s W)s_s = -g_i$. Thus, $H + \lambda_s W \succeq 0$, so (3.19) is satisfied. However, since $\lambda_1 = -\lambda_s$ is the leftmost eigenvalue of the pencil $(H_i; W)$, it follows that $(H_i + \lambda_s W)u_1 = 0$ for all u_1 corresponding to λ_1 and thus

$$g_i + (H_i + \lambda_s W)(s_s + \upsilon u_1) = 0 \tag{3.25}$$

for any scale $\upsilon \in \mathbb{R}$. Thus, in order to satisfy (3.20), we must additionally choose the scalar $\upsilon = \upsilon_s$ such that $\lambda_s = \frac{\beta}{2}\|s_s + \upsilon_s u_1\|_W + \sigma_c\|s_s + \upsilon_s u_1\|_W^2$. To obtain an exact solution in the hard case requires the generalised eigenvalue λ_1, a corresponding eigenvector u_1 of H_i, and the 'trajectory' vector $s_s = \lim_{\lambda \to \lambda_s} s(\lambda) = -(H_i + \lambda_s W)^+ g$ where $(H_i + \lambda_s W)^+$ is the generalised inverse of $H_i + \lambda_s W$. We summarize these results in the Theorem 3.2.4. More details and full proof can be found in [44, Corollary 8.3.1].

Theorem 3.2.4. *Any global minimizer of M_c may be expressed as*

$$s_c = s_s \text{ uniquely, if } \tau_1 \neq 0, \qquad \text{or} \qquad s_c = s_s + \upsilon_s u_1, \text{ if } \tau_1 = 0,$$

where $\tau_1 = u_1^T g_i$, $s_s = -(H_i + \lambda_s W)^+ g_i$ and $\lambda_s = \frac{\beta}{2}\|s_c\|_W + \sigma_c\|s_c\|_W^2 > -\lambda_1$, where υ_s is one of the two roots of $\lambda_s = \frac{\beta}{2}\|s_s + \upsilon_s u_1\|_W + \sigma_c\|s_s + \upsilon_s u_1\|_W^2$.

3.2.2.3 General Sufficient Optimality Conditions for the Global Minimum of M_c

In this subsection, we combine Theorem 3.2.2 and Theorem 3.2.3 to provide a general sufficient optimality result; see Figure 3.3 for a graphical illustration of the ensuing suitable values of β. The sufficient conditions in Theorem 3.2.3 correspond to Cases 1 to 3 in Theorem 3.2.5. The sufficiency conditions in Theorem 3.2.2 not only confirm the results of Theorem 3.2.3 but also extend the range of cases that we can address (see Case 4).

Theorem 3.2.5. *(General sufficient optimality conditions for the global minimum of M_c) Consider the CQR polynomial $M_c(s)$ in (3.2), let $s_c \in \mathbb{R}^n$ be a vector that satisfies (3.18), (3.19), and (3.24) with the corresponding branch selected as described in Case 1 to Case 5 below. Then, s_c is a global minimizer of $M_c(s)$ over $\mathbb{R}^n$.*

- ***Case 1:*** *(From Theorem 3.2.2 and Theorem 3.2.3) If H_i is indefinite, choose $\mathcal{L}_+^c$ branch in (3.24).*
- ***Case 2:*** *(From Theorem 3.2.2 and Theorem 3.2.3) If H_i is positive definite and $\beta \geq 0$, choose $\mathcal{L}_+^c$ branch in (3.24).*
- ***Case 3:*** *(From Theorem 3.2.3) If H_i is positive definite,*

$$-4\sqrt{\sigma_c \lambda_1} \leq \beta \leq 0, \quad \text{and} \quad -\frac{\beta}{4\sigma_c} > \psi_c\Big(-\frac{\beta^2}{16\sigma_c}\Big) = \left\|\left(H_i - \frac{\beta^2}{16\sigma_c}W\right)^{-1} g\right\|_W, \tag{3.26}$$

  choose $\mathcal{L}_-^c$ branch in (3.24).

- ***Case 4:*** *(From Theorem 3.2.2) H_i is positive definite, $\beta \leq -3\sqrt{2}\sqrt{\lambda_1 \sigma_c} \approx -4.24\sqrt{\lambda_1 \sigma_c}$, choose $\mathcal{L}_+^c$ branch in (3.24).*
- ***Case 5:*** *In the rest of the cases, namely, if H_i is positive definite and $-3\sqrt{2}\sqrt{\lambda_1 \sigma_c} \leq \beta \leq -4\sqrt{\lambda_1 \sigma_c}$; or H_i is positive definite,*

$$-4\sqrt{\lambda_1 \sigma_c} \leq \beta \leq 0 \quad \text{and} \quad -\frac{\beta}{4\sigma_c} < \psi_c\big(-\frac{\beta^2}{16\sigma_c}\big) = \left\|\big(H_i - \frac{\beta^2}{16\sigma_c}W\big)^{-1} g\right\|_W,$$

multiple roots satisfying (3.24) may exist. The root that minimizes the value of M_c is a global minimizer of M_c. A root, denoted by (λ_+, s_+), exists within the $\mathcal{L}_+^c$ branch and if $\beta \in \left(-\frac{3\sqrt{6}}{2}\sqrt{\lambda_1\sigma_c}, 0\right]$, this solution ensures $M_c(s_+) < M_c(0)$.

All possible values of $\beta \in \mathbb{R}$ are addressed in Theorem 3.2.5. This indicates that by finding the root of (3.24) that satisfies (3.18) and (3.20), we can find a global minimizer of M_c for all $\beta \in \mathbb{R}$ and $\sigma_c > 0$. The intersection point of the $\mathcal{L}_+^c$ and $\mathcal{L}_-^c$ branches, denoted as $\mathcal{A} = (\lambda_A, s_A) = \left(-\frac{\beta^2}{16\sigma_c}, -\frac{\beta}{4\sigma_c}\right)$, corresponds to various cases outlined in Theorem 3.2.5. When H is indefinite, regardless of the location of point $\mathcal{A}$, we are in Case 1 (as shown in the left plot of Figure 3.3). If H is indefinite, when point $\mathcal{A}$ falls within specific colored regions, it corresponds to the cases described in Theorem 3.2.5 (as shown in the right plot of Figure 3.3). As illustrated in Figure 3.3, Cases 1–4 encompass the majority of β values, while Case 5 covers the remaining situations. An example of Case 5 is presented in the right plot of Figure 3.2.

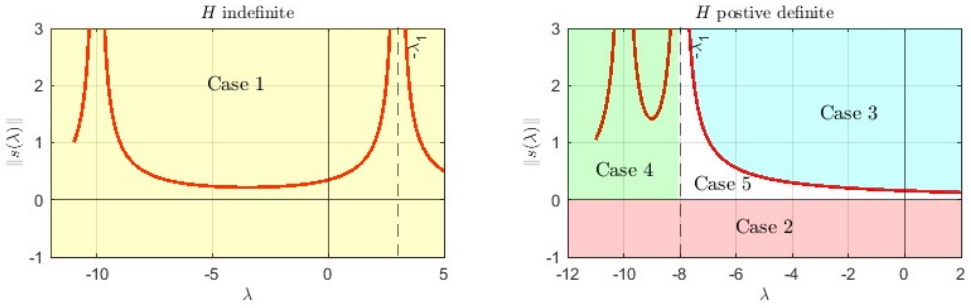


Figure 3.3: Illustration plot depicting the location of point $\mathcal{A}$ corresponding to different cases outlined in Theorem 3.2.5. $\sigma_c = 5$, $W = I_2$, $g_i = [1, -1]^T$, $H_i = [10, 0; 0, -3]$ for the left plot and $H_i = [10, 0; 0, 8]$ for the right plot. The red curve represents $\psi_c(\lambda)$.

Cases 1, 2, and 3 follow from Theorem 3.2.3. Cases 1, 2, and 4 are derived from Theorem 3.2.2, and therefore, $\|s_c\|_W$ satisfies $\|s_c\|_W \geq -\frac{\beta}{3\sigma_c}$.

Proof. of Theorem 3.2.5: **Case 1 and Case 2: follows from Theorem 3.2.2 and Theorem 3.2.3.** According to Theorem 3.2.2, if $\beta \geq 0$, we can observe from (3.17) that $\mathcal{F}_2 > 0$, and consequently, $M_c(s + w) > M_c(s)$ for all $w \in \mathbb{R}^n$. The condition on β is naturally satisfied. According to Theorem 3.2.3, as discussed in the case-by-case analysis, there exists a unique root $\lambda^* > -\lambda_1$ in the $\mathcal{L}_+^c(\lambda)$ for the nonlinear equation (3.24).

Case 3: follows from Theorem 3.2.3. This case is established using Theorem 3.2.3 and Lemma 3.2.1. It is worth noting that this case cannot be derived using Theorem 3.2.2. The reason is that any solution found in the $\mathcal{L}_-^c$ branch satisfies $\|s_c\|_W \leq s_A = -\frac{\beta}{4\sigma_c}$, which does not always meet the conditions of (3.16).

Case 4: follows from Theorem 3.2.2. Assuming $\beta \leq 0$, $H \succ 0$, and $\tau_1 \neq 0$, the sufficiency conditions provided in Theorem 3.2.2 offer an alternative approach to characterise the global minimizer. If point $\mathcal{A}$ is positioned to the left of $\lambda = -\lambda_1$ and $\mathcal{L}_+^c(-\lambda_1) \geq -\frac{\beta}{3\sigma_c}$ (as depicted in the mid plot of Figure 3.2), then there exists a solution that satisfies the sufficient condition in Theorem 3.2.2. This solution lies within the $\mathcal{L}_+^c$ branch and represents the global minimizer of M_c. In other words, to meet the sufficient condition in Theorem 3.2.2, the following conditions are required:

$$\lambda_A = -\frac{\beta^2}{16\sigma_c} \leq -\lambda_1, \quad \text{and} \quad \mathcal{L}_+^c(-\lambda_1) = \frac{1}{4\sigma_c}(-\beta + \sqrt{\beta^2 - 16\lambda_1\sigma_c}) \geq -\frac{\beta}{3\sigma_c}.$$

This implies $\beta \leq -3\sqrt{2}\sqrt{\lambda_1\sigma_c}$.

Case 5: In the remaining cases, there may be multiple solutions satisfying (3.24) as depicted in the right plot of Figure 3.2. By the necessary optimality conditions, the solution that minimizes the function value of M_c corresponds to the global minimizer. As point $\mathcal{A}$ either lies below the curve $\psi_c(\lambda)$

or to the left of $\lambda = -\lambda_1$, a root in the $\mathcal{L}_+^c$ branch is always guaranteed. Let (λ_+, s_+) be the root in the $\mathcal{L}_+^c$ branch satisfying (3.24) and consequently satisfying (3.18)–(3.20). We deduce that

$$\begin{aligned} M_c(0) - M_c(s_+) &= -g_i^T s_+ - \frac{1}{2}H_i[s_+]^2 - \frac{\beta}{6}\|s_+\|^3 - \frac{\sigma_c}{4}\|s_+\|^4 \underset{(3.18)}{=} \frac{1}{2}H_i[s_+]^2 + \frac{\beta}{3}\|s_+\|^3 + \frac{3\sigma_c}{4}\|s_+\|^4 \\ &\geq \|s_+\|^2\left(\frac{1}{2}\lambda_1 + \frac{\beta}{3}\|s_+\| + \frac{3\sigma_c}{4}\|s_+\|^2\right) > 0. \end{aligned}$$

Since $\lambda_1 > 0$ and if $\beta \in \left(-\frac{3\sqrt{6}}{2}\sqrt{\lambda_1\sigma_c}, 0\right]$, the discriminant of the quadratic equation $\frac{1}{2}\lambda_1 + \frac{\beta}{3}\|s_+\| + \frac{3\sigma_c}{4}\|s_+\|^2$ is negative, i.e., $\frac{\beta^2}{9} - \frac{3\lambda_1\sigma_c}{2} < 0$. This gives $M_c(0) - M_c(s_+) > 0$. □

3.2.3 An Algorithm for Minimizing the CQR Polynomial

As suggested by Theorem 3.2.1 and Theorem 3.2.5, we have transformed the task of identifying a global minimizer of M_c into the problem of solving the nonlinear equations (3.24) for λ in the corresponding branch. To numerically determine the root of (3.24), we use the global optimality characterisations outlined in Sections 3.2.1 to 3.2.2. Additionally, we extend well-established Cholesky factorization-based root-finding techniques, as detailed in [44, Ch 8.2.3]. The primary objective of this subsection is to formulate an algorithm for locating the root of the univariate (nonlinear) secular equation. We employ univariate Newton iteration to find the necessary root of (3.24). Further convergence analysis for the root-finding method is deferred to future work.

In practice, as suggested by [46, Sec 6.1], instead of solving (3.24), it may be preferable to solve

$$\phi_1(\lambda) := \left[\psi_c(\lambda)\right]^{-1} - \left[\mathcal{L}_\pm^c(\lambda)\right]^{-1} = 0 \qquad \text{for} \qquad \lambda \geq \lambda_1 \tag{3.27}$$

where the choice between using $\mathcal{L}_+^c$ branch or $\mathcal{L}_-^c$ branch is made on a case-by-case basis, as determined by Theorem 3.2.5. The updates for the Newton iteration are detailed in Lemma 3.2.2. The proof for Lemma 3.2.2 employs techniques similar to those found in [46, Lemma 6.1], [44, Corollary 9.1.1], and [62, Lemma 7.3.1].

Lemma 3.2.2. *Suppose $g_i \neq 0$, the Newton iteration updates are*

$$\Delta\lambda^{(k)} = -\frac{\phi_1(\lambda^{(k)})}{\phi_1'(\lambda^{(k)})} = -\frac{\|s(\lambda^{(k)})\|_W^{-1} - \left[\mathcal{L}_\pm^c(\lambda^{(k)})\right]^{-1}}{\|\bar{\omega}(\lambda^{(k)})\|^2\|s(\lambda^{(k)})\|_W^{-3} - \left[(\mathcal{L}_\pm^c(\lambda^{(k)}))^{-1}\right]'}, \lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)} \tag{3.28}$$

where $\bar{\omega}(\lambda)$ is given by $(H_i + \lambda W) = L(\lambda)L(\lambda)^T$ and $W^{-1}L(\lambda)\bar{\omega}(\lambda) = s(\lambda)$. Also,

$$\mathcal{L}_\pm^c(\lambda)^{-1} = \frac{2\sigma_c}{-\beta \pm \sqrt{\beta^2 + 4\lambda\sigma_c}}, \quad \left[(\mathcal{L}_\pm^c(\lambda))^{-1}\right]' = \frac{\mp 4\sigma_c^2}{(-\beta \pm \sqrt{\beta^2 + 4\lambda\sigma_c})^2(\beta^2 + 4\lambda\sigma_c)^{1/2}}.$$

Proof. Let $H_W(\lambda) := H_i + \lambda W$, which has the Cholesky factorization $H_W(\lambda) = L(\lambda)L(\lambda)^T$. Differentiating $(H_i + \lambda W)s(\lambda) = -g_i$ with respect to λ gives $(H_i + \lambda W)\nabla_\lambda s(\lambda) + Ws(\lambda) = 0$. Therefore,

$$\nabla_\lambda s(\lambda) = -H_W^{-1}(\lambda)Ws(\lambda) = L(\lambda)^{-T}L(\lambda)^{-1}Ws(\lambda). \tag{3.29}$$

On the other hand, $\Psi(\lambda) = \|s(\lambda)\|_W^2$ has derivative $\Psi'(\lambda) = 2\langle Ws(\lambda), \nabla_\lambda s(\lambda)\rangle$ where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product. Substituting (3.29) gives,

$$\begin{aligned} \Psi'(\lambda) &= -2\langle Ws(\lambda), L(\lambda)^{-T}L(\lambda)^{-1}Ws(\lambda)\rangle \\ &= -2\langle L(\lambda)^{-1}Ws(\lambda), L(\lambda)^{-1}Ws(\lambda)\rangle = -2\|L(\lambda)^{-1}Ws(\lambda)\| = -2\|\bar{\omega}(\lambda)\|^2 \end{aligned}$$

where $W^{-1}L(\lambda)\bar{\omega}(\lambda) = s(\lambda)$. Consequently,

$$(\psi_c(\lambda)^{-1})' = (\Psi(\lambda)^{-1/2})' = -\frac{1}{2}\Psi(\lambda)^{-3/2}\Psi_c'(\lambda) = \|s(\lambda)\|_W^{-3}\|\bar{\omega}(\lambda)\|^2. \tag{3.30}$$

Substituting (3.30) into (3.27) and then applying the Newton updates $\Delta\lambda^{(k)} = -\frac{\phi_1(\lambda^{(k)})}{\phi_1'(\lambda^{(k)})}$ yields the result. □

Numerical Set-up and Initialization for the Newton Step: In a similar manner to the techniques used to solve the trust region and ARC subproblems [44, Ch 8.2.3] [46, Sec 6.1], for the minimization of the CQR polynomial in (3.1), we employ Newton's method to solve the secular equation (3.27). As described in Theorem 3.2.5, the solution of the secular equation (3.27) falls into either Case 1, 2, 4, or 5, where it resides in the $\mathcal{L}_+^c$ branch, or Case 3, where it lies within the $\mathcal{L}_-^c$ branch.

In practical implementation, we first search for a solution, s_+, in the $\mathcal{L}_+^c$ branch using the initialization in Remark 3.2.1. If no solution is found in the $\mathcal{L}_+^c$ branch or if $M_c(s_+) \geq M_c(0)$ (which is only possible within Case 5), we start searching for a solution in the $\mathcal{L}_-^c$ branch. Based on our numerical experiments in Section 3.4, we find that in approximately 90% of the iterations, the solution lies in the $\mathcal{L}_+^c$ branch. The algorithm is detailed in Algorithm 3.2.1, and the initialization details are provided in Remark 3.2.1

Algorithm 3.2.1: Newton's method to Solve (3.27)

Initialization: Initialize the counter $k = 0$, $s^{(0)} = 0$ and $\lambda^{(0)}$ as in Remark 3.2.1. Set an accuracy level $\epsilon_N > 0$, usually $\epsilon_N = 10^{-8}$ smaller than the accuracy of the subproblem.
Initialize branch: Choose $\mathcal{L}_+^c$ branch: $\mathcal{L}_T^c(\cdot) := \mathcal{L}_+^c(\cdot)$.
Input: f_i, g_i, H_i, β and $\sigma_c > 0$.
while $\left| \|s^{(k)}\| - \mathcal{L}_T^c(\lambda^{(k)}) \right| > \epsilon$ **do**
  if $H_i + \lambda^{(k)} W \succ 0$ **then**
    Cholesky factorize $(H_i + \lambda^{(k)} W) = LL^T$. Solve $LL^T s^{(k)} = -g_i$ and $W^{-1} L \bar{\omega}^{(k)} = s^{(k)}$.
    Compute the Newton update using (3.28), with $\mathcal{L}_\pm^c$ set to the corresponding branch $\mathcal{L}_T^c$.
  else
    Switch to $\mathcal{L}_-^c$ branch: $\mathcal{L}_T^c(\cdot) = \mathcal{L}_-^c$. Set $\lambda^{(0)} = 0$.

Remark 3.2.1. *(Initialization for the Newton Algorithm) In Cases 1, 2, 4, and 5, where the solution resides (possibly) in the $\mathcal{L}_+^c$ branch, we initialize $\lambda^{(0)}$ as follows:*

- *Case 1 and Case 2: If H_i is indefinite, we set $\lambda^{(0)} := \max\{-\lambda_1, 0\} + 10^{-6}$. This ensures $H + \lambda^{(0)} I \succ 0$.*
- *Case 4 and 5: If $H_i \succ 0$ and $\beta < 0$, we set $\lambda^{(0)} := \max\{-\lambda_1, \lambda_A\} + 10^{-6}$ where $\lambda_A = -\frac{\beta^2}{16\sigma_c}$ denotes the intersection point for $\mathcal{L}_+^c$ and $\mathcal{L}_-^c$. This initialization also ensures $H + \lambda^{(0)} I \succ 0$. This choice is motivated by the observation that, as illustrated in the two leftmost plots of Figure 3.1, the solution to the secular equation (3.27) consistently resides in the right side of $\lambda = -\lambda_{\min}[H]$ and $\lambda = \lambda_A$.*

If no solution is found in the $\mathcal{L}_+^c$ branch, it implies we are in Case 3 or in Case 5. Consequently, we proceed to seek a solution within the $\mathcal{L}_-^c$ branch. In Case 3, where $H_i \succ 0$ and $\beta < 0$, as illustrated on the left plot in Figure 3.2, the solution to the secular equation (3.27) is near $\lambda = 0$ and falls within the interval $[\lambda_A, 0]$. Therefore, we initialize $\lambda^{(0)} = 0$.

Regardless of the case, we ensure that $H_i + \lambda^{(0)} I_n$ is positive definite, making the Cholesky factorization possible in the first iteration. While it is worth noting that this does not entirely avoid the hard case, as it is only the first iteration, this is a measure we take to ensure that we do not start the algorithm with an indefinite $H_i + \lambda^{(0)} I_n$. Throughout the iterations of Algorithm 3.2.1, in all the numerical examples we examined, the positive definiteness of $H_i + \lambda^{(k)} I_n$ is preserved, and we did not encounter the hard case. The Cholesky factorization also remains valid in the cases we tested. However, note that Algorithm 3.2.1 is a preliminary implementation of Newton root-finding for the nonlinear equation (3.24). Addressing and implementing the hard case, as well as the convergence analysis of Algorithm 3.2.1, is dedicated to future work.

Theorem 3.2.1 and Theorem 3.2.5 give a global optimality characterization for the CQR polynomial in (3.1). However, in Case 5 of Theorem 3.2.5, multiple solutions may exist. Then, since we are using

Newton's method, depending upon the initialization, Algorithm 3.2.1 may converge to a root in $\mathcal{L}_+^c$ which corresponds to a local minimizer of M_c. To locate the global minimizer in Case 5, one needs to locate all the roots in the $\mathcal{L}_+^c$ and $\mathcal{L}_-^c$ branch (there may be multiple roots) and evaluate the value of M_c at these roots. The roots that give the lowest function value of M_c will correspond to global minimizers of M_c. We defer the implementation of this additional root-finding procedure to future algorithmic developments. Numerically, in all the cases that we tested, the algorithm converges to a global minimizer in Cases 1–4 and converges to a local minimizer $s^{(K)}$, with a decrease in M_c value in Case 5, such that $M_c(s^{(K)}) < M_c(s^{(0)})$.

Our current approach here, for solving the system (3.18)–(3.20) and (3.24), uses Cholesky factorization and Newton's method to address the secular equation (3.24), and is just one of many possibilities. Similar to existing approaches for trust region and regularization subproblems ([46, Ch. 6] and [55, Ch. 8, 10]), scalable iterative algorithms could be designed here, based on Krylov methods, eigenvalue formulations, or subspace optimization. While we recognize the potential of these tools for solving (3.18)–(3.20) and finding the global minimum of the CQR polynomial, our primary focus in this chapter is on using the CQR polynomial to minimize the m_3 subproblem. As a result, we defer a detailed analysis of these alternative approaches to future work.

3.3 The CQR Algorithmic Framework for Minimizing Local Models

The CQR algorithm minimizes the AR3 subproblem by globally minimizing a sequence of potentially nonconvex CQR polynomials (i.e., quadratic models with cubic term and quartic regularization)[3]. Additionally, the CQR algorithmic framework is tensor-free for suitable choices of β_i and their bounds. This means that in the run of the algorithm, we do not require the entries of T to be stored; we only need vector and matrix products with the tensor T (such as $T[s]$ or $T[s]^2$). We define the norms of the derivatives as $\|g\| := \sqrt{g^T g}$, $\|H\| := \max_i |\lambda_i(H)|$, and $\|T\| = \max_{\|u_1\|=\|u_2\|=\|u_3\|=1} \left|T[u_1][u_2][u_3]\right| < \infty$. In this section and the rest of the chapter, when constructing the CQR model M_c in (3.1), we choose $W = I_n$.

It is worth noting that the CQR method can be applied directly to the minimization of an objective function $f : \mathbb{R}^n \to \mathbb{R}$ under suitable assumptions (not just the AR3 subproblem). Specifically, if f is three times continuously differentiable, bounded below, and the third derivative is globally Lipschitz continuous, then the CQR method can also be used for the minimization of a potentially nonconvex objective function. However, our interest here is not to create a general method for optimizing f; instead, we focus on efficiently minimizing the local model $m_3(s)$. Algorithm 3.3.1 outlines the framework of the CQR algorithm. For the purpose of generality, we use $\hat{m}$ to denote the general objective function to be minimized.

The purpose of β_i is to provide a scalar approximation of the third-order information $\frac{T[s]^3}{\|s\|^3}$. For a given tensor $T \in \mathbb{R}^{n\times n\times n}$, $\frac{T[s]^3}{\|s\|^3}$ is bounded. Also, the term $\frac{T[s]^3}{\|s\|^3}$ can be either positive or negative. Therefore, we allow β_i to take both positive and negative values. More details about the computation of β_i are given in Section 3.4.

The parameter d_i is an adaptive regularization weight; $d_i \geq 0$ ensures that M_c remains bounded from below. In some iterations, if β_i is positive, then $\beta_i\|s\|^3$ can serve as a regularization term for the M_c model, potentially superseding the role of the quartic regularization term $d_i\|s\|^4$. Thus, in these iterations, we have more flexibility in choosing d_i (i.e., we can set $d_i = 0$).

Remark 3.3.1. *(**Minimization of** M_c) In Algorithm 3.3.1, we assume that we find a global minimizer of M_c, such that $s_c^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_c(s^{(i)}, s)$. For optimal complexity analysis, we focus on a variant of the CQR algorithmic framework (Section 3.3.1). While our proof in Section 3.3.1 (in particular Lemma 3.3.1) assumes exact minimization of $\nabla M_c(s^{(i)}, s_c) = 0$, similar bounds can be derived when considering approximate minimization of M_c with a step-termination condition, such as $\|\nabla M_c(s^{(i)}, s_c)\| \leq \Theta\|s_c\|^3$ for some $\Theta \in (0, 1)$. Such step-termination conditions are also used in [53], with alternative variants available [46].*

[3]For a fixed subproblem in (Quartically Regularized Cubic Model), x_k is fixed, and we have access to the coefficients for the subproblem; namely, $f_0 = f(x_k)$, $g = \nabla_x f(x_k)$, $H = \nabla_x^2 f(x_k)$, $T = \nabla_x^3 f(x_k)$, and $\sigma > 0$.

Algorithm 3.3.1: CQR Algorithmic Framework with Adaptive Regularization

Initialization: Set $s^{(0)} = \mathbf{0} \in \mathbb{R}^n$ and $i = 0$. An initial regularization parameter $d_0 = 1$, constants $\eta_1 > \eta > 0$, $\gamma > 1 > \gamma_2 > 0$.

Input: $f_0 := \hat{m}(s^{(0)})$, $g_0 := \nabla \hat{m}(s^{(0)})$, $H_0 := \nabla^2 \hat{m}(s^{(0)})$, $\sigma > 0$ is the given regularization parameter in $\hat{m}$ as defined in (Quartically Regularized Cubic Model); an accuracy level $\epsilon > 0$.

Step 1: Test for termination. If $\|g_i\| = \|\nabla \hat{m}(s^{(i)})\| \leq \epsilon$, terminate with the ϵ-approximate first-order minimizer $s_\epsilon = s^{(i)}$.

Step 2: Step computation. Compute[a] β_i to give a scalar representation of the (local) tensor information in $\hat{m}$. Set

$$M_c(s^{(i)}, s) := f_i + g_i s + \frac{1}{2} H_i[s]^2 + \frac{1}{6}\beta_i \|s\|^3 + \frac{1}{4}\underbrace{(\sigma + 4d_i)}_{:=\sigma_c^{(i)}}\|s\|^4. \qquad \text{(CQR Model)}$$

Compute $s_c^{(i)} = \operatorname{argmin}_{s \in \mathbb{R}^n} M_c(s^{(i)}, s)$ (see Remark 3.3.1).

Step 3: Acceptance of trial point. Compute $\hat{m}(s^{(i)} + s_c^{(i)})$ and define

$$\rho_i = \frac{\hat{m}(s^{(i)}) - \hat{m}(s^{(i)} + s_c^{(i)})}{\hat{m}(s^{(i)}) - M_c(s^{(i)}, s_c^{(i)})}. \qquad \text{(CQR Ratio Test)}$$

Step 4: Regularization Parameter Update.

- If $\rho_i \geq \eta_1$ *(very successful step)*: Set $s^{(i+1)} = s^{(i)} + s_c^{(i)}$, $d_{i+1} = \gamma_2 d_i$, update β_{i+1} (more details in Section 3.4), and compute g_{i+1}.
- If $\eta \leq \rho_i \leq \eta_1$ *(successful step)*: Set $s^{(i+1)} = s^{(i)} + s_c^{(i)}$, $d_{i+1} = d_i$, update β_{i+1}, and compute g_{i+1}.
- If $\rho_i \leq \eta$, *(unsuccessful step)* $s^{(i+1)} = s^{(i)}$, increase $d_{i+1} = \gamma d_i$, $\beta_{i+1} = \beta_i$.

Increment i by one and go to Step 1.

[a] Details about selecting β_i can be found in Algorithm 3.4.1 in Section 3.4.

In the CQR model described by (3.1), we employ the standard Euclidean norm for the fourth-order regularization term. However, in the case of objective function separability, an alternative CQR model using a different fourth-order regularization term could provide a more accurate approximation of $M(s^{(i)}, s)$ and result in improved complexity bounds. Further details on this variant are discussed in Section 3.3. Additionally, some literature [20; 53] employs variants of the (CQR Ratio Test). For instance, $M_c(s^{(i)}, s_c^{(i)})$ can be replaced by the third-order Taylor expansion of $m_3(s)$ at $s = s^{(i)}$, denoted as $T_3(s^{(i)}, s_c^{(i)})$. In both formulations, it holds that for any base point, a successful iteration can be achieved after a finite number of regularization parameter updates.

This section is organized as follows: in Section 3.3.1, we introduce a variant of the CQR algorithmic framework for minimizing m_3 and establish a bound on its global rate of convergence. We demonstrate that this variant requires at most $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ function and derivative evaluations to compute an approximate first-order critical point of m_3 that satisfies $\|g_i\| \leq \epsilon$. This bound is at least as good as that of the ARC algorithm for minimizing m_3. In Section 3.3.2, we explore specific instances of m_3 where CQR algorithm exhibits improved convergence behaviour.

3.3.1 Convergence and Complexity of CQR Algorithmic Variant for Minimizing m_3

Algorithm 3.3.2 is a variant of the CQR algorithmic framework (Algorithm 3.3.1) for minimizing m_3. We prove an optimal worst-case complexity bound for Algorithm 3.3.2.

Algorithm 3.3.2: CQR Variant of Algorithm 3.3.1 for minimizing m_3

Let $\hat{m} = m_3$ in Algorithm 3.3.1. Fix parameters $B > 0$ and $\alpha \in (0, \frac{1}{2})$. Initialize and proceed through Steps 1 to 3 following the same procedure as outlined in the CQR algorithmic framework (Algorithm 3.3.1). In Step 4, we utilize the following rule for updating the regularization parameter, β_{i+1} and $s^{(i+1)}$. The updating rules for β_i is given in Remark 3.4.1 and Algorithm 3.4.1.

if $\|\nabla m_3(s^{(i)} + s_c^{(i)})\| \leq \epsilon$ **then**

  Terminate with an ϵ-approximate first-order minimizer s_ϵ of $m_3(s^{(i)} + s_c^{(i)})$.

else

  if $\rho_i \geq \eta$ *and* $\beta_i \in [\alpha, B]$ **then**

    Successful Step: $s^{(i+1)} := s^{(i)} + s_c^{(i)}$, update $0 \leq d_{i+1} \leq d_i$, update β_{i+1} (details in Section 3.4), compute g_{i+1}.

  else if $\rho_i \geq \eta$ *and* $\beta_i \in [-B, -4\alpha]$ *and* $\sigma_c^{(i)} \notin \left[\frac{1}{6}, \frac{2}{3}\right]\big(-\beta_i + \alpha\big)\|s_c^{(i)}\|^{-1}$ **then**

    Successful Step: $s^{(i+1)} := s^{(i)} + s_c^{(i)}$, update $0 \leq d_{i+1} \leq d_i$, update β_{i+1}, compute g_{i+1}.

  else if $\rho_i \geq \eta$ *and* $\beta_i \in [-4\alpha, \alpha]$ *and*

$$\sigma_c^{(i)} \geq \frac{2}{3}\big(-\beta_i + \alpha\big)\|s_c^{(i)}\|^{-1} \tag{3.31}$$

  then

    Successful Step: $s^{(i+1)} := s^{(i)} + s_c^{(i)}$, update $0 \leq d_{i+1} \leq d_i$, update β_{i+1}, compute g_{i+1}.

  else

    Unsuccessful Step: $s^{(i+1)} := s^{(i)}$, $\beta_{i+1} := \beta_i$ and increase $d_{i+1} := \gamma \max\{1, d_i\}$.

In this variant, we introduce a positive parameter B that controls the magnitude of the third-order approximations β_i, preventing them from becoming excessively large; namely, we include the following.

Assumption 3.3.1. *For every iteration, i, β_i is uniformly bounded, such that $|\beta_i| \leq B$.*

This bound for β_i is used for the convergence of the CQR algorithm, ensuring optimal complexity and practical numerical implementation. This bound can be easily applied in practice, for example, by

setting $B = \max_{1\leq \iota,j,k\leq n} \left|T[\iota, j, k]\right|$ in the practical CQR algorithm (Algorithm 3.4.1 in Section 3.4), though one has to be careful with this (and other) choice(s) of B in order to keep the tensor-free nature of the main CQR iterations.

In this variant, we also introduce a new constant, $\alpha \in [0, \frac{1}{2}]$, to further control the step length s_c. Due to this additional parameter and the step size control in (3.31), we need to consider two additional cases in which we reject the step and increase regularization. The first case occurs when $-B \leq \beta_i \leq -4\alpha$, and σ_c falls within the narrow region of $\left[\frac{1}{6}, \frac{2}{3}\right]\big(-\beta_i + \alpha\big)|s_c^{(i)}|^{-1}$. The second case arises when β_i is in the narrow range of $[-4\alpha, \alpha]$, and $\sigma_c^{(i)} \geq \frac{2}{3}\big(-\beta_i+\alpha\big)|s_c^{(i)}|^{-1}$. For the remaining cases where β_i and $\sigma_c^{(i)}$ do not fit into these two scenarios, the ratio test defined in (CQR Ratio Test) with an outcome of $\rho_i \geq \eta$ is sufficient to determine the success of the step. In practice, the two additional cases do not happen often. A numerical example is provided in Figure A.1 in Section A.1.3, comparing Algorithm 3.3.2 with and without the additional condition (3.31).The red asterisk represents the iteration that triggers these additional cases.

It is also important to note that whenever we satisfy $\rho_i \geq \eta$ as defined in the CQR algorithmic framework (Algorithm 3.3.1), even without the requirement for $\sigma_c^{(i)}$ in Algorithm 3.3.2, we are guaranteed to have a decrease in values of m_3 at each successful iteration,

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta\left[m_3(s^{(i)}) - M_c(s^{(i)}, s_c^{(i)})\right] > 0$$

where $\eta \in (0, 1]$. In essence, the CQR algorithmic framework (Algorithm 3.3.1) ensures convergence to an approximate local minimum of m_3 that adheres to first-order optimality conditions. But our aim in this section is to establish a global rate of complexity bound, hence the need to introduce an additional requirement involving α.

To establish convergence and determine the complexity bound for Algorithm 3.3.2, we adopt a framework similar to the convergence proof for AR2 as presented in [44, Sec 2.4.1]. The proof is structured as follows:

1. **Groundwork**: We establish upper and lower bounds for the step size (Theorem 3.3.1 and Lemma 3.3.1) using an upper bound on the tensor term (Corollary 3.3.1).

2. **Decrease in the value of** m_3: We prove that the CQR model constructed in successful steps of Algorithm 3.3.2 yields a decrease in the value of m_3 by at least $\mathcal{O}(\epsilon^{\frac{3}{2}})$, where ϵ represents the prescribed first-order optimality tolerances (Theorem 3.3.2 and Theorem 3.3.3).

3. **Bounded Iterations**: We establish that there is a limited number of unsuccessful iterations (Lemma 3.3.2 and Lemma 3.3.3).

Combining these results, we derive the complexity bound for Algorithm 3.3.2 (Lemma 3.3.4 and Theorem 3.3.4). Specifically, Algorithm 3.3.2 is guaranteed to find a first-order critical point of m_3 within at most $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ function and derivatives' evaluations.

Under Assumption 3.3.1, Theorem 3.3.1 and Lemma 3.3.1 establish a uniform upper and lower bound for the norm of the iterates $s^{(i)}$ for both successful and unsuccessful iterations.

Theorem 3.3.1. ***(An upper bound on step size)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is employed. Then, the norm of the iterates (both successful and unsuccessful) are uniformly bounded above independently of i, such that $\|s^{(i)}\| < r_c$ for all $i \geq 0$, where r_c is a constant that depends only on the coefficients of m_3.*

Proof. The proof of Theorem 3.3.1 uses a similar technique to [52, Lemma 3.2]. The proof for Theorem 3.3.1 is given in Section A.1.1. □

Corollary 3.3.1. ***(Upper bounding the tensor term, adapted from Corollary 2.3.2 and Lemma 2.3.3)*** *Let $L_H := \Lambda_0 + 6\sigma r_c$, where Λ_0 defined in Lemma 2.3.3. Then, for every iteration, i,*

$$\|T_i\| := \max_{\|u_1\|=\|u_2\|=\|u_3\|=1} \left|T_i[u_1][u_2][u_3]\right| = \Lambda_0 + 6\sigma\|s^{(i)}\| < \Lambda_0 + 6\sigma r_c = L_H.$$

Proof. See Corollary 2.3.2. □

Note that L_H is an iteration-independent constant that depends only on the coefficients in m_3.

Lemma 3.3.1. ***(A lower bound on the step)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is used. Assume that* $\|\nabla m_3(s^{(i)} + s_c^{(i)})\| > \epsilon$ *and* $\|g_i\| > \epsilon$*, then if* [4] $d_i \geq 0$*,*

$$\|s_c^{(i)}\| > \min\left\{(B + L_H)^{-1/2}\epsilon^{1/2}, \frac{1}{2}{d_i}^{-1/3}\epsilon^{1/3}\right\}. \tag{3.32}$$

Proof. For notational simplicity, in this proof, we use s_c to represent $s_c^{(i)}$. Since s_c is the global minimum of M_c, we have $\nabla M_c(s^{(i)}, s_c) = 0$. Thus,

$$\epsilon < \|\nabla m_3(s^{(i)} + s_c)\| = \|\nabla M(s^{(i)}, s_c)\| = \|\nabla M_c(s^{(i)}, s_c) - \nabla M(s^{(i)}, s_c)\|.$$

Using the expression of ∇M_c in (3.3) and $\nabla M(s^{(i)}, s) = g_i + H_i s + \frac{1}{2}T_i[s]^2 + \sigma\|s\|^2 s$, we have

$$\epsilon < \left\|\frac{1}{2}\beta_i\|s_c\|s_c - \frac{1}{2}T_i[s_c]^2 + 4d_i\|s_c\|^2 s_c\right\| \leq \frac{B}{2}\|s_c\|^2 + \frac{L_H}{2}\left\|s_c\right\|^2 + 4d_i\|s_c\|^3 \tag{3.33}$$

where the last inequality uses norm properties, $\beta_i > -B$ from Assumption 3.3.1 and $\left\|T_i\right\| \leq L_H$ from Corollary 3.3.1. If $d_i \geq 0$, either one of the following must be true

$$\frac{\epsilon}{2} < \left(\frac{B}{2} + \frac{L_H}{2}\right)\|s_c\|^2 \Rightarrow \|s_c\| > (B + L_H)^{-1/2}\epsilon^{1/2}, \quad \text{or} \quad \frac{\epsilon}{2} < 4d_i\|s_c\|^3 \Rightarrow \|s_c\| > \frac{1}{2}{d_i}^{-1/3}\epsilon^{1/3}$$

which gives (3.32). □

Lemma 3.3.1 demonstrates that we can establish a lower bound for the norm of the iterates $s^{(i)}$. The lower bound on step size plays a crucial role in guaranteeing a decrease in values of m_3. The proof incorporates standard techniques found in [23, Lemma 2.3] and in [53, Lemma 3.3], which demonstrate that the step cannot be arbitrarily small compared to the criticality conditions. It is important to note that Lemma 3.3.1 applies for both successful and unsuccessful iterations provided that the criticality conditions are not met, i.e., $\|\nabla m_3(s^{(i)} + s_c^{(i)})\| > \epsilon$ and $\|g_i\| \geq \epsilon$.

Theorem 3.3.2 provides an upper bound for m_3 at $m_3(s^{(i)} + s_c^{(i)}) = M(s^{(i)}, s_c^{(i)})$ where m_3 and M are defined in (Quartically Regularized Cubic Model) and (2.1), respectively.

Theorem 3.3.2. ***(A local upper bound for*** m_3***)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is used. Then, the following results are true.*

1. ***(Case 1)*** *If* $\beta_i > L_H$*, for any* $d_i \geq 0$*, we have* $m_3(s^{(i)} + s_c^{(i)}) = M(s^{(i)}, s_c^{(i)}) \leq M_c(s^{(i)}, s_c^{(i)})$.
2. ***(Case 2)*** *If any* $\beta_i \leq L_H$ *(*β_i *can be negative), for*

$$d_i > \frac{B + L_H}{6}\|s_c^{(i)}\|^{-1} > 0, \tag{3.34}$$

we have $m_3(s^{(i)} + s_c^{(i)}) = M(s^{(i)}, s_c^{(i)}) \leq M_c(s^{(i)}, s_c^{(i)})$.

In both cases, the equality only happens at $s_c^{(i)} = 0$.

Proof. For notational simplicity, in this proof, we use s_c to represent $s_c^{(i)}$. By definition of M_c and M, we have

$$M_c(s^{(i)}, s_c) - M(s^{(i)}, s_c) = \frac{1}{6}\left[\beta_i\|s_c\|^3 - T_i[s_c]^3\right] + d_i\|s_c\|^4. \tag{3.35}$$

If $s_c^{(i)} = 0$, clearly $M(s^{(i)}, 0) = M_c(s^{(i)}, 0)$. Assume that $\|s_c^{(i)}\| > 0$.

[4] If $d_i = 0$, $\|s_c^{(i)}\| > (B + L_H)^{-1/2}\epsilon^{1/2}$.

(**Case 1)** If $\beta_i \geq L_H$, for any $d_i \geq 0$, we deduce from (3.35) that

$$M_c(s^{(i)}, s_c) - M(s^{(i)}, s_c) \geq \frac{1}{6}\Big[\underbrace{L_H - \frac{T_i[s_c]^3}{\|s_c\|^3}}_{>0 \text{ by Corollary 3.3.1}} \Big]\|s_c\|^3 + d_i\|s_c\|^4 > 0$$

(**Case 2)** If $\beta_i < L_H$ (β_i can be negative) and let $d_i > \frac{1}{6}(B + L_H)\|s_c\|^{-1} > 0$, using $\beta_i > -B$ from Assumption 3.3.1 and $T[s_c]^3 \geq -L_H\|s_c\|^3$ from Corollary 3.3.1, we deduce from (3.35) that

$$M_c(s^{(i)}, s_c) - M(s^{(i)}, s_c) > \frac{1}{6}\big(- B - L_H\big)\|s_c\|^3 + d_i\|s_c\|^4 \underset{(3.34)}{\geq} 0. \tag{3.36}$$

□

Remark 3.3.2. *If d_i satisfies (3.34) in Theorem 3.3.2, then $\sigma_c^{(i)}$ satisfies the lower bound on $\sigma_c^{(i)} = \sigma + 4d_i$ in Algorithm 3.3.2, as given in (3.31). This is because if $d_i > \frac{1}{6}(B + L_H)\|s_c^{(i)}\|^{-1}$, using $B > -\beta_i$ and $L_H > 1 > \alpha > 0$, we have $d_i > \frac{1}{6}\big(- \beta_i + \alpha\big)\|s_c^{(i)}\|^{-1}$ and, consequently, $\sigma_c^{(i)} = \sigma + 4d_i \geq \frac{2}{3}\big(- \beta_i + \alpha\big)\|s_c^{(i)}\|^{-1}$.*

Theorem 3.3.2 provides some intuition about how β_i approximates the third-order information. In particular, if β_i is positive and the $\beta_i\|s\|^3$ term sufficiently regularizes the third-order information, then any $d_i \geq 0$ will ensure M_c stays above m_3. On the other hand, when β_i is negative, we need to ensure that the regularization term of M_c is large enough to guarantee $m_3(s^{(i)} + s_c^{(i)}) \leq M_c(s^{(i)}, s_c^{(i)})$. Therefore, in Case 2, we increase d_i until condition (3.34) is satisfied. This procedure is well-defined since $\|s_c^{(i)}\|$ scales as $d_i^{-1/3}$ as d_i increases. An explicit upper bound on the required value of d_i is established in Lemma 3.3.2.

We define the sets of successful and unsuccessful iterations for Algorithm 3.3.2 as $\mathcal{S}_i$ and $\mathcal{U}_i := [0 : i]\backslash\mathcal{S}_i$. Notice that for $i \in \mathcal{S}_i$, $\rho_j \geq \eta$ and $s^{(i+1)} = s^{(i)} + s_c^{(i)}$, while $s^{(i+1)} = s^{(i)}$ for $i \in \mathcal{U}_i$.

Lemma 3.3.2. ***(An upper bound for d_i)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is used. Then, for all $i \geq 0$*

$$d_i \leq d_{\max} := \gamma(B + L_H)^{3/2}\epsilon^{-1/2}, \tag{3.37}$$

where γ is a fixed constant from Algorithm 3.3.1.

Proof. **(Case 1)** If $\beta_i \geq L_H$, for any $d_i \in [0, d_{\max}]$, according to Theorem 3.3.2, we have

$$m_3(s^{(i)}) - m_3(s^{(i)} + s_c^{(i)}) = m_3(s^{(i)}) - M(s^{(i)}, s_c^{(i)}) \geq m_3(s^{(i)}) - M_c(s^{(i)}, s_c) > 0.$$

Therefore, $\rho_i := \frac{m_3(s^{(i)}) - m_3(s^{(i)} + s_c^{(i)})}{m_3(s^{(i)}) - M_c(s^{(i)}, s_c^{(i)})} \geq 1 \geq \eta$. The iteration i is successful.
(Case 2) If $\beta_i < L_H$ (β_i can be negative) and suppose that

$$d_i > (B + L_H)^{3/2}\epsilon^{-1/2}.$$

We deduce that $d_i = d_i^{2/3}d_i^{1/3} \geq (B + L_H)\epsilon^{-1/3}d_i^{1/3}$. This implies the bound

$$d_i \geq (B + L_H)\max\Big\{(B + L_H)^{1/2}\epsilon^{-1/2}, {d_i}^{1/3}\epsilon^{-1/3}\Big\} \underset{(3.32)}{\geq} (B + L_H)\|s_c^{(i)}\|^{-1} \tag{3.38}$$

where the last inequality follows from the upper bound on $\|s_c^{(i)}\|$ in Lemma 3.3.1. Since (3.38) satisfies the requirement on d_i in Theorem 3.3.2, we deduce that $m_3(s^{(i)} + s_c^{(i)}) \leq M_c(s^{(i)}, s_c^{(i)})$. The rest of the analysis follows similarly as in Case 1. We deduce that the iteration i is successful.

In both cases, it follows from the update of regularization parameters in Algorithm 3.3.2 that the next iteration $0 \leq d_{i+1} \leq d_i$. As a consequence, the mechanism of regularization parameters update for Algorithm 3.3.2 ensure that (3.37) holds. □

Remark 3.3.3. *It is worth noting that* (3.31) *in Algorithm 3.3.2 is below the upper bound* $d_{\max}$*; namely, there is no contradiction between* (3.31) *and* (3.37)*. To see this, assume instead that*

$$\sigma_c^{(i)} = \sigma + 4d_i > \frac{2}{3}\big(-\beta_i + \alpha\big)\|s_c^{(i)}\|^{-1} \geq \sigma + 4d_{\max}.$$

This implies $\frac{1}{6}\big(-\beta_i+\alpha\big)\|s_c^{(i)}\|^{-1} \geq d_{\max}$*. However, using* $0 < \alpha < 1 \leq L_H$*, the lower bound for* $\|s_c^{(i)}\|$ *in Lemma 3.3.1,* $d_{\max}$ *in* (3.37)*,* $\gamma > 1$ *and* $|\beta_i| < B$*, we deduce that*

$$\begin{aligned}
\frac{1}{6}\big(-\beta_i+\alpha\big)\|s_c^{(i)}\|^{-1} &< \frac{1}{6}(B+\alpha)\max\left\{(B+L_H)^{1/2}\epsilon^{-1/2}, 2{d_i}^{1/3}\epsilon^{-1/3}\right\} \\
&< \frac{1}{3}\max\left\{\frac{1}{2}\underbrace{(B+L_H)^{3/2}\epsilon^{-1/2}}_{=\gamma^{-1}d_{\max}}, \underbrace{(B+L_H)\epsilon^{-1/3}}_{=\gamma^{-2/3}d_{\max}^{2/3}}{d_i}^{1/3}\right\} \\
&< \frac{1}{3}\max\left\{(2\gamma)^{-1}, \gamma^{-2/3}\right\}d_{\max} < d_{\max}
\end{aligned}$$

which is a contradiction.

The intuition behind Lemma 3.3.2 is that whenever the regularization parameter exceeds the bound in (3.37), the CQR model $M_c(s^{(i)}, s)$ overestimates m_3 at its minimizer $s = s_c^{(i)}$ (Theorem 3.3.2). Therefore, any decrease obtained in minimizing $M_c(s^{(i)}, s)$ is a lower bound on the decrease in m_3, ensuring that every iteration is successful, and consequently removing the need to increase the regularization parameter any further.

Theorem 3.3.3. ***(Bounding the*** m_3 ***decrease on successful iterations)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is used. For* $\alpha \in (0, \frac{1}{2})$ *and* $\eta \in (0, 1]$*, assume that* $i+1$ *is a successful iteration which does not meet the termination condition, i.e.,* $\|g_{i+1}\| > \epsilon$ *and* $\|g_i\| > \epsilon$*. Then,*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\|s_c^{(i)}\|^3 \geq \kappa_s \epsilon^{\frac{3}{2}} \tag{3.39}$$

where $\kappa_s = \frac{\alpha\eta}{192\gamma}(B+L_H)^{-3/2}$*.*

Proof. For notational simplicity, in this proof, we use s_c to represent $s_c^{(i)}$ and σ_c to represent $\sigma_c^{(i)}$. Since $i \in \mathcal{S}_i$, (CQR Ratio Test) gives

$$\eta^{-1}\left[m_3(s^{(i)}) - m_3(s^{(i+1)})\right] \geq M_c(s^{(i)}, 0) - M_c(s^{(i)}, s_c) = -g_i^T s_c - \frac{1}{2}H_i[s_c]^2 - \frac{\beta_i}{6}\|s_c\|^3 - \frac{\sigma_c}{4}\|s_c\|^4$$

where $s_c^{(i)}$ is obtained by Algorithm 3.3.1 and is the global minimum of M_c. Then, $s_c^{(i)}$ satisfies (3.18) and (3.19), which implies

$$M_c(s^{(i)}, 0) - M_c(s^{(i)}, s_c) \underset{(3.18)}{=} \frac{1}{2}H_i[s_c]^2 + \frac{\beta_i}{3}\|s_c\|^3 + \frac{3\sigma_c}{4}\|s_c\|^4 \underset{(3.19)}{\geq} \frac{\beta_i}{12}\|s_c\|^3 + \frac{\sigma_c}{4}\|s_c\|^4. \tag{3.40}$$

- Assuming $\beta_i \in [\alpha, B]$, since $\sigma_c > 0$, we deduce from (3.40) that

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\beta_i\eta}{12}\|s_c\|^3 + \frac{\sigma_c\eta}{4}\|s_c\|^4 > \frac{\alpha\eta}{12}\|s_c\|^3.$$

- Assuming $\beta_i \in [-B, \alpha]$ and $\sigma_c \geq \frac{2}{3}\big(-\beta_i+\alpha\big)\|s_c\|^{-1} > 0$, we deduce from (3.40) that

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\beta_i\eta}{12}\|s_c\|^3 + \frac{\eta}{6}\big(-\beta_i+\alpha\big)\|s_c\|^3 \geq \left(-\frac{\beta_i}{12} + \frac{\alpha}{6}\right)\eta\|s_c\|^3 \geq \frac{\alpha\eta}{12}\|s_c\|^3.$$

  where the last inequality follows from $-\beta_i \geq -\alpha$ for $\beta_i \in [-\alpha, \alpha]$ and $-\beta_i \geq \alpha$ for $\beta_i \in [-B, -\alpha]$.

- Assuming $\beta_i \in [-B, -4\alpha]$ and $0 < \sigma_c \leq \frac{1}{6}\big(-\beta_i + \alpha\big)\|s_c\|^{-1}$: Since s_c is a global minimum, s_c is a local minimum. According to the second-order local optimality condition (3.4), $0 \leq \nabla^2 M_c(s^{(i)}, s_c)[s_c]^2 = H_i[s_c]^2 + \beta_i\|s_c\|^3 + 3\sigma_c\|s_c\|^4$. Substituting the local optimality condition into the middle equation in (3.40), we have

$$M_c(s^{(i)}, 0) - M_c(s^{(i)}, s_c) \geq -\frac{\beta_i}{6}\|s_c\|^3 - \frac{3}{4}\sigma_c\|s_c\|^4. \tag{3.41}$$

  Using $0 < \sigma_c \leq \frac{1}{6}\big(-\beta_i + \alpha\big)\|s_c\|^{-1}$, we deduce from (3.41) that

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq -\frac{\eta\beta_i}{6}\|s_c\|^3 - \frac{\eta}{8}\big(-\beta_i + \alpha\big)\|s_c\|^3 = \left(-\frac{\beta_i}{24} - \frac{\alpha}{8}\right)\eta\|s_c\|^3 \geq \frac{\alpha\eta}{24}\|s_c\|^3$$

  where the last inequality uses $\beta_i \leq -4\alpha$.

We deduce from (Case 1) that, in a successful step, if we have $\beta_i \in [\alpha, B]$ and $\sigma_c > 0$, then

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\|s_c\|^3. \tag{3.42}$$

Also, we can deduce from (Case 2) and (Case 3) that if $\beta_i \in [-B, -4\alpha]$ and $\sigma_c \geq \frac{2}{3}\big(-\beta_i + \alpha\big)\|s_c\|^{-1}$ or $\sigma_c \leq \frac{1}{6}\big(-\beta_i + \alpha\big)\|s_c\|^{-1}$, then (3.42) also holds. Lastly, using (Case 3), if $\beta_i \in [-4\alpha, \alpha]$ and $\sigma_c \geq \frac{2}{3}\big(-\beta_i + \alpha\big)\|s_c\|^{-1}$, then (3.42) also holds. These three cases correspond to the requirements of β_i and σ_c in Algorithm 3.3.2. Using the lower bound on $\|s_c\|$ in Lemma 3.3.1 and the bound on d_i in Lemma 3.3.2,

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\max\left\{(B + L_H)^{-3/2}\epsilon^{\frac{3}{2}}, \frac{1}{8}{d_{\max}}^{-1}\epsilon\right\} \geq \frac{\alpha\eta}{24}\min\left\{1, \frac{1}{8\gamma}\right\}(B + L_H)^{-3/2}\epsilon^{\frac{3}{2}}.$$

Using $\gamma > 1$, gives the desired result in (3.39). □

Remark 3.3.4. *It is important to note that the $\sigma_c^{(i)}$ condition in Algorithm 3.3.2 is meant to ensure that the decrease in values of m_3 is at least $\mathcal{O}(\epsilon^{\frac{3}{2}})$, specifically: $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}(B + L_H)^{-3/2}\epsilon^{\frac{3}{2}}$. If we only have the condition $\rho_i \geq \eta$, the CQR algorithmic framework (Algorithm 3.3.1) still results in a decrease in value of m_3 at each successful iteration. This is evident from the sufficient condition for a global minimizer of M_c in Theorem 3.2.2, from $\beta \geq -3\sigma_c\|s_c^{(i)}\|$, we deduce that the right-hand side of (3.40) is positive. We can also verify the decrease in value of m_3 with the following argument. Assume $\sigma_c^{(i)} > 0$; the eigenvalue of H_i is bounded for all $i \geq 0$,*

$$\begin{aligned}\lambda_{\max}(H_i) &= \max_{u\in\mathbb{R}^n, \|u\|=1}\left(H + T[s^{(i)}] + \sigma\|s^{(i)}\|^2 + 2\sigma(s^{(i)})^T s^{(i)}\right)[u]^2 \\ &\leq H + \Lambda_0\|s^{(i)}\| + 3\sigma\|s^{(i)}\|^2 \leq \|H\| + \Lambda_0 r_c + 3\sigma r_c^2 =: L_g\end{aligned}$$

where $\|H\| = \lambda_{\max}[H]$ and Λ_0 given in Definition 2.3.2. Assume that $|\beta_i| \leq B$ and set $v := -\mu\frac{g_i}{\|g_i\|}$ for any scalar $\mu > 0$. Since $s_c^{(i)}$ is the global minimizer, we have

$$\begin{aligned}&M_c(s^{(i)}, 0) - M_c(s^{(i)}, s_c^{(i)}) \geq f_i - M_c(s^{(i)}, v) = \mu\|g_i\| - \frac{\mu^2}{2}\frac{g_i^T H_i g_i}{\|g_i\|^2} - \frac{\beta_i}{6}\mu^3 - \frac{\sigma_c^{(i)}}{4}\mu^4 \\ &> \mu\epsilon - \frac{L_g}{2}\mu^2 - \frac{B}{6}\mu^3 - \frac{\sigma_c^{(i)}}{4}\mu^4 = \mu\left(\frac{\epsilon}{3} - \frac{L_g}{2}\mu\right) + \mu\left(\frac{\epsilon}{3} - \frac{B}{6}\mu^2\right) + \mu\left(\frac{\epsilon}{3} - \frac{\sigma_c^{(i)}}{4}\mu^3\right).\end{aligned} \tag{3.43}$$

Since $\epsilon \in (0, 1)$, we can choose μ such that

$$\frac{1}{2}\min\left\{\frac{2}{3L_g}, \left(\frac{2}{B}\right)^{1/2}, \left(\frac{4}{3\sigma_c^{(i)}}\right)^{1/3}\right\}\epsilon < \mu < \frac{1}{2}\min\left\{\frac{2}{3L_g}\epsilon, \left(\frac{2}{B}\epsilon\right)^{1/2}, \left(\frac{4}{3\sigma_c^{(i)}}\epsilon\right)^{1/3}\right\}.$$

Then, every term in the bracket of (3.43) is negative, and we have

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \eta \left[M_c(s^{(i)}, 0) - M_c(s^{(i)}, s_c^{(i)}) \right] \geq \frac{\epsilon \mu \eta}{6} > \frac{\epsilon^2 \eta \check{c}}{6} > 0$$

where $\check{c} = \frac{1}{2} \min \left\{ \frac{2}{3L_g}, (\frac{2}{B})^{1/2}, (\frac{4}{3\sigma_c^{(i)}})^{1/3} \right\}$. *This analysis is unaffected by* β_i *being positive or negative and does not require the* $\sigma_c^{(i)}$ *condition in Algorithm 3.3.2.*

We may now focus on how many successful iterations may be needed to compute an ϵ-approximate first-order minimizer using Algorithm 3.3.2.

Lemma 3.3.3. ***(Successful and unsuccessful adaptive-regularization iterations)*** *Suppose that Assumption 3.3.1 holds and Algorithm 3.3.2 is used. Then,*

$$i \leq |\mathcal{S}_i| + \frac{\log(d_{\max})}{\log \gamma}.$$

Lemma 3.3.3 employs a similar technique to that in [44, Lemma 2.3.1, Lemma 2.4.1], and the proof is provided in detail in Section A.1.2. Note that Lemma 3.3.3 is entirely independent of the form of the model M_c and depends on the mechanism defined by the update of regularization parameters in Algorithm 3.3.2. We do not need Assumption 3.3.1 for Lemma 3.3.3.

Lemma 3.3.4. ***(Bound on the number of successful iterations)*** *Suppose that Assumption 3.3.1 holds. Let* m_{low} *be a lower bound on* $m_3(s)$ *for* $s \in \mathbb{R}^n$. *Then, there exists a positive constant* κ_s *as defined in Theorem 3.3.3 such that Algorithm 3.3.2 requires at most*

$$|\mathcal{S}_i| \leq \frac{m_3(s^{(0)}) - m_{low}}{\kappa_s \epsilon^{\frac{3}{2}}} \tag{3.44}$$

successful iterations before an iterate s_ϵ *is computed for which* $\|g_i\| \leq \epsilon$.

Proof. Suppose that $\|g_j\| > \epsilon$ and $\|g_{j+1}\| > \epsilon$ for all $j \in [1 : i]$. Then,

$$m_3(s^{(0)}) - m_{\text{low}} \geq m_3(s^{(0)}) - m_3(s^{(i+1)}) = \sum_{j \in \mathcal{S}_i} \left[m_3(s^{(i)}) - m_3(s^{(i+1)}) \right] > |\mathcal{S}_i| \kappa_s \epsilon^{\frac{3}{2}}$$

where κ_s given in Theorem 3.3.3. This implies the bound (3.44). □

Given Lemma 3.3.4, we are now ready to state the worst-case evaluation bound for Algorithm 3.3.2.

Theorem 3.3.4. ***(Complexity bound for Algorithm 3.3.2)*** *Suppose that Assumption 3.3.1 holds. Then, there exists a positive constant* κ_s *(see Theorem 3.3.3),* κ_c *such that Algorithm 3.3.2 requires at most*

$$\frac{m_3(s^{(0)}) - m_{low}}{\kappa_s \epsilon^{\frac{3}{2}}} + \kappa_c + \log \epsilon^{-1/2} = \mathcal{O}(\epsilon^{-\frac{3}{2}}).$$

function evaluations (i.e., f_i*) and at most*

$$\frac{m_3(s^{(0)}) - m_{low}}{\kappa_s \epsilon^{\frac{3}{2}}} + 1 = \mathcal{O}(\epsilon^{-\frac{3}{2}}).$$

derivatives evaluations (i.e., g_i, H_i, T_i*) to compute an iterate* s_ϵ *such that* $\|g_i\| \leq \epsilon$ *or* $\|g_{i+1}\| \leq \epsilon$.

Proof. The number of successful iterations needed to find an ϵ-approximate first-order minimizer is bounded above by (3.44). The total number of iterations (including unsuccessful ones) can also be bounded above using Lemma 3.3.3, which yields that this number is at most

$$\frac{m_3(s^{(0)}) - m_{\text{low}}}{\kappa_s \epsilon^{\frac{3}{2}}} + \frac{\log(d_{\max})}{\log \gamma} \underset{(3.37)}{=} \frac{m_3(s^{(0)}) - m_{\text{low}}}{\kappa_s \epsilon^{\frac{3}{2}}} + 1 + \frac{3 \log(B + L_H)}{2 \log \gamma} + \log \epsilon^{-1/2}.$$

where we use $d_{\max} = \gamma(B + L_H)^{3/2} \epsilon^{-1/2}$. Since Algorithm 3.3.2 uses at most one evaluation of m_3 per iteration and at most one evaluation of its derivatives per successful iteration (plus one evaluation of m_3 and its derivatives at the final iteration), by setting $\kappa_c := 1 + \frac{3 \log(B+L_H)}{2 \log \gamma}$, we deduce the desired conclusion. □

Improved Complexity When the Tensor Term is Small: Although Algorithm 3.3.2 achieves a function value reduction of $\mathcal{O}(\epsilon^{\frac{3}{2}})$ for a general nonconvex m_3, in certain special cases, the CQR algorithm or its variants exhibit improved complexity bounds during specific iterations. One such instance is when the tensor term is sufficiently small. If $\|T_i\| \leq \epsilon^{1/3}$, and we choose $|\beta_i| \leq \epsilon^{1/3}$, then the decrease in value of m_3 in such an iteration is $\mathcal{O}(\epsilon^{\frac{4}{3}})$. To see this, note that since $|\beta_i| \leq \epsilon^{1/3}$, (3.33) becomes

$$\epsilon < \frac{\epsilon^{1/3}}{2}\|s_c\|^2 + \frac{\epsilon^{1/3}}{2}\|s_c\|^2 + 4d_i\|s_c\|^3.$$

Thus, Lemma 3.3.1 can be improved to:

$$\|s_c^{(i)}\| > \min\left\{2^{-1/2}, \frac{1}{2}d_i^{-1/3}\right\}\epsilon^{1/3}. \tag{3.45}$$

Since $B \leq \epsilon^{1/3}$ and $L_H \leq \epsilon^{1/3}$, (3.36) becomes

$$M_c(s^{(i)}, s_c) - M(s^{(i)}, s_c) > -\frac{1}{3}\epsilon^{1/3}\|s_c\|^3 + d_i\|s_c\|^4 > 0.$$

Additionally, (3.34) in Theorem 3.3.2 gives

$$d_i > \frac{\epsilon^{1/3}}{3}\|s_c^{(i)}\|^{-1} \geq \frac{1}{3}\max\left\{2^{1/2}, 2d_i^{1/3}\right\}. \tag{3.46}$$

Note that the last inequality is derived by using (3.45). Therefore, following the same analysis as in Theorem 3.3.2, we deduce that an upper bound for d_i satisfies $d_i \leq d_{\max} := 2\sqrt{2}$. Consequently, applying the analysis from the proof of Theorem 3.3.3, we conclude that the reduction in the value of m_3 during this iteration is $\mathcal{O}(\epsilon^{\frac{4}{3}})$.

3.3.2 A CQR Variant in the Presence of Separability

We start with a useful definition.

Definition 3.3.1. *$H_i \in \mathbb{R}^{n^2}$ and $T_i \in \mathbb{R}^{n^3}$ are jointly diagonalizable if there exists a non-singular matrix $\check{Q}_i \in \mathbb{R}^{n^2}$, such that*

- *$\check{Q}_i^T H_i \check{Q}_i = D_i = \text{diag}\{h_j^{(i)}\}_{1\leq j\leq n}$ is a $n \times n$ diagonal matrix.*
- *$T_i[\check{Q}_i]^3 = \mathcal{T}_i = \text{diag}\left\{t_j^{(i)}\right\}_{1\leq j\leq n}$ is a $n \times n \times n$ third-order diagonal tensor with diagonal entries $\{t_1^{(i)}, \ldots, t_n^{(i)}\}$ and other entries as 0.*

If the second-order term and third-order term in (2.1) are jointly diagonalizable in an iteration, we construct a variant of the CQR algorithm that uses univariate polynomials minimization and improves the guaranteed function-value reduction in such an iteration.

Remark 3.3.5. *We note that $\check{Q}_i$ can be non-symmetric and non-orthogonal. There are many ways to test whether H_i and T_i are jointly diagonalizable. A straightforward scenario occurs when both H_i and T_i are diagonal, in which case we can set $\check{Q}_i = I_n$. Another approach involves performing an eigenvalue decomposition on H_i to obtain $\check{Q}_i$, and then checking whether $T_i[\check{Q}_i]^3$ is diagonal tensor. However, since $\check{Q}_i$ is not required to be orthogonal nor symmetric, the case of joint diagonalisability encompasses more scenarios than merely H_i and T_i sharing the same eigenvalue decomposition; see Section A.1.5 for more examples.*

In an iteration, if H_i and T_i are jointly diagonalizable with the common basis $\check{Q}_i$, in contrast to (3.1), we construct the separable CQR model using a different fourth-order regularization norm

$$\hat{M}_c(s^{(i)}, s) := \underbrace{f_i + g_i^T s + \frac{1}{2}H_i[s]^2 + \frac{1}{6}T_i[s]^3}_{=T_3(s^{(i)},s)} + \frac{\sigma + d_i}{4}\|\check{Q}_i^{-1}s\|_4^4. \tag{3.47}$$

Here, g_i, H_i, T_i, and $T_3(s^{(i)}, s)$ are defined as in (2.1), and $d_i \geq 0$. The regularization norm is defined as $\|\tilde{s}\|_4 = (\sum_{j=1}^n \tilde{s}_j^4)^{\frac{1}{4}}$ for $\tilde{s} = [\tilde{s}_1, \ldots, \tilde{s}_n]^T$. The aim is to use $\hat{M}_c(s^{(i)}, s)$ to approximate $M(s^{(i)}, s)$ in (2.1). The first three terms of $\hat{M}_c(s^{(i)}, s)$ and $M(s^{(i)}, s)$ are the same; the only difference lies in the regularization term. Using the triangle inequality and Hölder's inequality, we relate the two regularization norms by

$$\|s\| \leq \|\check{Q}_i\| \|\check{Q}_i^{-1} s\| \leq n^{\frac{1}{4}} \|\check{Q}_i\| \|\check{Q}_i^{-1} s\|_4. \tag{3.48}$$

When $n = 1$ and $\check{Q}_i = 1$, the two regularization norms are equal, $\|\check{Q}_i^{-1} s\|_4 = \|s\| = |s|$, while setting $d_i = 0$, we have $\hat{M}_c(s^{(i)}, s) = M(s^{(i)}, s)$.

The advantage of the separable CQR model in (3.47) is that through a change of basis, (3.47) becomes entirely separable into n univariate polynomials with quartic regularizations, denoted by $\hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$ for $j = 1, \ldots, n$. We convert the minimization problem in (3.47), $\operatorname{argmin}_{s \in \mathbb{R}^n} \hat{M}_c(s^{(i)}, s)$, into minimizing n univariate polynomials, $\operatorname{argmin}_{s_j \in \mathbb{R}} \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$ for $j = 1, \ldots, n$.

Existing literature on the use of the $\|\cdot\|_4$ regularization norm includes [34], where efficient methods for minimizing convex quartic polynomials with this norm are presented. Additionally, [2] examines the relationship between the $\|\check{Q}_i^{-1} s\|_4$ and various polynomial norms. Besides our work, there are also other methods available for identifying and finding the approximate global minimum of univariate polynomials with quartic regularization $\hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$, as discussed in [9; 36; 164].

A CQR algorithm in the presence of separability is given in Algorithm 3.3.3.

Algorithm 3.3.3: A CQR Variant of Algorithm 3.3.1 in the Presence of Separability

Initialize and proceed through Step 1 following the same procedure as outlined in the CQR algorithmic framework (Algorithm 3.3.1). At each iteration, after Step 1, we add a step to test for separability.

Step 1b: Test for separability. If H_i and T_i are jointly diagonalizable with the change of variable matrix $\check{Q}_i$ (see Remark 3.3.5), proceed to Steps 2b-3b and then Step 4 in Algorithm 3.3.1; Else, proceed to Steps 2-4 in Algorithm 3.3.2.

Step 2b: Step computation. Compute $\check{Q}_i^T H_i \check{Q}_i = \operatorname{diag}\{h_j^{(i)}\}_{1 \leq j \leq n}$, $T_i[\check{Q}_i]^3 = \operatorname{diag}\left\{t_j^{(i)}\right\}_{1 \leq j \leq n}$ and $\tilde{g}_i = Q^{-T} g_i = [\tilde{g}_1^{(i)}, \ldots, \tilde{g}_n^{(i)}]^T \in \mathbb{R}^n$. Construct the subproblem as n univariate polynomials with quartic regularization

$$\hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j) := \tilde{g}_j^{(i)} \tilde{s}_j + \frac{h_j^{(i)}}{2} \tilde{s}_j^2 + \frac{t_j^{(i)}}{6} \tilde{s}_j^3 + \frac{\sigma + d_i}{4} \tilde{s}_j^4 \qquad \text{for} \qquad j = 1, \ldots, n. \tag{3.49}$$

For each $j = 1, \ldots, n$, compute $\tilde{s}_{c,j}^{(i)} = \operatorname{argmin}_{s_j \in \mathbb{R}} \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$. Set $\tilde{s}_c^{(i)} = [\tilde{s}_{c,1}^{(i)}, \ldots, \tilde{s}_{c,n}^{(i)}]^T$ and recover $s_c^{(i)} = \check{Q}_i \tilde{s}_c^{(i)}$.

Step 3b: Same as Step 3 in Algorithm 3.3.1 but with M_c in (CQR Ratio Test) changed to T_3.

In this section, we prove that, by adaptively adjusting the regularization term d_i, the successful iteration obtained from Algorithm 3.3.3 results in a function value reduction of at least $\mathcal{O}(\epsilon^{\frac{4}{3}})$.

We first use Lemma 3.3.5 to establish the link between the separable CQR model in (3.47) and the set of univariate polynomials in (3.49).

Lemma 3.3.5. *Assume that H_i and T_i are jointly diagonalizable with joint basis $\check{Q}_i$ such that $\check{Q}_i^T H_i \check{Q}_i = D_i = \operatorname{diag}\{h_j^{(i)}\}_{1 \leq j \leq n}$ and $T_i[\check{Q}_i]^3 = \mathcal{T}_i = \operatorname{diag}\left\{t_j^{(i)}\right\}_{1 \leq j \leq n}$. Let*

$$\hat{\mathcal{M}}_c(s^{(i)}, \tilde{s}) := f_i + g_i^T s + \frac{1}{2} D_i[\tilde{s}]^2 + \frac{1}{6} \mathcal{T}_i[\tilde{s}]^3 + \frac{\sigma + d_i}{4} \|\tilde{s}\|_4^4 = f_i + \sum_{j=1}^n \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j) \tag{3.50}$$

where $\tilde{s} = \check{Q}_i^{-1} s = [\tilde{s}_1, \ldots, \tilde{s}_j]^T$, $\tilde{g}_i = Q^{-T} g_i = [\tilde{g}_1^{(i)}, \ldots, \tilde{g}_n^{(i)}]^T \in \mathbb{R}^n$, and $\hat{\mathcal{M}}_{c,j}$ is defined in (3.49). Then, $\hat{M}_c(s^{(i)}, s) = \hat{\mathcal{M}}_c(s^{(i)}, \tilde{s})$. Let $s_c^{(i)} := \operatorname{argmin}_{s \in \mathbb{R}^n} \hat{M}_c(s^{(i)}, s)$, $\tilde{s}_{c,j}^{(i)} = \operatorname{argmin}_{s_j \in \mathbb{R}} \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$ and $\tilde{s}_c^{(i)} = [\tilde{s}_{c,1}^{(i)}, \ldots, \tilde{s}_{c,n}^{(i)}]^T$. Then, $s_c^{(i)} = \check{Q}_i \tilde{s}_c^{(i)}$.

Proof. Using the change of variable $\tilde{s} = \check{Q}_i^{-1} s = [\tilde{s}_1, \ldots, \tilde{s}_j]^T$, we have

$$\begin{aligned} \hat{M}_c(s^{(i)}, s) &= f_i + g_i^T \check{Q}_i \tilde{s} + \frac{1}{2}\check{Q}_i^T H_i \check{Q}_i [\tilde{s}]^2 + \frac{1}{6} T_i [\check{Q}_i]^3 [\tilde{s}]^3 + \frac{\sigma + d_i}{4} \|\tilde{s}\|_4^4 \\ &= \hat{\mathcal{M}}_c(s^{(i)}, \tilde{s}) = f_i + \sum_{j=1}^{n} \Big[\underbrace{\tilde{g}_j^{(i)} \tilde{s}_j + \frac{h_j^{(i)}}{2} \tilde{s}_j^2 + \frac{t_j^{(i)}}{6} \tilde{s}_j^3 + \frac{\sigma_c^{(i)}}{4} \tilde{s}_j^4}_{:= \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)} \Big]. \end{aligned}$$

By change of basis, $s_c^{(i)} = \check{Q}_i \tilde{s}_c^{(i)}$ where $\tilde{s}_c^{(i)} := \operatorname{argmin}_{\tilde{s} \in \mathbb{R}} \hat{\mathcal{M}}_c(s^{(i)}, \tilde{s})$. Since each $\tilde{s}_j$ is separable, the minimization of $\hat{M}_c(s^{(i)}, s)$ can be rewritten as the minimization of n univariate polynomials and the jth entry of $\tilde{s}_c^{(i)}$ can be obtained by $\tilde{s}_{c,j}^{(i)} = \operatorname{argmin}_{s_j \in \mathbb{R}} \hat{\mathcal{M}}_{c,j}(s^{(i)}, \tilde{s}_j)$. □

Lemma 3.3.6. *Let* $\hat{\sigma}_c^{(i)} := \sigma + d_i$. *If*

$$\hat{\sigma}_c^{(i)} \geq \frac{\sigma n \|\check{Q}_i\|^4}{(1-\eta)}, \tag{3.51}$$

then, $\rho_i = \frac{m_3(s^{(i)}) - m_3(s^{(i)} + s_c^{(i)})}{m_3(s^{(i)}) - T_3(s^{(i)}, s_c^{(i)})} \geq \eta$ *and iteration* i *is successful.*

Proof. Since $s_c^{(i)} = \operatorname{argmin}_{s \in \mathbb{R}^n} \hat{M}_c(s^{(i)}, s)$, then

$$m_3(s^{(i)}) - T_3(s^{(i)}, s_c^{(i)}) - \frac{\hat{\sigma}_c^{(i)}}{4} \|\tilde{s}\|_4^4 = m_3(s^{(i)}) - \hat{M}_c(s^{(i)}, s_c^{(i)}) \geq 0. \tag{3.52}$$

We deduce that

$$|\rho_i - 1| \leq \frac{|m_3(s^{(i)} + s_c^{(i)}) - T_3(s^{(i)}, s_c^{(i)})|}{|m_3(s^{(i)}) - T_3(s^{(i)}, s_c^{(i)})|} \underset{(2.1),(3.52)}{\leq} \frac{\sigma \|s_c^{(i)}\|^4}{\hat{\sigma}_c^{(i)} \|\tilde{s}_c^{(i)}\|_4^4} \underset{(3.48)}{\leq} \frac{\sigma}{\hat{\sigma}_c^{(i)} n^{-1} \|\check{Q}_i\|^4}.$$

Therefore, assume $\hat{\sigma}_c^{(i)} \geq \frac{\sigma n \|\check{Q}_i\|^4}{(1-\eta)}$, we deduce that $|\rho_i - 1| \leq 1 - \eta$ and thus $\rho_i \geq \eta$. Then, iteration k is successful. □

Theorem 3.3.5. ***(Lower bound on the step size)*** *For any* $0 < \epsilon < 1$,

$$\|\tilde{s}_c^{(i)}\| > \left(\frac{\epsilon}{\sigma \|\check{Q}_i\|^3 + \hat{\sigma}_c^{(i)}} \right)^{\frac{1}{3}}. \tag{3.53}$$

Proof. Note that while Algorithm 3.3.3 does not terminate, we have

$$\begin{aligned} \epsilon &< \|\nabla m_3(s^{(i)} + s_c^{(i)})\| = \|\nabla M(s^{(i)}, s_c^{(i)})\| = \|\nabla \mathcal{M}_c(s^{(i)}, s_c^{(i)}) - \nabla M(s^{(i)}, s_c^{(i)})\| \\ &\leq \sigma \|s_c^{(i)}\|^3 + \hat{\sigma}_c^{(i)} \sum_{j=1}^{n} |\tilde{s}_{c,j}^{(i)}|^3 \leq \big(\sigma \|\check{Q}_i\|^3 + \hat{\sigma}_c^{(i)}\big) \|\tilde{s}_c^{(i)}\|_2^3. \end{aligned} \tag{3.54}$$

The last inequality uses the standard norm inequalities $(\sum_{j=1}^n |\tilde{s}_{c,j}^{(i)}|^3)^{1/3} = \|\tilde{s}_c^{(i)}\|_3 \leq \|\tilde{s}_c^{(i)}\|$ and $\|s_c^{(i)}\| = \|\check{Q}_i \tilde{s}_c^{(i)}\| \leq \|\check{Q}_i\| \|\tilde{s}_c^{(i)}\|$. Rearranging gives (3.53). □

Theorem 3.3.6. ***(Function value decrease in fully separable iterations)*** *The mechanism of Algorithm 3.3.3 guarantees that, in a successful iteration and for any* $0 < \epsilon < 1$,

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) > \kappa_1 \eta \epsilon^{\frac{4}{3}}, \tag{3.55}$$

where κ_1 *is* ϵ*-independent, but depends on* n *and* i.

Proof. Let $E_i = m_3(s^{(i)}) - m_3(s^{(i)} + s_c^{(i)})$. Using the standard norm inequality $\|\tilde{s}\|_4 \leq \|\tilde{s}\| \leq n^{\frac{1}{4}}\|\tilde{s}\|_4$, we obtain

$$\eta^{-1}E_i \geq m_3(s^{(i)}) - T_3(s^{(i)}, s_c^{(i)}) \geq \frac{\hat{\sigma}_c^{(i)}}{4}\|\tilde{s}_c^{(i)}\|_4^4 \geq \frac{\hat{\sigma}_c^{(i)}}{4n}\|\tilde{s}_c^{(i)}\|^4 \underset{(3.53)}{\geq} \frac{\hat{\sigma}_c^{(i)}}{4n}\left(\frac{\epsilon}{\sigma\|\check{Q}_i\|^3 + \hat{\sigma}_c^{(i)}}\right)^{\frac{4}{3}}.$$

If $\hat{\sigma}_c^{(i)} \leq \hat{\sigma}_c^{(i)}\|\check{Q}_i\|^3$, using $\hat{\sigma}_c^{(i)} = \sigma + d_i \geq \sigma$, we have

$$\eta^{-1}E_i \geq \frac{\hat{\sigma}_c^{(i)}}{4n}\left(\frac{1}{2\sigma\|\check{Q}_i\|^3}\right)^{\frac{4}{3}}\epsilon^{\frac{4}{3}} \geq \frac{1}{16n}\sigma^{-\frac{1}{3}}\|\check{Q}_i\|^{-4}\epsilon^{\frac{4}{3}}.$$

If $\hat{\sigma}_c^{(i)} \geq \sigma\|\check{Q}_i\|^3$, then

$$\eta^{-1}E_i \geq \frac{\hat{\sigma}_c^{(i)}}{4n}\left(\frac{1}{2\hat{\sigma}_c^{(i)}}\right)^{\frac{4}{3}}\epsilon^{\frac{4}{3}} \geq \frac{1}{16n}(\hat{\sigma}_c^{(i)})^{-\frac{1}{3}}\epsilon^{\frac{4}{3}} \underset{(3.51)}{\geq} \frac{1}{16n}\left[\frac{\sigma n\|\check{Q}_i\|^4}{(1-\eta)}\right]^{-1/3}\epsilon^{\frac{4}{3}} \geq \frac{\|\check{Q}_i\|^{-\frac{4}{3}}}{16n^{\frac{4}{3}}}\left(\frac{\sigma}{(1-\eta)}\right)^{-\frac{1}{3}}\epsilon^{\frac{4}{3}}.$$

Let $\kappa_1 = \min\left\{\frac{1}{16n}\sigma^{-\frac{1}{3}}\|\check{Q}_i\|^{-4}, \frac{\|\check{Q}_i\|^{-\frac{4}{3}}}{16n^{\frac{4}{3}}}\left(\frac{\sigma}{(1-\eta)}\right)^{-\frac{1}{3}}\right\}$ gives (3.55). □

It is important to note that the analysis in Theorem 3.3.6 only addresses the function value reduction for the jointly diagonalizable iteration (i.e., in this iteration H_i and T_i are jointly diagonalizable). If we aim to establish the complexity for all iterates $\{s^{(i)}\}_{k\geq 0}$, further assumptions on the objective function are required. Given that the CQR algorithmic framework (Algorithms 3.3.1 and 3.3.3) applies to a general objective function under suitable conditions (i.e., the function is three times continuously differentiable, bounded below, and its third derivative is globally Lipschitz continuous), we denote the general objective function by $\hat{m}(s)$. Assuming that $\nabla^2\hat{m}(s)$ and $\nabla^3\hat{m}(s)$ are jointly diagonalizable for all $s \in \mathbb{R}^n$ with a common basis $Q(s)$, and that the largest absolute eigenvalue of $Q(s)$ is bounded in each iteration such that $\|Q(s)\| \leq \lambda_Q$ for all $s = s^{(i)}$, then the test for separability in Step 1b of Algorithm 3.3.3 is satisfied for all iterations. By replacing $\|\check{Q}_i\|$ with λ_Q, the parameter κ_1 in (3.55) becomes an iteration-independent and ϵ-independent constant. Employing techniques similar to those in Lemmas 3.3.3–3.3.4 and Theorem 3.3.4, we can track the number of successful and unsuccessful adaptive-regularization iterations. Consequently, we deduce that Algorithm 3.3.3 requires at most $\mathcal{O}(\epsilon^{-\frac{4}{3}})$ function evaluations (i.e., $\hat{m}(s^{(i)})$) and derivative evaluations (i.e., $\nabla^j\hat{m}(s^{(i)})$ for $j \geq 1$) to compute an iterate s_ϵ such that $\|g_i\| \leq \epsilon$ for any $0 < \epsilon < 1$, for a fully separable objective function.

Remark 3.3.6 (Relationship with difference-of-convex (DC) programming)**.** *The QQR and CQR models are closely related to DC programming. The models can be written as the difference of two convex functions by adding and subtracting the same quadratic regularization term. This DC representability does not imply that the QQR and CQR solvers are standard DC algorithms. Generic DC methods typically linearize one convex component and seek a critical point, whereas QQR and CQR exploit the specific quadratic–cubic–quartic structure of the models, together with dedicated global optimality conditions and structured minimization procedures. A more detailed investigation of the relationship between their secular equations and the fixed-point conditions of suitable DC algorithms is left for future work.*

3.4 Numerical Implementation and Preliminary Results

In this section, we present an implementation of the CQR method in Algorithm 3.4.1 and provide preliminary numerical results when we apply this algorithm to minimize $m_3(s)$. In Section 3.4.1, we compare the CQR method with state-of-the-art algorithms, while in Section 3.4.2, we present special numerical cases where the CQR method has a performance advantage over ARC and QQR.

While Algorithm 3.3.1 was incomplete in its specification of β_i, Algorithm 3.4.1 makes this precise by offering three options for updating β_i to choose from. The first choice of β_i in Algorithm 3.4.1 stems from our theoretical analysis in Theorem 3.3.2 and Corollary 3.3.1. By setting $\beta_i = L_H(s^{(i)}) =$

Algorithm 3.4.1: An Implementable CQR Variant for Minimizing $m_3(s)$

Initialize and proceed through Steps 1 to 4 of the CQR algorithmic framework (Algorithm 3.3.1). Specifically, in each iteration (i), compute $s_c^{(i)} := \operatorname{argmin}_{s\in\mathbb{R}^n} M_c(s^{(i)}, s)$, where β_i is initialised and updated using one of the following updating rules:

1. $\beta_{i+1} := L_H(s^{(i)}) = \Lambda_0 + 6\sigma\|s^{(i)}\|$ for $i \geq 0$; $\quad \beta_0 := \Lambda_0 = \sqrt{\sum_\iota \max_{1\leq j,k\leq n} T[\iota, j, k]^2}$.
2. $\beta_{i+1} := \frac{T_i[s_c^{(i)}]^3}{\|s_c^{(i)}\|^3}$ for $i \geq 0$; $\quad \beta_0 := -\max_{1\leq j\leq n} |T[j, j, j]|$.
3. $\beta_{i+1} := \frac{1}{n}\sum_{j=i}^n T_i[j, j, j]$ for $i \geq 0$; $\quad \beta_0 := -\max_{1\leq j\leq n} |T[j, j, j]|$.

In choices 2 or 3 above: initialise an upper bound $B > 0$ at the start of the algorithm; at iteration (i), if $|\beta_{i+1}| > B$, then set[b] $\beta_{i+1} := \operatorname{sign}(\beta_{i+1})B$. In practice, we choose B to be sufficiently large, typically $B = 10^9$, so that this safeguard is only activated in extreme cases.

[b] See Remark 3.4.1.

$\Lambda_0 + 6\sigma\|s^{(i)}\|$ where Λ_0 is calculated as in [43, Thm 2.6], we ensure that $M(s^{(i)}, s_c^{(i)}) \leq M_c(s^{(i)}, s_c^{(i)})$ for all i. This guarantees that $\rho_i > 1$ in every iteration of Algorithm 3.4.1, namely every step is a successful step. A numerical example of this choice of β_i is given in Figure A.2 of Section A.1.3 where we plot the value of β_i, the size of d_i, and the performance ratio for both successful and unsuccessful iterations. Our experimental findings confirm our theoretical results that each iteration satisfies $\rho_i \geq 1$ (right plot of Figure A.2), so there are no unsuccessful iterations and d_i is decreased accordingly. The second and third choices of β_i in Algorithm 3.4.1 are of more practical relevance and hence will be tested henceforth. The second choice of β_i in Algorithm 3.4.1 is derived from the observation in (3.35). For each iteration, Theorem 3.3.2 holds if the condition $\beta_{i+1} \geq T_i[s_c^{(i+1)}]^3\|s_c^{(i+1)}\|^{-3}$ is satisfied. However, prior to minimizing $M_c(s^{(i)}, s)$, $s_c^{(i)}$ remains unknown. Therefore, in the algorithmic implementation, we update the $\beta_{i+1} := T_i[s_c^{(i)}]^3\|s_c^{(i)}\|^{-3}$ which is available. For the third choice of β_i in Algorithm 3.4.1, we opt for β_i to be the scaled trace of T_i: $\beta_{i+1} := \frac{1}{n}\sum_{j=i}^n T_i[j, j, j]$. Preliminary examples for these two choices of β_i are provided in Section A.1.3. In addition to choices 1–3, other techniques for selecting β_i are possible, such as letting β_i be the smallest or largest eigenvalue or entry of the tensor T. It is important to note that care is needed in making such choices to maintain the tensor-free nature of the iterations. For instance, Algorithm 3.4.1 is tensor free for choices 2 and 3 of β_i.

Remark 3.4.1. *In Assumption 3.3.1, we assume that β_i is bounded for all iterations. In Algorithm 3.4.1, the parameter B prevents $|\beta_i|$ from becoming excessively large, thereby averting potential numerical issues. Careful consideration is needed when selecting B to ensure that the algorithm maintains its tensor-free nature. Currently, we set B as the maximum value among the entries in T_0, which requires the tensor at x_0. Alternatively, other tensor-free options for B exist, such as setting B as a uniform bound.*

When contrasting the implementable CQR variant (Algorithm 3.4.1) with the CQR variant featuring optimal complexity (Algorithm 3.3.2), we note the former allows more freedom in the choice $\sigma_c^{(i)}$. Yet, according to the analysis presented in Remark 3.3.4, Algorithm 3.4.1 ensures a reduction in the value of m_3 during each successful iteration. A numerical example comparing Algorithm 3.3.2 and Algorithm 3.4.1 is given in Figure A.1 of Section A.1.3.

3.4.1 Numerical Results: Comparison Across Methods

In this section, we carry out numerical experiments for CQR algorithm (Algorithm 3.4.1) with practical β choices (Choices 2 and 3). To assess the performance of CQR solvers, we generate AR3 subproblems randomly with dimensions, n, ranging from 5 to 100. These test problems comprise AR3 subproblems with varying gradients, Hessians, tensors, and regularization terms. Specifically, we consider singular, ill-conditioned, indefinite, and diagonal Hessians, as well as large and small tensor terms, diagonal tensor terms, badly scaled tensor terms, and dense tensor terms. We also provide examples arising from the minimization of multi-dimensional objective functions that illustrate further aspects of the

CQR method's performance. Additional numerical results on the Moré–Garbow–Hillstrom (MGH) test set are provided in Section 4.5 after we introduced the DTM algorithm.

3.4.1.1 Numerical Set-up

The test problems are constructed the same way as in Section 2.5.3 and in (2.59). We set the stopping criterion to be $\|g_i\| < \epsilon$, where $\epsilon = 10^{-5}$ unless otherwise specified. The parameters in Algorithm 3.4.1 are set to $\eta_1 = 0.9$, $\eta = 0.1$, $\gamma_2 = 0.5$, $\gamma = 2.0$ and $B = \max_{1\leq\iota,j,k\leq n} \left|T[\iota, j, k]\right|$. The reported evaluation counts include the successful and total iterations, but do not include the cost of calculating B. The successful iteration count corresponds to the number of derivative evaluations, while the total iterations match the number of objective evaluations. It is worth noting that, apart from possibly the setting/calculation of the parameter B at initialization, Algorithm 3.4.1 is tensor-free, requiring only tensor-vector and tensor-matrix products.

3.4.1.2 Comparison Across Methods

We now compare the numerical performance of several minimization algorithms for the AR3 subproblem. These algorithms include the ARC method [46] (as baseline), Nesterov's method (specifically for convex m_3), `QQR-v1` method, and `QQR-v2` method. Nesterov's method, introduced in [142], is tailored for convex subproblems only. The QQR method in Chapter 2 can be considered as a non-convex generalization of Nesterov's method; it approximates the third-order tensor term through a linear combination of quadratic and quartic terms. In our numerical testing, we assess both variants of QQR (`QQR-v1` and `QQR-v2`). `QQR-v1` follows a framework similar to Nesterov's method, employing a single adaptive parameter to govern the quadratic and quartic terms. In contrast, `QQR-v2` employs two adaptive parameters to manage the convexity scenario and regularization magnitude, respectively.

The numerical results (Tables A.1–A.5 in Section A.1.4) consistently show that the CQR method (with β choices 2 or 3 in Algorithm 3.4.1) performs competitively with the ARC method and Nesterov's method. In particular, CQR methods typically require fewer evaluations of functions/derivatives or iterations than the ARC method, Nesterov's method, and `QQR-v1`. Moreover, CQR methods exhibit robustness when handling ill-conditioned Hessian terms and perform effectively for minimizing m_3 with singular or diagonal Hessians (Table A.2). In the case of convex and locally strictly convex m_3 functions (Table A.1), the CQR algorithm (Algorithm 3.4.1) terminates within only a few iterations and function evaluations. Notably, in these scenarios, the CQR method requires fewer iterations and evaluations to converge than Nesterov's method (which is specifically for convex m_3). The primary reason is that our CQR implementation employs β_i to estimate the tensor direction, whereas Nesterov's method (and `QQR-v1`) solely utilizes a single-parameter linear combination of the local upper and lower bounds. Furthermore, CQR effectively locates a point satisfying $\|g_i\| < \epsilon$ for quartic polynomials characterised by diagonal tensor terms, and large, negative, directional tensors (Tables A.4–A.5). Lastly, CQR is robust across a range of σ values (Table A.3), allowing for dynamic σ adjustments while employing CQR method to solve the AR3 problem.

3.4.2 CQR Performance in Challenging Scenarios

As observed in Table A.1 to Table A.5, for well-scaled subproblems, CQR methods tend to perform comparably with the `QQR-v2` method in terms of iterations and function evaluations. However, we have noticed that in certain practical problems, such as Problem 4 of Moré, Garbow, and Hillstrom's test set [129], known as the Brown badly scaled problem, our investigation revealed a performance advantage of CQR when compared to ARC and QQR. We solved this problem using the AR3 method as described in [53] by minimizing (AR3 Model) with various subproblem solvers and compared their performances. The performance comparison is provided in Table 3.1. We observe that the AR3 method with the CQR subproblem solver requires only approximately 2 iterations (or evaluations) per subproblem to converge, while the AR3 method in (AR3 Model) with the ARC or QQR subproblem solver requires at least three times the iterations (or evaluations). After examining the subproblem linked to this challenging scenario, we noticed that the gradient, Hessian, and tensor exhibited significantly larger magnitudes in a specific entry (or in a few entries), while all other entries remained relatively small,

typically around a magnitude of 1. Furthermore, the search directions were usually predominantly influenced by these tensor directions.

Table 3.1: The objective function to minimize is Problem 4 of Moré, Garbow, and Hillstrom's test set, known as the Brown badly scaled problem. The first-order optimality tolerances for minimizing the objective function and the subproblem are 10^{-4} and 10^{-5} respectively, and the initialization is $x_0 = \mathbf{0}$ with $\sigma = 1$. The AR3 model is outlined in (AR3 Model), more details can be found in [53].

	No. of iterations per subproblem	
Method	Total Iter.	Successful Iter.
AR3 with ARC	16.4	6.3
AR3 with QQR-v2	10.3	10.2
AR3 with CQR-Choice 2	1.9	1.9
AR3 with CQR-Choice 3	1.9	1.9

3.4.2.1 CQR Performance on Badly Scaled Problems

Given these challenging conditions, we conducted a series of tests to assess the performance of the CQR solver in handling such badly scaled problems. We first design a test scenario where all other entries in g, H, and T were set to 0, with only the first diagonal entry varying in magnitude from 0.1 to 10^5. In these examples, as illustrated in the right two plots of Figure A.3 in Section A.1.5, CQR converges in just 1 or 2 iterations, essentially irrespective of problem dimension, whereas ARC and `QQR-v2` required 6 to 15 iterations. We attribute this behaviour to the problem's strong dependence on the direction characterised by the largest tensor entry. The successful identification of a point that satisfies $\|g_i\| < \epsilon$ was closely linked to how effectively the algorithms progressed along the dominant tensor direction, which can be naturally captured by the CQR method due to its model construction.

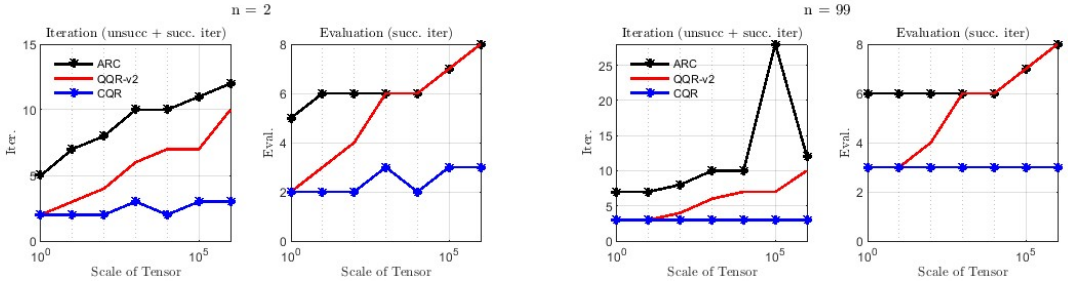
We also added random perturbations to the zero entries of the g, H, and T. We chose the gradient, Hessian, and tensors with entries of relatively small magnitudes, following a normal distribution with zero mean and variance 0.1, while the first entry of g, H, and T was systematically varied from small to large magnitudes in the same direction. We tested these scenarios on subproblems ranging in size from $n = 2$ to $n = 100$. As illustrated in Figure 3.4, when the tensor term is well-scaled, QQR-v2 and the CQR method perform similarly. However, as the magnitude of the first entry increases, signifying a shift in the descent direction towards the dominant direction in T, the CQR method starts to outperform ARC and `QQR-v2` in terms of both iteration and evaluation counts. In particular, for a tensor scaling of 10^6, the CQR algorithm (Algorithm 3.4.1) terminates in just 4 iterations, while ARC and `QQR-v2` require 10 to 25 iterations. This improvement could be primarily attributed to the β parameter, which effectively captures the dominant tensor direction that exerts a substantial influence on the subproblem.

In our exploration of additional examples, we also considered scenarios where the first diagonal entry was large, and we introduced variations in some off-diagonal entries of the tensor T (Figure A.5 Section A.1.5). As expected, similar trends are observed, as the descent direction of m_3 remains predominantly influenced by the diagonal tensor direction T. When we expanded our experiments to include more entries with large magnitudes, the performance of CQR began to match that of `QQR-v2` and ARC. This behaviour was attributed to the introduction of multiple large diagonal entries in T, resulting in a higher rank tensor. The CQR method, with its single β parameter, faces limitations in capturing all the tensor directions introduced.

3.4.2.2 Problems with Gradient and Tensor having Similar Orientation

Another situation in which CQR performs well is when the gradient direction aligns with the dominant tensor direction, known as the rank-one tensor direction. After a change of basis, the gradient and tensor exhibit significantly larger magnitudes in a specific entry, while all other entries remain relatively small. This problem can be transformed to those discussed in Section 3.4.2.1 as follows. Assume that

Figure 3.4: Performance profile of a badly scaled problem: the first entry of g, H, and T scales from -0.1 to -10^5, while the rest of the entries follow a normal distribution $N(0, 0.1)$, and $\sigma = 50$. The left two plots have dimension $n = 2$, and the right two plots have dimension $n = 100$. The tensor scale is defined as the maximum absolute value of the tensor entries divided by the mean absolute value of tensor entries. β_i is chosen as per Choice 2 in Algorithm 3.4.1, with similar performance observed for β_i as per Choice 3 (Figure A.4 in Section A.1.5).



m_3 has a rank-one tensor term $T^{\mathbb{1}}$, which aligns with the gradient term, such that $a_1 = \frac{g}{\|g\|}$ and $T^{\mathbb{1}} = \bar{\mu}_1 \bar{a}_1 \otimes \bar{a}_1 \otimes \bar{a}_1 \in \mathbb{R}^{n^3}$,

$$m_3(s) := f_0 + g^T s + \frac{1}{2} s^T H s + \frac{1}{6} T^{\mathbb{1}}[s]^3 + \frac{\sigma}{4}\|s\|^4. \tag{3.56}$$

Then, we use a change of basis matrix

$$C_1 := \left[-\bar{a}_1 \middle| a_1{}^{\perp}\right]^T \in \mathbb{R}^{n^2} \tag{3.57}$$

and change of variables $\tilde{s} = C_1 s$. Note that $a_1{}^{\perp} \in \mathbb{R}^{n\times(n-1)}$ is the orthogonal complement to a_1. The matrix C_1 is orthogonal with $C_1^T C_1 = C_1 C_1^T = I_n$, $\text{rank}(C_1) = n$ and $\|C_1 s\| = \|s\|$. Let $\tilde{g} = C_1 g$, $\tilde{H} = C_1 H C_1^T$ and $\tilde{T}^{\mathbb{1}} = T^{\mathbb{1}}[C_1^T]^3$, we can transform m_3 as

$$\begin{aligned} m_3(s) &= f_0 + g^T C_1^T (C_1 s) + \frac{1}{2}(C_1 s)^T C_1 H C_1^T (C_1 s) + \frac{1}{6} T^{\mathbb{1}}[C_1^T C_1 s]^3 + \frac{\sigma}{4}\|C_1 s\|^4 \\ &= f_0 + \tilde{g}^T \tilde{s} + \frac{1}{2}\tilde{s}^T \tilde{H} \tilde{s} + \frac{1}{6}\tilde{T}^{\mathbb{1}}[\tilde{s}]^3 + \frac{\sigma}{4}\|\tilde{s}\|^4. \end{aligned} \tag{3.58}$$

Using the definition of C_1 in (3.57) and $a_1 = \frac{g}{\|g\|}$, we deduce that

$$\begin{aligned} \tilde{g} &= C_1 g = [-\|g\|, 0, \ldots, 0]^T \in \mathbb{R}^n, \\ \tilde{T}^{\mathbb{1}} &= \bar{\mu}_1 \big(C_1 a_1 \otimes C_1 a_1 \otimes C_1 a_1\big) = -\bar{\mu}_1 e_1 \otimes e_1 \otimes e_1 = -\bar{\mu}_1 E_{1,1,1} \in \mathbb{R}^{n^3}. \end{aligned}$$

Substituting $\tilde{g}$ and $\tilde{T}^{\mathbb{1}}$ into (3.58) gives

$$m_3(s) = \tilde{m}_3(\tilde{s}) := f_0 - \|g\| e_1^T \tilde{s} + \frac{1}{2}\tilde{s}^T \tilde{H} \tilde{s} - \frac{\bar{\mu}_1}{6} E_{1,1,1}[\tilde{s}]^3 + \frac{\sigma}{4}\|\tilde{s}\|^4 \tag{3.59}$$

where $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^n$ and $E_{1,1,1} = e_1 \otimes e_1 \otimes e_1 \in \mathbb{R}^{n^3}$ has the $(1,1,1)$-entry as 1 and all the other entries are 0; (3.59) is in our desired form.

We provide numerical experiments to demonstrate the efficiency of CQR on these scenarios, when the gradient and the tensor are similarly aligned. Two types of tests are considered.

Problems with rank-1 tensors and gradients with similar orientation: The test problems are of the form (3.56), with $f_0 = 0$, and the coefficients g, H, and $T^{\mathbb{1}}$ generated as

$$a_1 = \texttt{normalized(randn(n, 1))},\ g = \bar{\mu}_1 \bar{a}_1,\ H = \texttt{symm(randn(n, n))},\ T^{\mathbb{1}} = \bar{\mu}_1 \bar{a}_1 \otimes \bar{a}_1 \otimes \bar{a}_1 \tag{3.60}$$

where $\bar{\mu}_1 > 0$ and $a_1 \in \mathbb{R}^n$ is a randomly generated unit vector. Under this construction, the gradient g is a randomly generated vector with $\|g\| = \bar{\mu}_1$, while the tensor $T^{\mathbb{1}}$ is a rank-1 tensor with the dominant direction a_1 and magnitude $\bar{\mu}_1$. The change of variable matrix C_1 is generated as defined in

(3.57). The numerical performance of this set of subproblems with $\bar{\mu}_1$ varies from 0.1 to 10^5 is given in the left plot of Figure 3.5.

Problems with low-rank tensors with a dominating direction: We consider a subproblem m_3 where the gradient g and Hessian H are generated the same as in (3.60). However, instead of the tensor term being rank-1, it is $T^{\mathbb{P}}$ of rank P (where $P \geq 2$), with the dominant tensor direction aligned with that of the gradient. We construct $T^{\mathbb{P}}$ as follows,

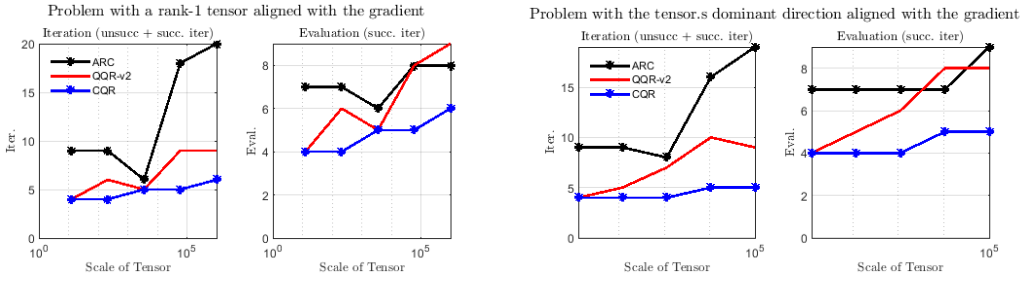
$$T^{\mathbb{P}} = \underbrace{\bar{\mu}_1 \bar{a}_1 \otimes \bar{a}_1 \otimes \bar{a}_1}_{\text{dominant term}} + \sum_{j=2}^{P} \underbrace{\bar{\mu}_j \bar{a}_j \otimes \bar{a}_j \otimes \bar{a}_j}_{\text{smaller terms}} \tag{3.61}$$

where $a_1 \in \mathbb{R}^n$ is in the same direction as the gradient and $\bar{a}_j \in \mathbb{R}^n$ are randomly generated unit vectors for all $j = 2, \ldots, P$. The parameter $\bar{\mu}_1$ is significantly larger than $\bar{\mu}_2, \ldots, \bar{\mu}_P$. Note that the main requirement needed here is that $T^{\mathbb{P}}$, irrespective of its rank P, has only one dominant direction that is significantly larger than other directions, and this direction aligns with the gradient. We vary the magnitude of the gradient and the dominant direction of the tensor from $\bar{\mu}_1 = 0.1$ to $\bar{\mu}_1 = 10^5$ in the right plot of Figure 3.5.

In the numerical examples of Figure 3.5, all three methods converge to the same local minimum of m_3. Figure 3.5 shows that, as $\bar{\mu}_1$ increases (and the tensor term is more aligned with the gradient), the CQR algorithm requires fewer iterations and evaluations than ARC or QQR to find an approximate stationary point for this set of subproblems. These performance advantages align with those observed for the CQR method in Section 3.4.2.1, essentially due to gradient and tensor scaling and conditioning (subject possibly to a change of basis). This is because, after a change of basis, both test sets can be transformed into the badly scaled test set, in which the gradient and tensor display significantly larger magnitudes in specific entries, while all other entries remain relatively small. Generally, under an appropriate change of basis, the CQR method could exhibit a performance advantage over competing methods such as ARC and QQR when the search directions are predominantly influenced by the gradient and tensor directions. Although the CQR method requires fewer iterations and evaluations, the cost of each CQR iteration may be higher (due to the more involved subproblem) and remains to be assessed precisely.

It is worth noting that CQR requires slightly more iterations/evaluations to converge in the second plot in Figure 3.5 compared to Figure 3.4. We believe the main reason for this is the addition of directional noise to the tensor term, as illustrated in (3.61). The CQR method, with its single β parameter, faces limitations in capturing all the (noise) tensor directions introduced. Nonetheless, since there is still a dominant tensor term that aligns with the gradient, we observe that CQR outperforms its competitor methods in terms of iterations/evaluations as the magnitude μ of this dominant term increases. On the other hand, when μ is small, CQR performs similarly with its competitor methods.

Figure 3.5: For both plots, with $\sigma = 50$ and $n = 20$, $\bar{\mu}_1$ scales from 0.1 to 10^5. β_i is chosen as per Choice 2 in Algorithm 3.4.1. **Left**: Performance profile for a problem featuring a rank-1 tensor aligned with the gradient. **Right**: Performance profile for a problem where the tensor's dominant direction is aligned with the gradient, with $\bar{\mu}_2 = \cdots = \bar{\mu}_P = 0.1$ and $P = n$.



Chapter 4

Global Optimality and Local Methods for the Regularized Third-order Subproblem III

4.1 Introduction

In the previous chapters, we have introduced several efficient algorithms and proved their convergence, such as the Quadratic Quartic Regularization (QQR) and Cubic Quartic Regularization (CQR) to minimize the AR3 subproblem. Yet, for $p \geq 3$, characterizing global optimality and efficiently minimizing high-order Taylor models with regularization terms remains an open question. This chapter establishes global optimality conditions for general, nonconvex cubic polynomials with quartic regularization. These criteria generalise existing results, recovering the optimality results for regularized quadratic polynomials, and can be further simplified in the low-rank and diagonal tensor cases. Under suitable assumptions on the Taylor polynomial, we derive a lower bound for the regularization parameter such that the necessary and sufficient criteria coincide, establishing a connection between this bound and the subproblem's convexification and SoS convexification techniques. Leveraging the optimality characterization, we develop a Diagonal Tensor Method (DTM) for minimizing quartically-regularized cubic Taylor polynomials by iteratively minimizing a sequence of local models that incorporate both diagonal cubic terms and quartic regularization (DTM model). We prove that the DTM algorithm is convergent, with a global evaluation complexity of $\mathcal{O}(\epsilon^{-3/2})$. Furthermore, when a special structure is present (such as low rank or diagonal), DTM can exactly solve the given problem (in one iteration). In our numerical experiments, we propose practical DTM variants that exploit local problem information for model construction, which we then show to be competitive with cubic regularization and other subproblem solvers, with superior performance on problems with special structure.

We have introduced several convergent and efficient algorithms, such as the *Quadratic Quartic Regularization (QQR)* and *Cubic Quartic Regularization (CQR)*. However, the key question of global optimality characterization for the subproblem remains unresolved. This chapter addresses key research gaps by characterizing the global optimality of the AR3 subproblem. Additionally, we link the necessary and sufficient conditions to the SoS convex polynomials, demonstrating conditions under which the necessary and sufficient criteria align. We leverage these optimality conditions to design an efficient algorithm tailored specifically for AR3 subproblem minimization. The chapter makes the following key contributions.

1. To the best of our knowledge, we are the first to establish global optimality conditions for a generally nonconvex cubic polynomial with quartic regularization. These characterizations can be seen as a generalization of optimality conditions for various forms of quadratic polynomials with simpler tensor terms or alternative regularization norms. For instance, our results recover the optimality conditions established by Cartis, Gould, and Toint[44, Thm 8.2.8] for quadratic polynomials with quartic regularization. Unlike quadratic polynomials with regularization, in (AR3 Model), the necessary and sufficient global optimality conditions do not generally coincide.

Interestingly, we find that the difference between the necessary and sufficient global optimality conditions is closely related to the convexification and SoS convexification of the polynomial. Specifically, the size of the regularization parameter required to close the gap between the necessary and sufficient conditions differs by a constant from that required for SoS convexification. This provides insight into how the space of SoS-convex quartic-regularized polynomials relates to the space of convex quartic-regularized polynomials. Notably, SoS convexity can be verified in solving an SDP of polynomial size, whereas determining the convexity of quartic-regularized polynomials is strongly NP-hard [1; 7].

2. We propose a convergent algorithm, the Diagonal Tensor Method (DTM), for minimizing $m_3(s)$. The complexity for the algorithm is at least as good as ARC/AR2 at $\mathcal{O}(\epsilon^{\frac{3}{2}})$, with improvements in specific cases. For example, if the third-order derivative of the objective function has rank P ($P \leq n$), then through an appropriate change of basis, DTM can exactly recover the subproblem, leading to convergence in a single iteration. Compared to previous algorithms (such as QQR and CQR in Chapter 2 and Chapter 3), DTM offers greater flexibility in approximating the tensor term and incorporating tensor information. Numerical experiments further indicate that DTM performs competitively with state-of-the-art algorithms in minimizing (AR3 Model).

The chapter is organized as follows. In Section 4.2, we establish the global optimality conditions for the AR3 model and identify circumstances under which the global necessary and sufficient conditions coincide. In Section 4.4, we present an iterative algorithm, the Diagonal Tensor Method (DTM), which leverages these optimality conditions to efficiently minimize the AR3 model. We prove the convergence of the algorithm and analyze its complexity, showing improved performance for cases with low-rank tensor structures. In Section 4.5, we introduce a practical variant of the algorithm and provide preliminary numerical results.

4.2 Global Optimality of AR3 Model

In this section, we examine the global optimality conditions of (AR3 Model). The necessary condition is presented in Section 4.2.1, while the sufficient condition is discussed in Section 4.2.2. Section 4.2.3 explores the differences between the necessary and sufficient conditions for global optimality and under what circumstances this gap could be closed. In this section, since we focus on AR3 sub-problem for a single iteration with a fixed x_k, unless otherwise stated, for notational simplicity, we write $m_3(x_k, s)$ as $m_3(s) = f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}T[s]^3 + \frac{\sigma}{4}\|s\|_W^4$, where $f_0 = f(x_k) = m_3(x_k, 0) \in \mathbb{R}$, $g = \nabla_x f(x_k) \in \mathbb{R}^n$, $H = \nabla_x^2 f(x_k) \in \mathbb{R}^{n\times n}$ and $T = \nabla_x^3 f(x_k) \in \mathbb{R}^{n^3}$. Let

$$\mathcal{B}_d(s) := H + \frac{1}{2}T[s] + \sigma\|s\|_W^2 W, \qquad \mathcal{G}_d(s) := H + T[s] + \sigma\|s\|_W^2 W. \tag{4.1}$$

If s_* is a second order local minimum of m_3, then the second-order local optimality conditions of (AR3 Model) is

$$\nabla m_3(s_*) = 0 \quad \Rightarrow \quad \mathcal{B}_d(s_*)s_* = -g, \tag{4.2}$$
$$\nabla^2 m_3(s_*) \succeq 0 \quad \Rightarrow \quad \mathcal{G}_d(s_*) + 2\sigma(Ws_*)(Ws_*)^T \succeq 0. \tag{4.3}$$

where ∇ denote the derivative with respect to s throughout this chapter. Before proving global optimality, we first present the following lemmas and corollaries for tensor norms.

Corollary 4.2.1. *Assume W is a symmetric, positive-definite matrix. Let Λ_0 be defined as in Definition 2.3.2. $\Lambda_W := \Lambda_0[\lambda_{\min}(W)]^{-3/2}$, then*

$$\max_{v,u\in\mathbb{R}^n} \frac{|T[u][v]^2|}{\|u\|_W\|v\|_W^2} \leq \Lambda_0[\lambda_{\min}(W)]^{-3/2} =: \Lambda_W. \tag{4.4}$$

Proof. Using (2.19), $\lambda_{\min}(W)\|v\|^2 \leq v^T W v = \|v\|_W^2$ for all $v \in \mathbb{R}^n$, we deduce that

$$\max_{u,v\in\mathbb{R}^n} \frac{T[u][v]^2}{\|u\|_W\|v\|_W^2} = \max_{u,v\in\mathbb{R}^n} \frac{T[u][v]^2}{\|u\|\|v\|^2}\frac{\|u\|\|v\|}{\|u\|_W\|v\|_W^2} \leq \Lambda_0[\lambda_{\min}(W)]^{-3/2}. \tag{4.5}$$

□

Lemma 4.2.1. *In some special cases of T and W, Λ_W has a simplified expression.*

- *If T is a diagonal tensor with diagonal entries $\{t_1, t_2, \ldots, t_n\}$. Then,*
$$\Lambda_W = \max\{|t_1|, |t_2|, \ldots, |t_n|\}[\lambda_{\min}(W)]^{-3/2}.$$
- *If T is a diagonal tensor with diagonal entries $\{t_1, \ldots, t_n\}$ and $W = \text{diag}\{|t_j|^{2/3}\}_{1\leq j\leq n}$, then $\Lambda_W = 1$.*

Proof. If $T \in \mathbb{R}^{n^3}$ is a tensor with diagonal entries $\{t_1, \ldots, t_n\}$, we have

$$\max_{u,v\in\mathbb{R}^n} \left|T[u][v]^2\right| = \sum_{j=1}^n |t_j||v_j|^2|u_j| \leq t_*\left(\sum_{j=1}^n |v_j|^2\right)\|u\| \leq t_*\|v\|^2\|u\| \tag{4.6}$$

where $t_* = \max\{|t_1|, |t_2|, \ldots, |t_n|\}$ and the first inequality uses $|u_i| \leq \|u\|$. Substituting (4.6) into (4.5) gives the first result in Lemma 4.2.1. For the second result, if $W = \text{diag}\{|t_j|^{2/3}\}_{1\leq j\leq n}$, we have

$$\max_{u,v\in\mathbb{R}^n} \left|T[u][v]^2\right| \leq \sum_{j=1}^n (|t_j|^{2/3}|v_j|^2)(|t_j|^{1/3}|u_j|) \leq \sum_{j=1}^n (|t_j|^{2/3}|v_j|^2)\|u\|_W = \|v\|_W^2\|u\|_W$$

where the first inequality uses $|t_j|^{1/3}|u_j| \leq \left[\sum_{j=1}^n |t_j|^{2/3}|u_j|^2\right]^{1/2} = \|u\|_W$. □

We also give a technical lemma before proving the global optimality for the AR3 subproblem. Note that Lemma 4.2.2 can be seen as a generalization of Lemma 4.7 in [185]. The proof of Lemma 4.2.2 uses a similar framework as Lemma 4.7 in [185] and an alternative proof is provided in Section A.1.6.

Lemma 4.2.2. *Let $\mathcal{B}_d(s)$ and $\mathcal{G}_d(s)$ be defined as in (4.1). For any vector $s, v \in \mathbb{R}^n$, we have*

$$m_3(s+v) - m_3(s) = \left[g + \mathcal{B}_d(s)s\right]^T v + \frac{1}{2}\mathcal{G}_d(s)[v]^2 + \frac{1}{6}T[v]^3 + \frac{\sigma}{4}\left[\|s+v\|_W^2 - \|s\|_W^2\right]^2. \tag{4.7}$$

Proof. For all vectors $v \in \mathbb{R}^n$, we have

$$\begin{aligned}
m_3(s+v) - f_0 &= g^T(s+v) + \frac{1}{2}H[s+v]^2 + \frac{1}{6}T[s+v]^3 + \frac{\sigma}{4}\|s+v\|_W^4, \\
&= \underbrace{g^T(s+v) + \frac{1}{2}\mathcal{B}_d(s)[s+v]^2 - \frac{1}{12}T[s]^3 - \frac{\sigma}{4}\|s\|_W^4 + \frac{1}{4}T[s][v]^2}_{\mathcal{F}_1} + \\
&\underbrace{-\frac{1}{2}\left[\frac{1}{2}T[s] + \sigma W\|s\|_W^2\right][s+v]^2 + \frac{1}{6}T[s+v]^3 + \frac{\sigma}{4}\|s+v\|_W^4 + \frac{1}{12}T[s]^3 + \frac{\sigma}{4}\|s\|_W^4 - \frac{1}{4}T[s][v]^2}_{\mathcal{F}_2}.
\end{aligned}$$

Note that

$$\begin{aligned}
\mathcal{F}_1 &= g^T s + \frac{1}{2}\left[\mathcal{B}_d(s) - \frac{1}{6}T[s]^3 - \frac{\sigma}{2}\|s\|^2\right][s]^2 + \left[g + \mathcal{B}_d(s)s\right]^T v + \frac{1}{2}\mathcal{B}_d(s)[v]^2 + \frac{1}{4}T[s][v]^2, \\
&= (m_3(s) - f_0) + \left[g + \mathcal{B}_d(s)s\right]^T v + \frac{1}{2}\mathcal{G}_d(s)[v]^2.
\end{aligned}$$

Rearranging $\mathcal{F}_2$ gives $\mathcal{F}_2 = \frac{1}{6}T[v]^3 + \frac{\sigma}{4}\left(\|s+v\|_W^2 - \|s\|_W^2\right)^2$. In conclusion, we have $m_3(s+v) = m_3(s) + \mathcal{F}_1 + \mathcal{F}_2$ which gives (4.7). □

4.2.1 Necessary Global Optimality Conditions

Theorem 4.2.1. ***(Necessary Global Conditions)*** *If s_* is a global minimum of $m_3(s)$ over $\mathbb{R}^n$, then s_* satisfies $\mathcal{B}_d(s_*)s_* := -g$ and*

$$H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W + \frac{\Lambda_W}{3}\|s_*\|_W W \succeq 0 \tag{4.8}$$

where $\mathcal{B}_d$ and $\mathcal{G}_d$ are defined in (4.1), and Λ_W is given in Corollary 4.2.1.

Proof. When s_* is a global minimum of $m_3(s)$ and $\mathcal{B}_d(s_*)s_* := -g$, we deduce from (4.7) that

$$0 \leq 2\big[m_3(s_* + v) - m_3(s_*)\big] = \mathcal{G}_d(s_*)[v]^2 + \frac{1}{3}T[v]^3 + \frac{\sigma}{2}\big(\|s_*\|_W^2 - \|v + s_*\|_W^2\big)^2. \tag{4.9}$$

If $v^T W s_* = 0$, then (4.9) becomes

$$0 \leq \left[\mathcal{G}_d(s_*)\left[\frac{v}{\|v\|}\right]^2 + \frac{1}{3}T\left[\frac{v}{\|v\|}\right]^3 + \frac{\sigma}{2}\frac{\|v\|_W^4}{\|v\|^2}\right]\|v\|^2.$$

As $\|v\| \to 0$, we have the first term $\mathcal{G}_d(s_*)\big[\frac{v}{\|v\|}\big]^2$ dominates, therefore $\mathcal{G}_d(s_*)[v]^2 \geq 0$. Since $-\frac{1}{3}T[s_*] + \frac{\Lambda_W}{3}[v]^2 \geq 0$, we deduce that $\big[H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W + \frac{\Lambda_W}{3}\big][v]^2 \geq \mathcal{G}_d(s_*)[v]^2 \geq 0$. If $\tilde{v}^T W s_* \neq 0$, since $\tilde{v}$ and s_* are not orthogonal with respect to the W-norm, the line $s_* + \tilde{k}\tilde{v}$ intersects the ball centred at the origin with radius $\|s_*\|_W$ at two points, s_* and u, with $\|s_*\|_W = \|u\|_W$. Denote $v := u - s$, and note that v is parallel to $\tilde{v}$. Then, (4.9) can be written as

$$\begin{aligned} 0 \leq 2\big[m_3(s_* + v) - m_3(s_*)\big] &= v^T\left[H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W\right]v + \frac{1}{3}T[u][v]^2 \\ &\leq v^T\left[H + \frac{2}{3}T[s_*] + \sigma W\|s_*\|_W^2 + \frac{\Lambda_W}{3}W\|u\|_W\right]v \end{aligned}$$

where the last inequality uses Corollary 4.2.1, namely $T[u][v]^2 \leq \Lambda_W\|u\|_W\|v\|_W^2$. Finally, using $\|s_*\|_W = \|u\|_W$, we obtain (4.8). □

Remark 4.2.1. *(Other Global Necessary Conditions) If s_* is a global minimum of $m_3(s)$ over $\mathbb{R}^n$, then s_* satisfies $\mathcal{B}_d(s_*)s_* = -g$ where $\mathcal{B}_d(s_*)$ and the following conditions also hold.*

1. *If $v^T W s = 0$, then (4.8) can be tightened to $\mathcal{G}_d(s_*)[v]^2 \geq 0$.*
2. *For all $v \in \mathbb{R}^n$, $\mathcal{G}_d(s_*)[v]^2 + \frac{1}{3}T[v]^3 \geq 0$ for $\|s_*\|_W = \|v + s_*\|_W$. This is derived from (4.9).*
3. *For all $v \in \mathbb{R}^n$, $\mathcal{G}_d(s_*) + \frac{2\Lambda_W}{3}\|s_*\|_W W \succeq 0$. This follows from (4.8) since $-\frac{1}{3}T[s_*][v^2] + \frac{\Lambda_W}{3}\|s_*\|_W \geq 0$.*

The proof of Theorem 4.2.1 (and Lemma 4.2.2) relies on reformulations using quadratic forms, drawing inspiration from Nocedal and Wright [150, Thm 4.1]. The same necessary conditions can be derived through the approach presented by Cartis, Gould, and Toint [44, Sec 8.2.1].

4.2.2 Sufficient Global Optimality Conditions

Theorem 4.2.2. ***(Global Sufficient Condition)*** *Assume $s_* \in \mathbb{R}^n$ satisfies $\mathcal{B}_d(s_*)s_* := -g$ and*

$$H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W - \frac{\Lambda_W}{3}\|s_*\|_W W - \frac{\Lambda_W^2}{18\sigma}W \succeq 0 \tag{4.10}$$

where $\mathcal{B}_d$ and $\mathcal{G}_d$ are defined in (4.1), and Λ_W is given in Corollary 4.2.1. Then, s_* is a global minimum of $m_3(s)$ over $\mathbb{R}^n$.

Proof. From (4.7), when $\mathcal{B}_d(s_*)s_* := -g$. We have

$$2\big[m_3(s_*+v)-m_3(s_*)\big] = v^T\Big[\underbrace{H+\frac{2}{3}T[s_*]+\sigma\|s_*\|_W^2 W}_{:=\mathcal{G}_2(s_*)}\Big]v+\frac{1}{3}T[u][v]^2+\frac{\sigma}{2}\big(\|s_*\|_W^2-\|u\|_W^2\big)^2$$

where $u = v + s_*$. Using $\frac{1}{3}T[u][v]^2 \geq -\frac{\Lambda_W}{3}\|u\|_W\|v\|_W^2$, we have

$$\begin{aligned}2\big[m_3(s_*+v)-m_3(s_*)\big] &\geq v^T\mathcal{G}_2(s_*)v-\frac{\Lambda_W}{3}\|u\|_W\|s+u\|_W^2+\frac{\sigma}{2}\big(\|s_*\|_W^2-\|u\|_W^2\big)^2 \qquad (4.11)\\ &= v^T\mathcal{G}_2(s_*)v+\|s_*+u\|_W^2\Big[-\frac{\Lambda_W}{3}\|u\|_W+\frac{\sigma}{2}(\|s_*\|_W-\|u\|_W)^2\underbrace{\frac{(\|s_*\|_W+\|u\|_W)^2}{\|s_*+u\|_W^2}}_{\geq 1}\Big].\end{aligned}$$

Rearranging gives

$$\begin{aligned}&2\big[m_3(s_*+v)-m_3(s_*)\big] \geq v^T\mathcal{G}_2(s_*)v+\|v\|_W^2\Big[-\frac{\Lambda_W}{3}\|u\|_W+\frac{\sigma}{2}(\|s_*\|_W-\|u\|_W)^2\Big] \qquad (4.12)\\ &= v^T\mathcal{G}_2(s_*)v+\|v\|_W^2\Big[\frac{\sigma}{2}\|u\|_W^2-\Big(\sigma\|s_*\|_W+\frac{\Lambda_W}{3}\Big)\|u\|_W+\frac{\sigma}{2}\|s_*\|_W^2\Big]\\ &= v^T\mathcal{G}_2(s_*)v+\|v\|_W^2\Big[\frac{\sigma}{2}\Big(\|u\|_W-\big(\|s_*\|_W+\frac{\Lambda_W}{3\sigma}\big)\Big)^2-\frac{{\Lambda_W}^2}{18\sigma}-\frac{\Lambda_W}{3}\|s_*\|\Big]. \qquad (4.13)\end{aligned}$$

Using $\mathcal{G}_2 - \frac{{\Lambda_W}^2}{18\sigma}W - \frac{\Lambda_W}{3\sigma}\|s_*\|W \succeq 0$, we obtain that $m_3(s_*+v)-m_3(s_*) \geq 0$ for all $u \in \mathbb{R}^n$ and therefore, s_* is the global minimum. □

Remark 4.2.2. *The bound in Theorem 4.2.2 sharp and attained for some m_3. Assume for a certain subproblem, that the descent direction of T is dominated by s_*, such that $T[\frac{s_*}{\|s_*\|}]^3 = -\Lambda_W$. Choose u parallel to s_* with norm $\|u\|_W = \|s_*\|_W + \frac{t^*}{3\sigma}$. We have $\frac{\|s_*\|_W+\|u\|_W}{\|s_*+u\|_W} = 1$ and therefore, (4.11) and (4.12) become equalities. Also, since $\|u\|_W = \|s_*\|_W + \frac{t^*}{3\sigma}$, (4.13) becomes*

$$2\big[m_3(s_*+v)-m_3(s_*)\big] = v^T\Big[\mathcal{G}_2-\frac{{\Lambda_W}^2}{18\sigma}W-\frac{\Lambda_W}{3}\|s_*\|W\Big]v.$$

Global Optimality and Local Optimality: The global necessary conditions (4.8) do not inherently imply the second-order local optimality condition (4.3). This is because the necessary and sufficient global optimality conditions for m_3 are not equivalent. However, the global sufficient conditions (4.10) do lead to the second-order local optimality condition in (4.3). To see this, for all $v \in \mathbb{R}^n$, we have

$$\begin{aligned}&v^T\Big[H+T[s_*]+\sigma\|s_*\|_W^2+2\sigma Ws(Ws)^T\Big]v \underset{(4.10)}{\geq}\\ &\frac{1}{3}\underbrace{[-T[s_*][v]^2+\Lambda_W\|s_*\|_W\|v\|_W^2]}_{\geq 0 \text{ by Corollary } 4.2.1}+\frac{t^*}{18\sigma}\|v\|_W^2+2\sigma(v^TWs_*)^2 \geq 0.\end{aligned}$$

Therefore, if s_* satisfies the premise of Theorem 4.2.2, then s_* also satisfies the second-order local optimality conditions in (4.2)–(4.3).

4.2.3 Difference Between the Global Necessary and Sufficient Conditions

In this section, we explore the conditions under which the global necessary and sufficient conditions coincide for the AR3 subproblem. Clearly, the first-order condition $\mathcal{B}_d(s_*)s_* = -g$ is identical for both the necessary and sufficient conditions. The second-order global necessary condition is

$$H+\frac{2}{3}T[s_*]+\sigma\|s_*\|_W^2W+\frac{\Lambda_W}{3}\|s_*\|_W W \succeq 0.$$

The second-order global sufficient condition is

$$H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W - \frac{\Lambda_W}{3}\|s_*\|_W W - \frac{{\Lambda_W}^2}{18\sigma}W \succeq 0.$$

Generally, when σ is large or when T is small, the gap between the global necessary and sufficient conditions narrows. For instance, if $T = 0$, the global necessary and sufficient conditions coincide, giving $\mathcal{B}_d(s_*)s_* = -g$ and $H+\sigma\|s_*\|_W^2 W \succeq 0$, which matches the optimality condition for the quadratic quartically regularized polynomial as stated in [44, Theorem 8.2.8]. In the remainder of this section, we discuss the optimality conditions and convexification of m_3 where σ is large.

4.2.3.1 Locally Nonconvex m_3: Large σ and Global Optimality

Under suitable assumptions, if the first-order optimality condition of the objective function is not satisfied, namely $\|g\| \geq \epsilon_g$ for some positive tolerance ϵ_g, then for a locally nonconvex m_3 (i.e., H is not positive definite) and for σ as large as $\mathcal{O}(\epsilon_g^2)$, the global necessary condition (4.8) and the sufficient condition (4.10) coincide.

Lemma 4.2.3. *Assume that $s_* \in \mathbb{R}^n$ satisfies $\mathcal{B}_d(s_*)s_* := -g$ and σ satisfies*

$$\sigma \geq 3\max\left\{ -\lambda_{\min}[H]\|s_*\|_W^{-2}\lambda_{\min}(W)^{-1}, \qquad \frac{7}{3}\Lambda_W\|s_*\|_W^{-1}\right\}, \tag{4.14}$$

then (4.8) and (4.10) are equivalent. Moreover, if s_* and σ satisfy $\mathcal{B}_d(s_*)s_* := -g$, (4.8) and (4.14), then s_* is a global minimum.

Proof. Clearly, (4.8) implies (4.10). To prove we converse, we rewrite (4.10) as

$$\begin{aligned}\frac{2}{3}\left[H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W + \frac{\Lambda_W}{3}\|s_*\|_W W\right] + \\ \frac{1}{3}\Bigg(\underbrace{H + \frac{2}{3}T[s_*] + \sigma\|s_*\|_W^2 W - \frac{5}{3}\Lambda_W\|s_*\|_W W - \frac{{\Lambda_W}^2}{6\sigma}W}_{:=\mathcal{J}_2}\Bigg) \succeq 0.\end{aligned} \tag{4.15}$$

Using (4.14), we deduce that

$$\sigma > 2^{-1/2}\Lambda_W\|s_*\|_W^{-1} \quad \Rightarrow \quad \frac{\sigma}{3}\|s_*\|_W^2 > \frac{{\Lambda_W}^2}{6\sigma}, \tag{4.16}$$

$$\sigma \geq 7\Lambda_W\|s_*\|_W^{-1} \quad \Rightarrow \quad \frac{\sigma}{3}\|s_*\|_W^2 \geq \frac{7}{3}\Lambda_W\|s_*\|_W, \tag{4.17}$$

$$\sigma \geq -3\lambda_{\min}[H]\|s_*\|_W^{-2}\lambda_{\min}(W)^{-1} \quad \Rightarrow \quad \frac{\sigma}{3}\|s_*\|_W^2\lambda_{\min}(W) \geq -\lambda_{\min}[H]. \tag{4.18}$$

(4.17) gives $\frac{\sigma}{3}\|s_*\|_W^2 W \geq -\frac{2}{3}T[s_*][v]^2+\frac{5}{3}\Lambda_W\|s_*\|_W$ for all unit vector $v \in \mathbb{R}^n$. (4.18) gives $\frac{\sigma}{3}\|s_*\|_W^2 W \succeq -H$. Consequently, we deduce that $\mathcal{J}_2$ defined in (4.15) is positive semi-definite. Thus, (4.10) implies (4.8).

To prove the last statement, if σ and s_* satisfy (4.14), then (4.8) and (4.10) are equivalent. With the condition in $\mathcal{B}_d(s_*)s_* := -g$, using Theorem 4.2.2, we can deduce that s_* is a global minimum. □

The condition stated in (4.14) indicates that for any arbitrarily small $s_0 > 0$, there exists a sufficiently large σ such that m_3 is convex everywhere except in a small region around the origin where $\|s\|_W < s_0$. However, the bound in Lemma 4.2.3 depends on s_*, which is unknown *a priori*. To obtain a bound that does not rely on s_*, similar to [53, Lemma 3.1], we require a uniform lower bound for $\|s_*\|_W$.

Theorem 4.2.3. *Let s_* be a first-order critical point of m_3 such that $\nabla m_3(x_k, s_*) = 0$. For a fixed tolerance $\epsilon_T \in (0, 1)$, either s_* is also an approximate first-order critical point of the Taylor series T_3, satisfying $\|\nabla T_3(x_k, s_*)\| \leq \epsilon_T$, or the following hold:*

1. *The following lower bound for the stepsize holds:* $\|s_*\|_W \geq (\epsilon_T/\sigma)^{1/3}$.

2. *If* σ *satisfies*

$$\sigma \geq \max\left\{(3\lambda)^3\epsilon_T^{-2}, (7\Lambda_W)^{3/2}\epsilon_T^{-1/2}\right\},$$

then the necessary condition (4.8) *and the sufficient condition* (4.10) *are equivalent.*

Proof. To prove the first result, we use s_* is a first-order critical point, therefore $\nabla m_3(x_k, s_*) = 0$ and

$$\epsilon_T < \left\|T_3(x_k, s_*)\right\| = \left\|T_3(x_k, s_*) - \nabla m_3(x_k, s_*)\right\| + \underbrace{\left\|\nabla m_3(x_k, s_*)\right\|}_{=0} = \sigma\|s_*\|_W^3$$

which gives the first result. To prove the second result, we use that σ satisfies the condition in Result 2). Then, we have

$$\sigma^{1/3} \geq 3\lambda\epsilon_T^{-2/3}, \qquad \sigma^{2/3} \geq 7\Lambda_W\epsilon_T^{-1/3}.$$

From this, we deduce that

$$\sigma \geq 3\max\left\{\lambda\sigma^{2/3}\epsilon_T^{-2/3}, \frac{7}{3}\Lambda_W\sigma^{1/3}\epsilon_T^{-1/3}\right\} \geq 3\max\left\{-\lambda\|s_*\|_W^{-2}, \frac{7}{3}\Lambda_W\|s_*\|_W^{-1}\right\}$$

where the last inequality uses $\|s_*\| \geq (\epsilon_T/\sigma)^{1/3}$. Using Lemma 4.2.3, we obtain that (4.8) and (4.10) are equivalent. □

Combining the result of partially convexifying a locally nonconvex m_3 (Lemma 5.3.3) with Theorem 4.2.3, we deduce that, if $\sigma \geq 4(-3\lambda)^3\epsilon_T^{-2} = \mathcal{O}(\epsilon_T^{-2})$ and $\|\nabla T_3(x_k, s_*)\| \geq \epsilon_T$, then, we have the following three results. Firstly, m_3 is convex except in a small region around the origin $\|s\| < \mathcal{O}(\epsilon_T)$. Secondly, $\|s_*\| \geq \left[\frac{\epsilon_T}{\sigma}\right]^{1/3} = \mathcal{O}(\epsilon_T^{1/3})$, which indicates that the global minimizer lies within the convex region of m_3 (Result 1 of Theorem 4.2.3). Thirdly, the necessary and sufficient conditions coincide, namely, the global minimum of m_3 is characterized by $\mathcal{B}_d(s_*)s_* = -g$ and (4.8).

The model m_3 is typically used in the AR3 algorithm and is based on the third-order Taylor expansion of $f(x_k + s)$ at x_k. A point s_* satisfying $\|\nabla T_3(x_k, s_*)\| \leq \epsilon_T$ is considered an approximate first-order critical point of the third-order Taylor expansion T_3. Since the Taylor series approximates $f(x)$ in the vicinity of x_k, the point $x_k + s_*$ can be regarded as a good approximation to a first-order critical point of $f(x)$ near x_k. However, if s_* lies too far from x_k, the Taylor approximation may no longer be accurate, and s_* may fail to approximate a critical point of $f(x)$. To address this, the AR3 framework checks at each iteration whether a sufficient decrease in the objective function has been achieved. If s_* does not yield sufficient decrease, it is rejected, and the regularization parameter σ_k is increased, typically leading to more conservative steps and smaller step sizes in subsequent iterations. A similar analysis of the size of the regularization weight, ensuring that the sufficient and necessary global optimality conditions coincide, can also be applied to the locally nonconvex subproblems in the AR3 algorithm. More details can be found in Section A.2.2.

4.2.3.2 Locally Convex m_3: Convexification and Link to sum-of-squares (SoS) Method

In this section, we demonstrate that if m_3 is strictly locally convex ($H \succeq \delta I_n > 0$) and σ is large enough, then m_3 remains convex for all $s \in \mathbb{R}^n$, and the global necessary and sufficient conditions coincide. We provide an explicit bound for the size of σ. We also give a link on how convexification links to SoS and SoS-convex. We begin with a brief description of two key notions: SoS polynomials and SoS-convexity. A SoS polynomial can be written as a finite sum of squared polynomials, and SoS-convexity refers to polynomials whose Hessians admit such a decomposition. These concepts provide tractable certificates of nonnegativity and convexity and play a central role in modern polynomial optimisation. The formal definitions of SoS polynomials and SoS-convexity are given in Chapter 5, see Definition 5.2.1 and Definition 5.2.2.

It is well known that SoS-convexity is often proposed as a tractable sufficient condition for the convexity of polynomials through the SoS decomposition [6]. Here we proved that the bound we propose for σ shares the same order of magnitude as the SoS convexification bound proposed in [1] (more details are discussed in Chapter 6).

Lemma 4.2.4. *(Convexification for Locally Convex m_3) If $H \succeq \delta I_n \succ 0$ and σ satisfies*

$$\sigma > \frac{1}{4}\Lambda_W^2 \delta^{-1}, \tag{4.19}$$

then $m_3(s)$ is convex, and any s_* satisfying $\mathcal{B}_d(s_*)s_* = -g$ is the global minimum.

Proof. If $H \succeq \delta I_n \succ 0$, for any unit vector $v \in \mathbb{R}$,

$$\nabla^2 m_3(s)[v]^2 = \left[H + T[s] + \sigma\left(\|s\|_W W + 2(Ws)(Ws)^T\right)\right][v]^2 \geq \delta - \Lambda_W \|s\|_W + \sigma\|s\|_W^2. \tag{4.20}$$

The discriminant of the quadratic equation $\delta - \Lambda_W\|s\|_W + \sigma\|s\|_W^2$ is $\Lambda_W^2 - 4\delta\sigma < 0$. The condition in (4.19) ensures that the discriminant is negative. Therefore, we can ensure $\nabla^2 m_3(s)[v]^2 > 0$ for all $s, v \in \mathbb{R}^n$, thus m_3 is convex. If m_3 is convex, any stationary point satisfying $\mathcal{B}_d(s_*)s_* = -g$ is the global minimum. □

Lemma 4.2.5. *Lemma 4.2.4 can be extended to the case of $p \geq 3$, where (4.19) becomes*

$$\sigma \geq (p-2) \max_{3\leq j\leq p} \Lambda_{j,W} \min_{3\leq j\leq p} \left[\frac{\delta}{(p-2)\Lambda_{j,W}}\right]^{-\frac{(p+1)-j}{j-2}}, \tag{4.21}$$

and $\Lambda_{j,W} := (j-2)! \max_{s,v\in\mathbb{R}^n, \|v\|=\|s\|=1} \frac{\nabla^j f(x_k)[s]^{j-2}[v]^2}{\|s\|_W^{j-2}\|v\|_W^2}$ for $3 \leq j \leq p$. Note that for $j = 3$, $\Lambda_{j,W} = \Lambda_W$, and the bound in (4.21) reduces to the same order as (4.19), which is $\mathcal{O}(\Lambda_W^2\delta^{-1})$.

Proof. We write

$$\begin{aligned}\nabla^2 m_p(s)[v]^2 &= \left[H + \frac{1}{(j-2)!}\sum_{j=3}^{p} \nabla^j f(x_k)[s]^{j-2} + \sigma\left(\|s\|_W^{p-1} W + (p-1)(Ws)(Ws)^T\|s\|_W^{p-3}\right)\right][v]^2 \\ &\geq \delta - \sum_{j=3}^{p} \Lambda_{j,W}\|s\|_W^{j-2} + \sigma\|s\|_W^{p-1}. \end{aligned} \tag{4.22}$$

- If $\|s\|_W \leq \min_{3\leq j\leq p}\left(\frac{\delta}{(p-2)\Lambda_{j,W}}\right)^{\frac{1}{j-2}}$, then we deduce that $\frac{\delta}{p-2} - \Lambda_{j,W}\|s\|_W^{j-2} \geq 0$ for all $3 \leq j \leq p$. Consequently, (4.22) is positive.
- In the case of $\|s\|_W \geq \min_{3\leq j\leq p}\left(\frac{\delta}{(p-2)\Lambda_{j,W}}\right)^{\frac{1}{j-2}}$, using (4.21), we have $-\Lambda_{j,W}\|s\|_W^{j-2} + \frac{\sigma}{p-2}\|s\|^{p+1} \geq 0$ for all $3 \leq j \leq p$. Consequently, (4.22) is positive.

□

Link to SoS Convexification: Recently, [1] proved that a sufficiently regularized locally convex polynomial becomes SoS-convex. Specifically, if m_3 is locally convex (i.e., $H \succeq \delta I_n \succ 0$), then for

$$\sigma \geq 8\Lambda_W^2 \delta^{-1}, \tag{4.23}$$

$m_3(s)$ is SoS-convex for all $s \in \mathbb{R}^n$; see [1, Lemma 3] and Section 6.3 – Section 6.3.1. For such locally convex regularized polynomials, both determining a valid regularization parameter σ and globally minimizing the resulting SoS-convex model reduce to solving semidefinite programs in polynomial time. The bound in (4.23) is of the same order as (4.19), with (4.23) being tighter by a constant factor. This is consistent with the fact that SoS-convexity implies convexity [6]. In this regime,

the global minimizer is unique and the necessary and sufficient optimality conditions coincide. This comparison illustrates the gap between convexity and SoS-convexity, and clarifies how increasing the regularization parameter σ eventually forces the problem to satisfy necessary and sufficient global optimality conditions and to become tractable. A full discussion of this connection is beyond the scope of the current chapter, and we refer the reader to Chapter 5. By examining the size of the regularization parameter, Lemma 4.2.4 together with (4.19) provides insight into how the class of SoS-convex quartically regularized polynomials compares with the class of convex quartically regularized polynomials.

4.3 Optimality Conditions in Special Cases

The optimality conditions for (AR3 Model) in (4.7), Theorem 4.2.1 and Theorem 4.2.2 represent the most general form of the optimality conditions for polynomials in this class. They can be simplified to the optimality conditions of various polynomials with either a simpler tensor term T or with alternative regularization norms.

For instance, quadratic polynomials with regularization are crucial components of the (adaptive) cubic regularization framework. By setting $T = 0$, m_3 simplifies to a quadratic polynomial with quartic regularization,

$$m_{\mathbf{QQR}}(s) = f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{\sigma}{4}\|s\|_W^4. \tag{QQR}$$

Polynomials of this form have been extensively studied in Chapter 2 and [44]. For these polynomials, since $T = 0$, we have $\mathcal{B}_d(s) = \mathcal{G}_d(s)$, and Equation (4.7) reduces to

$$m_{\mathbf{QQR}}(s+v) - m_{\mathbf{QQR}}(s) = \big[g + \mathcal{B}_d(s)s\big]^T v + \frac{1}{2}\mathcal{B}_d(s)[v]^2 + \frac{\sigma}{4}\Big[\|s+v\|^2 - \|s\|^2\Big]^2.$$

As a result, the optimality conditions in Theorem 4.2.1 and Theorem 4.2.2 simplify to $g+\mathcal{B}_d(s_*)s_* = 0$ and $\mathcal{B}_d(s_*) \succeq 0$, which align with Theorem 8.2.8 in [44].

Also, by replacing the term $T[s]^3$ with $\beta\|s\|^3$ where $\beta \in \mathbb{R}$, we have

$$m_{\mathbf{CQR}}(s) = f_0 + g^T s + \frac{1}{2}H[s]^2 + \beta\|s\|^3 + \frac{\sigma}{4}\|s\|_W^4. \tag{CQR}$$

These polynomials are useful for approximating ARp subproblems and are studied in detail in Chapter 3. For these polynomials, since $T[s]^3$ is replaced by $\beta\|s\|^3$, Equation (4.7) becomes

$$\begin{aligned} m_{\mathbf{CQR}}(s+v) - m_{\mathbf{CQR}}(s) = \big[g + \mathcal{B}_d(s)s\big]^T v + \frac{1}{2}\mathcal{B}_d(s)[v]^2 + \frac{\sigma}{4}\Big[\|s+v\|^2 - \|s\|^2\Big]^2 + \\ + \frac{\beta}{12}\Big(\|s+v\| - \|s\|\Big)^2\Big(2\|s+v\|^2 + \|s\|\Big). \end{aligned}$$

The global sufficiency condition in Theorem 4.2.2 becomes $g + \mathcal{B}_d(s_*)s_* = 0$, $\mathcal{B}_d(s_*) \succeq 0$, and $\beta \geq -\frac{3}{2}\sigma\|s_*\|_W$, which coincides with the conditions given in Theorem 3.2.2.

The optimality conditions can also be extended to polynomials with a diagonal third-order tensor term and a separable quartics regularization component (SQR). Let $s = [s_1, \ldots, s_n]^T \in \mathbb{R}^n$, we express these polynomials as

$$m_{\mathbf{SQR}}(s) = f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}\sum_{j=1}^{n} t_j s_j^3 + \frac{1}{4}\sum_{j=1}^{n}\sigma_j s_j^4. \tag{SQR}$$

where $\{t_j\}_{1\leq j\leq n}$ are scalars and $\{\sigma_j\}_{1\leq j\leq n}$ are positive scalars. Let $\mathcal{T} = \operatorname{diag}\{t_j\}_{1\leq j\leq n} \in \mathbb{R}^{n\times n\times n}$,

$$\hat{\mathcal{B}}(s) := H + \frac{1}{2}\mathcal{T}[s] + \Sigma[s]^2, \qquad \text{and} \qquad \hat{\mathcal{G}}(s) := H + \frac{1}{3}\mathcal{T}[s] + \Sigma[s]^2 - \operatorname{diag}\Big\{\frac{t_j^2}{18\sigma_j}\Big\} \tag{4.24}$$

where $\Sigma[s]^2 := \text{diag}\{\sigma_j s_j^2\}_{1\le j\le n} \in \mathbb{R}^{n\times n}$. Then Equation (4.7) becomes

$$m_{\mathbf{SQR}}(s+v) - m_{\mathbf{SQR}}(s) = \left[g + \hat{\mathcal{B}}(s)s\right]^T v + \frac{1}{2}\hat{\mathcal{G}}(s)[v]^2 + \frac{1}{4}\left\|\boldsymbol{\sigma}^{\frac{1}{2}} \cdot v \cdot \left(v + 2s + \frac{t\cdot\boldsymbol{\sigma}^{-1}}{3}\right)\right\|^2. \quad (4.25)$$

where $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_n]^T \in \mathbb{R}^n$ and $t = [t_1, \ldots, t_n]^T \in \mathbb{R}^n$. The global sufficiency condition in Theorem 4.2.2 becomes

$$\hat{\mathcal{B}}(s_*)s_* = -g, \qquad \hat{\mathcal{G}}(s_*) \succeq 0. \quad (4.26)$$

The global necessary condition in Theorem 4.2.1 becomes

$$\hat{\mathcal{B}}(s_*)s_* = -g, \qquad \hat{\mathcal{G}}(s_*) \succeq -2\|H_0\|_1 I_n \quad (4.27)$$

where $\|H_0\|_1 = \sum_{\iota,j=1,\iota\neq j}^n |H_{\iota,j}|$ is the sum of the absolute value of the off-diagonal entries of H. For (SQR) polynomials, if the second order term H is a diagonal matrix, $\|H_0\|_1 = 0$. The global necessary condition (4.27) and sufficient condition (4.26) coincide. Moreover, for univariate case with $n = 1$, $m_{\mathbf{SQR}}(s) = m_3(s)$, and the global optimality condition for univariate $m_3(s)$ (or equivalently $m_{\mathbf{SQR}}(s)$ with $n = 1$) becomes $\nabla m_3(s_*) = 0$ and $H + \frac{1}{3}T[s_*] + \sigma\|s_*\|^2 \ge \frac{T^2}{18\sigma}$. Proofs and more details can be found in Section A.2.1.

4.3.1 Transforming a Low-rank Quartically Regularized Polynomial into a Diagonal Tensor Polynomial

In this subsection, we prove that if the tensor term in (AR3 Model), denoted as $T^{\mathbb{P}}$, has rank P with $0 \le P \le n$, then by applying an appropriate change of basis, the model

$$m_3^{\mathbb{P}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}T^{\mathbb{P}}[s]^3 + \frac{\sigma}{4}\|s\|_W^4 \quad (4.28)$$

can be transformed into a quartically regularized polynomial with a diagonal tensor term (i.e. Diagonal Tensor Model),

$$m_3^{\mathbb{D}}(s) := f_0 + \tilde{g}^T\tilde{s} + \frac{1}{2}\tilde{H}[\tilde{s}]^2 + \frac{1}{6}\mathcal{T}[\tilde{s}]^3 + \frac{\sigma}{4}\|\tilde{s}\|_{\tilde{W}}^4. \quad (4.29)$$

Note that $\tilde{g}$, $\tilde{H}$, $\tilde{s}$ are derived from g, H, s after a change of basis and $\mathcal{T}$ is a diagonal tensor with P non zero entries. Note that $T^{\mathbb{P}}$ could have a dense structure while $\mathcal{T}$ has a sparse structure that only has P nonzero terms on the diagonal. The special diagonal structure of the tensor in (4.29) allows the optimality conditions to be simplified and provides an explicit expression for Λ_W. According to Lemma 4.2.1, for $\mathcal{T} = \text{diag}\{t_j\}_{1\le j\le n}$, we have

$$\Lambda_W = \hat{t} = \max\{|t_1|, |t_2|, \ldots, |t_n|\}[\lambda_{\min}(W)]^{-3/2}. \quad (4.30)$$

The global necessary conditions in (4.8) and the sufficient conditions in (4.10) can be simplified by replacing Λ_W with $\hat{t}$.

To show that (AR3 Model) with a low-rank tensor term (with rank less than n) is equivalent to one with a diagonal tensor term through an appropriate change of basis, we begin by defining the rank P tensor, following the conventions in [168; 170; 169].

Definition 4.3.1. *A tensor $T^{\mathbb{P}}$ is said to have rank P if there exist P linearly independent vectors $\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(P)}$ such that*

$$T^{\mathbb{P}} = \sum_{k=1}^P \boldsymbol{a}^{(k)} \otimes \boldsymbol{a}^{(k)} \otimes \boldsymbol{a}^{(k)}, \qquad \textit{and} \qquad T^{\mathbb{P}}[s]^3 = \sum_{k=1}^P \left(a_1^{(k)}s_1 + \cdots + a_n^{(k)}s_n\right)^3. \quad (4.31)$$

Here, $\otimes$ denotes the outer product, $\boldsymbol{a}^{(k)} = [a_1^{(k)}, \ldots, a_n^{(k)}]^T \in \mathbb{R}^n$, and $s = [s_1, s_2, \ldots, s_n]^T \in \mathbb{R}^n$. The (l, j, κ)-th entry of the tensor is given by $(T^{\mathbb{P}})_{lj\kappa} = \sum_{k=1}^P a_l^{(k)}a_j^{(k)}a_\kappa^{(k)}$.

Under a suitable change of basis, a rank P third-order term can be diagonalized to have P non-zero diagonal entries, while a rank-one third-order term reduces to a single non-zero element.

$$\text{Rank 1 Tensor:} \quad T^{\mathbb{1}}[s]^3 = (a_1^{(1)}s_1 + , \ldots, + a_n^{(1)}s_n)^3 \underset{\text{change of basis}}{\rightsquigarrow} \tilde{s}_1^3$$

$$\text{Rank } P \text{ Tensor:} \quad T^{\mathbb{P}}[s]^3 = \sum_{k=1}^{P}(a_1^{(k)}s_1 + , \ldots, + a_n^{(k)}s_n)^3 \underset{\text{change of basis}}{\rightsquigarrow} \tilde{s}_1^3 + \tilde{s}_2^3 + \ldots + \tilde{s}_P^3.$$

We provide the result in Theorem 4.3.1.

Theorem 4.3.1. *Let $m_3^{\mathbb{P}}(s)$ be defined as in (4.28), and let $T^{\mathbb{P}}$ be defined as in Definition 4.3.1 with $P \leq n$. Define*

$$\hat{C} := \left[\boldsymbol{a}^{(1)}, \quad \ldots, \quad \boldsymbol{a}^{(P)}\right]^T \in \mathbb{R}^{P\times n}, \qquad C := \left[\hat{C}, \quad \hat{C}_\perp\right]^T \in \mathbb{R}^{n\times n}, \tag{4.32}$$

where $\hat{C}_\perp = [\boldsymbol{a}_\perp^{(P+1)}, \ldots, \boldsymbol{a}_\perp^{(n)}]^T$ with $\boldsymbol{a}^{T(i)}\boldsymbol{a}_\perp^{(j)} = 0$ for all $i = 1, \ldots, P$ and $j = P+1, \ldots, n$. Define the transformed variables

$$\tilde{s} = Cs, \quad \tilde{g} = C^{-T}g, \quad \tilde{H} = C^{-T}HC^{-1}, \quad \tilde{W} = C^{-T}WC^{-1}, \quad \mathcal{T} = \text{diag}\{t_j\}_{1\leq j\leq n} \tag{4.33}$$

where $t_1 = \ldots = t_P = 1, t_{P+1} = \ldots = t_n = 0$. Note that $\tilde{W}$ is derived from W after a change of basis. Then,

$$\tilde{m}(\tilde{s}) = f_0 + \tilde{g}^T\tilde{s} + \frac{1}{2}\tilde{s}^T\tilde{H}\tilde{s} + \frac{1}{6}\mathcal{T}[\tilde{s}]^3 + \frac{\sigma}{4}\|\tilde{s}\|_{\tilde{W}}^4 = m_3^{\mathbb{P}}(s).$$

Proof. Note that $Cs = \left[(\boldsymbol{a}^{(1)})^Ts, \ldots, (\boldsymbol{a}^{(P)})^Ts, \ldots\right]^T \in \mathbb{R}^n$, and let $[Cs]_k$ denote the kth entry of Cs. Then

$$\mathcal{T}[Cs]^3 = \sum_{k=1}^{n} t_k^3([Cs]_k)^3 = \sum_{k=1}^{P}(\boldsymbol{a}^{(k)^T}s)^3 \underset{(4.31)}{=} T^{\mathbb{P}}[s]^3,$$

where the first equality arises from the tensor operation on the diagonal tensor $\mathcal{T}$, and the second equality holds because $t_1 = \ldots = t_P = 1$ and $t_{P+1} = \ldots = t_n = 0$.

Since $\hat{C}$ has linearly independent columns. Consequently, $C := \begin{bmatrix}\hat{C} & \hat{C}_\perp\end{bmatrix}^T$ is full rank, and C^{-1} exists. Using $T^{\mathbb{P}}[s]^3 = \mathcal{T}[Cs]^3$, we rewrite (4.28) as follows:

$$\begin{aligned} m_3^{\mathbb{P}}(s) &= f_0 + g^TC^{-1}Cs + \frac{1}{2}(C^{-1}Cs)^THC^{-1}Cs + \frac{1}{6}\mathcal{T}[Cs]^3 + \frac{\sigma}{4}\|C^{-1}Cs\|_W^4 \\ &\underset{\tilde{s}=Cs}{=} f_0 + (C^{-T}g)^T\tilde{s} + \frac{1}{2}\tilde{s}^T(C^{-T}HC^{-1})\tilde{s} + \frac{1}{6}\mathcal{T}[\tilde{s}]^3 + \frac{\sigma}{4}\left[\tilde{s}^T(C^{-T}WC^{-1})\tilde{s}\right]^2. \end{aligned} \tag{4.34}$$

Note that $\tilde{H} = C^{-T}HC^{-1}$ is a symmetric matrix, and $\tilde{W} = C^{-T}WC^{-1} \succ 0$. Therefore, the norm $\|\tilde{s}\|_{\tilde{W}}^2 = \tilde{s}^T(C^{-T}WC^{-1})\tilde{s}$ is well-defined. Substituting $\tilde{g} = C^{-T}g$, $\tilde{H} = C^{-T}HC^{-1}$, and $\tilde{W} = C^{-T}WC^{-1}$ into (4.34) yields the desired result. □

In practice, many objective functions have a rank-one third-order derivative or a low-rank tensor term. A basic example is a function $f : \mathbb{R}^n \to \mathbb{R}$ with $x = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ given by

$$f(x) = (a_1x_1 + \ldots + a_nx_n)^3 + \text{lower-degree separable cross terms.}$$

In this case, the third-order derivative of f is rank-one. The lower-order separable cross terms may include combinations such as x_jx_k, x_j^2, and x_j for any $j, k = 1, \ldots, n$.

Furthermore, tensor data from real-world scenarios (e.g., in image processing [118], computer vision [152], machine learning [103; 180], and bioinformatics [32]) often lie in low-dimensional subspaces and exhibit low-rank structures. The problem of low-rank tensor learning, where only a limited number of samples are available to infer the underlying tensor, can be formulated as

$$\min_{\mathcal{X}\in\hat{D}} f_{\hat{n},\mathcal{Y}}(\mathcal{X}) + \hat{\beta}\cdot\text{rank}(\mathcal{X}),$$

where $f_{\hat{n},\mathcal{Y}}(\cdot)$ is a loss function dependent on the number of samples $\hat{n}$ and the observed tensor $\mathcal{Y}$. Here, $\hat{D}$ represents a given constraint set, $\hat{\beta} > 0$ is a regularization parameter, and the parameter tensor $\mathcal{X} \in \mathbb{R}^{n^3}$ and observed tensor $\mathcal{Y}$ have low-rank structures. In Section 4.5.3, we give an illustrative example—namely, the sphere-packing problem—where low-rank and spare third-order tensors emerge empirically at the optimized configuration. Whether such behaviour persists more broadly in large-scale real-world problems remains an open question for future investigation.

Extensive literature provides methods for finding the best rank-1 approximation for symmetric third-order tensors, including higher-order singular value decomposition (HOSVD), a multilinear extension of the singular value decomposition [64], the higher-order power method (HOPM) [63], and symmetric-HOPM [65; 100]. While these methods typically operate on existing tensors, the tensor term in (AR3 Model) possesses a special structure, as it represents third-order derivative information and can thus be estimated from lower-order data. Recently proposed Generalized quasi-Newton methods [183] approximate the third-order tensor using Hessian evaluations, resulting in a low-rank structure. Additionally, [169; 170] employ p interpolation points (not necessarily consecutive) to derive tensor approximation expressions in the form

$$m^{\mathcal{S}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{2}\sum_{k=1}^{P}\left(\boldsymbol{a}^{(k)^T}s\right)^2\left(\boldsymbol{b}^{(k)^T}s\right) + \frac{1}{6}\sum_{k=1}^{P}\gamma_k\left(\boldsymbol{a}^{(k)^T}s\right)^4, \qquad \text{(Schnabel Model)}$$

where $\gamma_k > 0$ and $\boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k)} \in \mathbb{R}^n$. In the case where the high-order derivatives are low-rank, and we have $\boldsymbol{a}^{(k)} = \boldsymbol{b}^{(k)}$, using a similar change of basis as in Theorem 4.3.1, (Schnabel Model) can be transformed into a separable quartically regularized polynomial (SQR),

$$m^{\mathcal{S}}(\tilde{s}) = f_0 + \tilde{g}^T\tilde{s} + \frac{1}{2}\tilde{H}[\tilde{s}]^2 + \frac{1}{2}\sum_{k=1}^{P}\tilde{s}_k^3 + \frac{1}{6}\sum_{k=1}^{P}\gamma_k\tilde{s}_k^4 = f_0 + \tilde{g}^T\tilde{s} + \frac{1}{2}\tilde{H}[\tilde{s}]^2 + \frac{1}{2}\mathcal{T}[\tilde{s}]^3 + \frac{1}{6}\Sigma[\tilde{s}]^4,$$

where $\tilde{s} = [\tilde{s}_1, \ldots, \tilde{s}_n]^T \in \mathbb{R}$, $\mathcal{T} \in \mathbb{R}^{n^3}$ is a diagonal tensor with P nonzero entries, and $\Sigma \in \mathbb{R}^{n^4}$ is a diagonal tensor with nonzero entries $\{\gamma_1, \ldots, \gamma_P\}$. Developing low-rank tensor approximations for our subproblem and efficiently incorporating these approximations into Algorithm 4.4.1 is a standalone area of study. Since this chapter focuses on designing a convergent iterative algorithmic framework for (AR3 Model), we leave the tensor approximation for future work.

4.3.2 An Algorithm for Minimizing the AR3 Model

To numerically minimize (AR3 Model), we use the global optimality characterizations outlined in (4.8). The primary objective of this subsection is to formulate an algorithm for locating the root of the (nonlinear) secular equation. We employ Cholesky factorization and Newton's iteration to find the root of the secular equation. A convergence analysis for the root-finding method is deferred to future work. We first introduce the secular equation based on the sufficient condition. We seek a vector $s(\lambda)$ which satisfies the system of equation,

$$\mathcal{B}_d(s)s := \left(H + \frac{1}{2}\Gamma_d + \lambda W\right)s = -g, \tag{4.35}$$

$$H + \frac{2}{3}\Gamma_d + \lambda W - \frac{\mathcal{L}_W}{3}W - \frac{\Lambda_W^2}{18\sigma}W \succeq 0, \tag{4.36}$$

$$\text{where} \qquad \lambda = \sigma\|s\|_W^2, \qquad \mathcal{L}_W := \Lambda_W\|s\|_W, \qquad \Gamma_d = Ts. \tag{4.37}$$

Note that in the case of m_3 with diagonal tensor (4.29), we have $\mathcal{L}_W = \hat{t}\|s\|_W, \Gamma_d = \text{diag}\left\{t_j s_j\right\}_{1\le j\le n}$. Let $D_\Gamma \in \mathbb{R}^{n\times n}$ be a diagonal matrix with diagonal entries $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ as the eigenvalues of $H + \frac{1}{2}\Gamma_d$. Let $U \in \mathbb{R}^{n\times n}$ with each column $u_1, u_2, \ldots, u_n \in \mathbb{R}^n$ correspond to the generalized eigenvectors. Then,

$$\left(H + \frac{1}{2}\Gamma_d\right)U = WUD_\Gamma, \text{ where } U^TWU = UWU^T = I_n. \tag{4.38}$$

Define $\hat{g} := U^T g$ and $\hat{s} = WU^T s$. From this, we may deduce that $s = U\hat{s}$ and $g = WU\hat{g}$. It follows from (4.35) that

$$-WU\hat{g} = -g = \Big(H + \frac{1}{2}\Gamma_d + \lambda W\Big)s = \Big(H + \frac{1}{2}\Gamma_d + \lambda W\Big)U\hat{s} \underset{(4.38)}{=} WU\big(D_\Gamma + \lambda I_n\big)\hat{s}.$$

Since U is non-singular, therefore $\hat{s} := (D_\Gamma + \lambda I_n)^{-1}\hat{g}$. For $\lambda \neq \lambda_j(\Gamma_d)$ for any $1 \leq j \leq n$,

$$\Psi_d(\lambda) := \|s(\lambda)\|_W^2 = \|\hat{s}\|^2 = \Big\|(D_\Gamma + \lambda W)^{-1}\hat{g}\Big\|^2 = \sum_{j=1}^{n} \frac{\hat{g}_j^2}{\big(\lambda + \lambda_j\big)^2} \tag{4.39}$$

where $\hat{g}_j$ is the jth component of $\hat{g} = U^T g$. Using (4.39), we convert $\lambda = \sigma\|s\|_W^2$ to the following system of nonlinear equations

$$\phi_d(\lambda) := \frac{1}{\sqrt{\Psi_d(\lambda)}} - \frac{1}{\mathcal{K}_d(\lambda)} = 0 \text{where} \quad \mathcal{K}_d(\lambda) := \big[\lambda/\sigma\big]^{1/2}. \tag{4.40}$$

Assume that the sufficient condition (4.36) is satisfied, then we have

$$\mathcal{B}(s)s = H + \frac{1}{2}\Gamma_d + \lambda W \succ \Big(H + \frac{1}{2}\Gamma_d + \lambda W\Big) + \underbrace{\Big(\frac{1}{6}\Gamma_d - \frac{1}{3}\mathcal{L}_W\Big)}_{\prec 0} \underset{(4.36)}{\succ} 0. \tag{4.41}$$

To numerically determine the root of (4.40), we employ a univariate Newton iteration combined with root-finding techniques based on Cholesky factorization. The update steps for the Newton iteration are outlined in Lemma 4.3.1, with the proof framework based on Lemma 3.2.2, [46, Lemma 6.1], and [62, Lemma 7.3.1].

Lemma 4.3.1. *Assume $H + \frac{1}{2}\Gamma_d + \lambda W \succ 0$ and $g \neq 0$, the Newton iteration updates are*

$$\Delta\lambda^{(l)} = -\frac{\phi_d(\lambda^{(l)})}{\nabla_\lambda \phi_d(\lambda^{(l)})} = \frac{\|s(\lambda^{(l)})\|_W^{-1} - \mathcal{K}_d(\lambda^{(l)})^{-1}}{\|\omega(\lambda^{(l)})\|_W^2 \|s(\lambda^{(l)})\|_W^{-3} - \big[\nabla_\lambda \mathcal{K}_d(\lambda^{(l)})^{-1}\big]}, \qquad \lambda^{(k+1)} = \lambda^{(l)} + \Delta\lambda^{(l)} \tag{4.42}$$

where $\omega(\lambda)$ is given by $(H + \frac{1}{2}\Gamma_d + \lambda W) = L(\lambda)L^T(\lambda)$ and $W^{-1}L(\lambda)\omega(\lambda) = s(\lambda)$. Also,

$$\mathcal{K}_d(\lambda)^{-1} = \big[\sigma/\lambda\big]^{1/2}, \qquad \big[\nabla_\lambda \mathcal{K}_d(\lambda)^{-1}\big]' = -\frac{1}{2}\sigma^{1/2}\lambda^{-3/2}.$$

Proof. Since $H + \frac{1}{2}\Gamma_d + \lambda W \succ 0$, the Cholesky factorization gives $H_W(\lambda) := (H + \frac{1}{2}\Gamma_d + \lambda W) = L(\lambda)L^T(\lambda)$. Differentiating $(H + \frac{1}{2}\Gamma_d + \lambda W)s(\lambda) = -g$ with respect to λ gives $(H + \frac{1}{2}\Gamma_d + \lambda W)\nabla_\lambda s(\lambda) + Ws(\lambda) = 0$. Therefore,

$$\nabla_\lambda s(\lambda) = -H_W^{-1}(\lambda)Ws(\lambda) = L(\lambda)^{-T}L(\lambda)^{-1}Ws(\lambda). \tag{4.43}$$

On the other hand, $\Psi_d(\lambda) = \|s(\lambda)\|_W^2$ has derivative $\nabla_\lambda \Psi_d(\lambda) = 2\big\langle Ws(\lambda), \nabla_\lambda s(\lambda)\big\rangle$ where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product. Substituting (4.43) gives,

$$\begin{aligned} \nabla_\lambda \Psi_d(\lambda) &= -2\big\langle Ws(\lambda), L(\lambda)^{-T}L(\lambda)^{-1}Ws(\lambda)\big\rangle \\ &= -2\big\langle L(\lambda)^{-1}Ws(\lambda), L(\lambda)^{-1}Ws(\lambda)\big\rangle = -2\big\|L(\lambda)^{-1}Ws(\lambda)\big\| = -2\big\|\omega(\lambda)\big\|^2 \end{aligned} \tag{4.44}$$

where $W^{-1}L(\lambda)\omega(\lambda) = s(\lambda)$. Consequently,

$$\nabla_\lambda(\Psi_d(\lambda)^{-1}) = \nabla_\lambda(\Psi_d(\lambda)^{-1/2}) = -\frac{1}{2}\Psi_d(\lambda)^{-3/2}\nabla_\lambda \Psi_d(\lambda) \underset{(4.44)}{=} \|s(\lambda)\|_W^{-3}\|\omega(\lambda)\|^2. \tag{4.45}$$

Substituting (4.45) into (4.40) and then applying the Newton updates $\Delta\lambda^{(l)} = -\frac{\phi_d(\lambda^{(l)})}{\nabla_\lambda \phi_d(\lambda^{(l)})}$ yields the result. □

An algorithm to solve (4.40) is provided in Algorithm 4.3.1, where we alternately update Γ_d and apply Newton steps to the minimizer of m_3. The approach is as follows: at each kth iteration, we first compute and fix $\Gamma_d := T[s^{(\kappa)}]$. Using this fixed Γ_d, we then perform a univariate Newton iteration in the second while loop to find $s^{(l)}$ such that $\phi_d(\lambda^{(l)}) = 0$. After this step, we update $s^{(\kappa+1)} := s^{(l)}$ and reset Γ_d based on the new value of $s^{(\kappa+1)}$. This iterative process is repeated until convergence, ultimately yielding an approximate root of (4.35).

Remark 4.3.1. *(Connection of Algorithm 4.3.1 to the Global Optimality) The global optimality conditions are implicitly built into Algorithm 4.3.1. Specifically, the first-order condition yields the linear system (4.35) and the sufficient global condition ensures that the matrix system is positive definite (i.e., $\mathcal{B}(s) \succeq 0$ see (4.41)). These allow us to reformulate the minimisation of m_3 as solving a sequence of positive definite linear systems. When Algorithm 1 converges, then we have an (approximate) stationary point satisfying the global necessary condition*

$$H + \tfrac{2}{3}\Gamma_d + \lambda W + \tfrac{\Lambda_W}{3}\|s_*\|_W = (H + \tfrac{1}{2}\Gamma_d + \lambda W) + \underbrace{(\tfrac{1}{6}\Gamma_d + \tfrac{\Lambda_W}{3}\|s_*\|_W)}_{\succeq 0} \succeq 0.$$

We show in Theorem 4.2.3 for sufficiently large σ the necessary and sufficient conditions coincide, then the algorithm is guaranteed to obtain the global minimiser of m_3 (whether convex or nonconvex).

Algorithm 4.3.1: Newton's Method to Solve (4.40)

Input: g, H, T, $\sigma > 0$. Maximum iterations $l_{\max}$ and $\kappa_{\max}$; accuracy levels $\epsilon_\kappa > \epsilon_l > 0$.

Initialization: Set $s^{(0)} = \mathbf{0}$, $\Gamma_d = T[s^{(0)}]$, $\lambda^{(0)} := \max\{-\lambda_{\min}[H], 0\}$. Initialize $l = 0$, $\kappa = 0$.

Main Step: while $\kappa \le \kappa_{\max}$ *and* $\left\|(H + \frac{1}{2}\Gamma_d + \lambda^{(l)}W)s^{(\kappa)} - g\right\| > \epsilon_\kappa$ **do**

$$\lambda^{(0)} := \max\left\{\lambda^{(l)}, \lambda_{\min}\left[H + \frac{1}{2}\Gamma_d + \lambda^{(l)}W\right], 0\right\}; \qquad (\lambda\text{-correction step})$$

Reset $l = 0$.

while $l \le l_{\max}$ *and* $\left|\phi_d(\lambda^{(l)})\right| \ge \epsilon_l$ **do**

Newton Step: Fix Γ_d in this while loop, update $s^{(l)}$ and $\lambda^{(l)}$ using Newton's updates.

Cholesky factorize $H + \frac{1}{2}\Gamma_d + \lambda^{(l)}W = LL^T$.

if *Cholesky factorization fails* **then**

Return, go to **Safeguard Step**.

Solve $LL^T s(\lambda^{(l)}) = -g$ and compute $\omega(\lambda^{(l)}) = L^{-1}Ws(\lambda^{(l)})$.

Update $\lambda^{(l)}$ using the Newton step in Equation (4.42) and (4.42) and $l := l + 1$.

Set $s^{(\kappa+1)} := s(\lambda^{(l)})$, $\Gamma_d := T[s^{(\kappa+1)}]$, $\kappa := \kappa + 1$.

Safeguard Step: if $\kappa = \kappa_{\max}$ *or* $l = l_{\max}$ *or Cholesky factorization fails* **then**

Return to solve using standard solvers, i.e., ARC solver.

Remark 4.3.2. *(λ Correction Step) Since we alternately update $\Gamma_d := T[s^{(\kappa)}]$ and use the Newton step to compute $s^{(\kappa+1)}$, the value $\lambda^{(l)}$ obtained from the Newton step ensures that $H + \frac{1}{2}T[s^{(\kappa)}] + \lambda^{(l)}W \succeq 0$. This is why ($\lambda$-correction step) is used to apply a slight adjustment, ensuring that $H + \frac{1}{2}T[s^{(\kappa)}] + \lambda^{(l)}W \succeq 0$ holds.*

Remark 4.3.3. *(Special Case for Diagonal T) If T is a diagonal tensor, we have $\Gamma_d = \operatorname{diag}\left\{t_j s_j\right\}_{1\le j\le n} = \boldsymbol{t}\cdot s^{(\kappa)}$ where $\boldsymbol{t} := [t_1, \ldots, t_n]^T$ and $\mathcal{L}_W$ can be explicitly expressed as $\mathcal{L}_W = \hat{t}\|s\|_W$. The sparse structure of the diagonal tensor enables the computation of Γ_d without requiring tensor-vector multiplications; instead, only vector multiplications are needed. This significantly improves both storage efficiency and computational time. Numerical results demonstrating the computational time improvement achieved with diagonal tensors that avoid tensor-vector multiplications are illustrated in Figure A.7.*

Remark 4.3.4. *In Algorithm 4.3.1, the second-order global sufficient optimality condition ensures that $\mathcal{B}$ is positive semidefinite. However, it is not guaranteed that $\mathcal{B}^{(l)} = H + \frac{1}{2}\Gamma_d + \lambda^{(l)}W$ remains positive*

semidefinite throughout the iterations, nor that the minimum obtained by Algorithm 4.3.1 satisfies the second-order global necessary optimality condition (Figure A.6). As a safeguard, whenever Cholesky factorization fails in Algorithm 4.3.1 or the maximum number of iterations is exceeded, Algorithm 4.3.1 reverts to the ARC method proposed by Cartis, Gould, and Toint [46]. The numerical results in Section 4.5 indicate that failures in the Cholesky factorization of $\mathcal{B}^{(l)}$ usually happens due to the ill-conditioning, and $\mathcal{B}^{(l)}$ typically remains positive definite throughout the algorithm.

While Algorithm 4.3.1 efficiently finds stationary points of cubic quartically regularized subproblem, further research is necessary to establish its convergence. After obtaining a stationary point from Algorithm 4.3.1, we always test whether it satisfies the second-order local optimality condition (4.3) and the second-order sufficient global optimality condition (4.10). Our numerical experiments in Section 4.5 indicate that the former is always satisfied, while the latter is usually satisfied for simple or small tensor terms or when n is small. Additionally, a hard case may occur when the eigenvector corresponding to the largest eigenvalue of $H + \frac{1}{2}\Gamma_d$, denoted as u_1, is orthogonal to g, i.e., $u_1^T g = 0$. In this scenario, a similar analysis to the trust-region subproblem [44, Sec 8.3.1] would need to be applied, resulting in non-unique global minimizers for the subproblem. We refer readers to [44, Sec 8.3.1] for a detailed discussion. To conclude, note that in this subsection, we employ Cholesky factorization and Newton's method to address the secular equation, though this is only one of several possible approaches. For instance, scalable iterative algorithms could be developed using Krylov methods, eigenvalue-based formulations, or subspace optimization. While we acknowledge the potential of these techniques for minimizing the diagonal tensor polynomial our primary aim here is to utilize the diagonal tensor polynomial in minimizing the m_3 subproblem. Consequently, a comprehensive analysis of these alternative approaches and the convergence of Algorithm 4.3.1 is left for future work.

4.4 A Diagonal Tensor Algorithmic Framework for Minimizing the AR3 Model

While we have characterized the optimality conditions for (AR3 Model) and discussed the special cases where these conditions can simplify, the next goal is to develop an efficient algorithm for minimizing a general (AR3 Model). We aim to minimize (AR3 Model) iteratively, and at each iteration, we use a quartically regularized polynomial with a diagonal tensor term (known as the Diagonal Tensor Model as given in (4.29)), to approximate (AR3 Model). As highlighted in Section 4.3.1, using the Diagonal Tensor Model to approximate m_3 is advantageous because minimizing such a model avoids the operation of dense tensor entries and has a simpler optimality condition.

In Chapter 2, we introduce the Quadratic Quartic Regularisation (QQR) method which is an iterative algorithm specifically designed to solve (AR3 Model). In QQR, the third-order tensor term $T_i[s]^3$ is approximated by a linear combination of quadratic and quartic terms. Following QQR, the Cubic Quartic Regularization (CQR) framework was introduced in Chapter 3. In the CQR method, the third-order tensor term is approximated by a single-parameter cubic term $\beta\|s\|^3$. However, since $T_i[s]^3$ can span multiple tensor directions, it is natural to consider approximating the tensor term with a more comprehensive tensor expression.

Motivated by this, in this section, we introduce the diagonal tensor model that approximates T_i using a diagonal tensor term $\mathcal{T}_i \in \mathbb{R}^{n^3}$ with diagonal entries $\{t_1^{(i)}, \ldots, t_n^{(i)}\}$. The Diagonal Tensor Model (or Diagonal Tensor Polynomial, i.e., DTM) is defined as

$$M_d(s^{(i)}, s) = f_0 + g_i^T s + \frac{1}{2} s^T H_i s + \frac{1}{6} \underbrace{\sum_{j=1}^{n} t_j^{(i)} s_j^3}_{:=\mathcal{T}_i[s]^3} + \frac{1}{4} \underbrace{(\sigma + d_i)}_{:=\sigma_d^{(i)}} \|s\|_W^4, \qquad \text{(Diagonal Tensor Model)}$$

where $W \succ 0$ is a symmetric matrix, the matrix norm is defined as $\|v\|_W := \sqrt{v^T W v}$ and $d_i > 0$. Note that each $t_j^{(i)} \in \mathbb{R}$ is a scalar and $t_j^{(i)}$ can be chosen as negative. For instance, we can use $t_j^{(i)}$ as the diagonal entries of $\nabla^2 m_3(s^{(i)})$. More details for choosing $t_j^{(i)}$ can be found in Algorithm 4.5.1 and

Section 4.3.1. The parameter d_i is an adaptive regularization weight; $\sigma_d^{(i)} = \sigma + d_i \geq 0$ ensures that M_d remains bounded from below.

There are several advantages and motivations for using (Diagonal Tensor Model) as an approximation for (AR3 Model). Firstly, a diagonal tensor term provides n degrees of freedom in the tensor direction to approximate and capture information from the third-order derivative. Many tensor data from real-world scenario lies in low-dimensional subspaces and exhibits low-rank structures. In Section 4.3.1, we will show that if (AR3 Model) has a tensor term with rank $< n$, then (Diagonal Tensor Model) can exactly recover (AR3 Model) under a change of basis (Section 4.3.1).

Since (Diagonal Tensor Model) is a quartic polynomial, the optimality conditions specified in Theorem 4.2.1 and Theorem 4.2.2 apply to it. Due to the special diagonal structure of the tensor in (Diagonal Tensor Model), the optimality conditions are significantly simplified, allowing for an explicit expression of Λ_W. According to Lemma 4.2.1, we have

$$\Lambda_W = \hat{t}_i = \max\{|t_1^i|, |t_2^i|, \ldots, |t_n^i|\}[\lambda_{\min}(W)]^{-3/2}. \tag{4.46}$$

The global necessary conditions for (Diagonal Tensor Model) state that if $s_d^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_d(s^{(i)}, s)$, let $\mathcal{B}_i(s) := H_i + \frac{1}{2}\mathcal{T}_i[s] + \sigma_d^{(i)} W\|s\|_W^2$. Then,

$$\mathcal{B}_i(s_d^{(i)})s_d^{(i)} = -g, \qquad H + \frac{2}{3}\mathcal{T}[s_d^{(i)}] + \sigma\|s_d^{(i)}\|_W^2 W + \frac{\hat{t}_i}{3}\|s_d^{(i)}\|_W W \succeq 0. \tag{4.47}$$

On the other hand, the global sufficiency conditions for (Diagonal Tensor Model) gives if

$$\mathcal{B}_i(s_d^{(i)})s_d^{(i)} = -g, \qquad H + \frac{2}{3}\mathcal{T}[s_d^{(i)}] + \sigma\|s_d^{(i)}\|_W^2 W - \frac{\hat{t}_i}{3}\|s_d^{(i)}\|_W W - \frac{\hat{t}_i^2}{18\sigma} W \succeq 0. \tag{4.48}$$

Then, $s_d^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_d(s^{(i)}, s)$.

Lastly, notice that the sparse structure of the diagonal tensor term and the expression of Λ_W in (4.46) allow us to develop a tensor-free framework for minimizing (AR3 Model). Operations related to the third-order term and the computation of $\hat{t}_i$ can be effectively converted into vector operations with size $\mathcal{O}(n)$. Specifically, let $\mathbf{t}^{(i)} = [t_1^{(i)}, \ldots, t_n^{(i)}]^T$, then we have $\mathcal{T}_i[s] = \mathbf{t}^{(i)} \cdot s$, $\mathcal{T}_i[s]^2 = \mathbf{t}^{(i)} \cdot s \cdot s$, and $\mathcal{T}_i[s]^3 = \mathbf{t}^{(i)} \cdot s \cdot s \cdot s$, where $\cdot$ represents the entrywise dot product. Therefore, the evaluation of $M_d(s^{(i)}, s)$ and its derivatives, remains tensor-free, as we avoid operating with the $\mathcal{O}(n^3)$ (dense) tensor term T. This framework saves both storage and computational time, allowing for a more efficient algorithm. Numerical experiments illustrating this efficiency can be found in Section 4.5 and Figure A.7. We introduce in Algorithm 4.4.1 the DTM algorithmic framework for minimizing (AR3 Model) following a similar adaptive regularization framework as Chapter 3.

It is worth noting that the Diagonal Tensor framework can be applied directly to the minimization of an objective function $f : \mathbb{R}^n \to \mathbb{R}$ under suitable assumptions (not just the AR3 subproblem). Specifically, if f is three times continuously differentiable, bounded below, and the third derivative is globally Lipschitz continuous, then Algorithm 4.4.1 can also be used for the minimization of a potentially nonconvex objective function. However, our interest here is not to create a general method for optimizing f; instead, we focus on efficiently minimizing the local model $m_3(s)$.

Remark 4.4.1. ***(Minimization of** M_d**)*** *While our proof in Section 4.4.1 (in particular Lemma A.2.4) assumes exact minimization of $\nabla M_d(s^{(i)}, s_d) = 0$, similar bounds can be derived when considering approximate minimization of M_d with a step-termination condition, such as $\|\nabla M_d(s^{(i)}, s_d)\| \leq \Theta\|s_d\|_W^3$ for some $\Theta \in (0, 1)$. Such step-termination conditions are also used in [53], with alternative variants available [46].*

4.4.1 Convergence and Complexity

Algorithm 4.4.2 is a variant of the Diagonal Tensor algorithmic framework (Algorithm 4.4.1) for minimizing m_3 that we prove an optimal worst-case complexity bound for Algorithm 4.4.2. The convergence and complexity analysis of Algorithm 4.4.2 is an extension of the CQR algorithm in

Algorithm 4.4.1: Diagonal Tensor Algorithmic Framework with Adaptive Regularization

Initialization: Set $s^{(0)} = \mathbf{0} \in \mathbb{R}^n$ and $i = 0$. An initial regularization parameter $d_0 = 0$, constants $\eta_1 > \eta > 0$, $\gamma > 1 > \gamma_2 > 0$.

Input: $f_0 := m_3(s^{(0)})$, $g_0 := \nabla m_3(s^{(0)})$, $H_0 := \nabla^2 m_3(s^{(0)})$, a symmetric $n \times n$ matrix $W \succ 0$, $\sigma > 0$ a given regularization parameter; an accuracy level $\epsilon > 0$.

Step 1: Test for termination. If $\|g_i\| = \|\nabla m_3(s^{(i)})\| \leq \epsilon$, terminate with the ϵ-approximate first-order minimizer $s_\epsilon = s^{(i)}$.

Step 2: Step computation. Use the coefficient $g_i, H_i, \mathcal{T}_i, \sigma, d_i$ to the (Diagonal Tensor Model), M_d and compute $s_d^{(i)} = \operatorname{argmin}_{s \in \mathbb{R}^n} M_d(s^{(i)}, s)$ using Algorithm 4.3.1. (see Remark 4.4.1).

Step 3: Acceptance of trial point. Compute $m_3(s^{(i)} + s_d^{(i)})$ and define

$$\rho_i = \frac{m_3(s^{(i)}) - m_3(s^{(i)} + s_d^{(i)})}{m_3(s^{(i)}) - M_d(s^{(i)}, s_d^{(i)})}. \qquad \text{(Ratio Test)}$$

Step 4: Regularization Parameter Update. if $\rho_i \geq \eta$. Successful Iter. **then**
  Set $s^{(i+1)} = s^{(i)} + s_d^{(i)}$, update g_{i+1}, H_{i+1} at $s^{(i+1)}$; get $\mathcal{T}_{i+1}$ using Algorithm 4.5.1; $d_{i+1} = d_i$.

else if $\rho_i \geq \eta_1$. Very Successful Iter. **then**
  Set $s^{(i+1)} = s^{(i)} + s_d^{(i)}$, update g_{i+1}, H_{i+1} at $s^{(i+1)}$; get $\mathcal{T}_{i+1}$ using Algorithm 4.5.1; decrease $d_{i+1} = \gamma_2 d_i$.

else if $\rho_i \leq \eta$. Unsuccessful Iter. **then**
  Set $s^{(i+1)} = s^{(i)}$, $g_{i+1} = g_i$, $H_{i+1} = H_i$, $\mathcal{T}_{i+1} = \mathcal{T}_i$; increase $d_{i+1} = \gamma d_i$.

Repeat with $i := i + 1$.

Algorithm 4.4.2: Diagonal Tensor Variant of Algorithm 4.4.1 for minimizing m_3

Fix parameters $\hat{B} > 0$ and $\alpha \in (0, \frac{1}{2})$. Initialize and proceed through Steps 1 to 3 as Algorithm 4.4.1. In Step 4, we use the following rule for updating the regularization parameter, $\mathcal{T}_{i+1}$ and $s^{(i+1)}$.

if $\|\nabla m_3(s^{(i)} + s_d^{(i)})\| \leq \epsilon$ **then**
  Terminate with an ϵ-approximate first-order minimizer s_ϵ of $m_3(s^{(i)} + s_d^{(i)})$.

else
  Let $\beta_i := \mathcal{T}_i \left[\frac{s_d^{(i)}}{\|s_d^{(i)}\|_W} \right]^3$ where $\mathcal{T}_i$ is updated using diagonal or low rank rule in Algorithm 4.5.1.
  if $\rho_i \geq \eta$ *and* $\beta_i \geq \alpha$ **then**
    Successful Step
  else if $\rho_i \geq \eta$ *and* $\beta_i \leq -4\alpha$ *and* $\sigma_d^{(i)} \notin \left[\frac{1}{6}, \frac{2}{3}\right]\big(-\beta_i + \alpha\big)\|s_d^{(i)}\|_W^{-1}$ **then**
    Successful Step
  else if $\rho_i \geq \eta$ *and* $\beta_i \in [-4\alpha, \alpha]$ *and* $\sigma_d^{(i)} \geq \frac{2}{3}\big(-\beta_i + \alpha\big)\|s_d^{(i)}\|_W^{-1}$ **then**
    Successful Step
    In all *Successful Step*, update $s^{(i+1)} := s^{(i)} + s_d^{(i)}$, update $0 \leq d_{i+1} \leq d_i$, update $\mathcal{T}_i$ with each diagonal entry $t_j^{(i)}[\lambda_{\min}(W)]^{-3/2} \leq \hat{B}$, and compute g_{i+1}.
  else
    Unsuccessful Step: $s^{(i+1)} := s^{(i)}$, $\beta_{i+1} := \beta_i$ and increase $d_{i+1} := \gamma \max\{1, d_i\}$.

Chapter 3. Since the complexity proofs in this section follow a similar framework to those presented in Section 3.3.1 and [44, Sec 2.4.1], we give the details of the proof in Section A.2.3 and provide only a proof sketch and the conclusions in this section.

In this variant, we introduce a positive parameter $\hat{B}$ that controls the magnitude of the third-order approximations $\mathcal{T}_i$, preventing them from becoming excessively large.

Assumption 4.4.1. *For every iteration, i, the scalars $\{t_j^{(i)}\}_{1\le j\le n}$ are uniformly bounded. Specifically, $\max_{i,1\le j\le n} |t_j^{(i)}|[\lambda_{\min}(W)]^{-3/2} \le \hat{B}$.*

Remark 4.4.2. *Using Lemma 4.2.1 and Assumption 4.4.1, we obtain that for every iteration, i,*

$$\max_{u,v\in\mathbb{R}^n} \mathcal{T}_i[u][v]^2 \le \max\{|t_1^{(i)}|, |t_2^{(i)}|, \dots, |t_n^{(i)}|\}[\lambda_{\min}(W)]^{-3/2} \le \hat{B}\|v\|_W^2\|u\|_W.$$

In other words, Assumption 4.4.1 implies that, for all i, $|\mathcal{T}_i[s]^3| \le \hat{B}\|s\|_W^3$ and $\|\mathcal{T}_i[s]^2\| \le \hat{B}\|s\|_W^2$ for any $s \in \mathbb{R}^n$.

In this variant, similar to the CQR Algorithm in Chapter 3, we introduce an additional constant, $\alpha \in [0, \frac{1}{2}]$, to further control the step length $s_d^{(i)}$. With this parameter, we account for two additional cases where the step is rejected and regularization is increased. Define $\beta_i := \mathcal{T}_i\left[\frac{s_d^{(i)}}{\|s_d^{(i)}\|_W}\right]^3$. The first case occurs when $\beta_i \le -4\alpha$ and σ_d falls within the narrow range $\left[\frac{1}{6}, \frac{2}{3}\right)(-\beta_i+\alpha)|s_d^{(i)}|^{-1}$. The second case arises when β_i lies in the range $[-4\alpha, \alpha]$ and $\sigma_d^{(i)} \ge \frac{2}{3}(-\beta_i+\alpha)|s_d^{(i)}|^{-1}$. For all other cases where β_i and $\sigma_d^{(i)}$ do not match these scenarios, the standard procedure in Step 4 Algorithm 4.4.1 is sufficient to ensure the success of the step. More details can be found in Theorem A.2.5.

It is worth noting that whenever $\rho_i \ge \eta$, even without the specific requirement for $\sigma_d^{(i)}$ in Algorithm 4.4.2, one could ensure a decrease in the value of m_3 at each successful iteration (Remark A.2.4), such that $m_3(s^{(i)}) - m_3(s^{(i+1)}) \ge \eta\left[m_3(s^{(i)}) - M_d(s^{(i)}, s_d^{(i)})\right] > 0$ where $\eta \in (0, 1]$. Therefore, the Diagonal Tensor framework (Algorithm 4.4.1) guarantees convergence to an approximate first-order local minimum of m_3. However, our goal in this section is to establish a global complexity bound; hence, we introduce the additional control parameter (i.e., $\alpha, \hat{B}$ and updating rules for β_i) in Algorithm 4.4.2.

To establish the complexity bound for Algorithm 4.4.2, we first derive an upper bound on the step size. Then, use the upper bound of step size to give an iteration-independent upper bound on the tensor term. The proof for the upper bound on step size is adapted from Theorem 3.3.1. The key modification is changing the regularization norm from $\|\cdot\|$ to $\|\cdot\|_W$ and substituting $\left|\beta_i\|s_d^{(i)}\|^3\right| \le \hat{B}\|s_d^{(i)}\|$ with $|\mathcal{T}_i[s_d^{(i)}]^3| \le \hat{B}\|s_d^{(i)}\|_W^3$, more details can be found in Theorem A.2.4 and Corollary A.2.1.

Next, we establish a lower bound for the step size. The lower bound on step size is essential for ensuring a reduction in the values of m_3 ([23, Lemma 2.3] and [53, Lemma 3.3]), as it shows that the step cannot diminish to an arbitrarily small size relative to the criticality conditions ($\|\nabla m_3(s^{(i)} + s_d^{(i)})\| > \epsilon$ and $\|g_i\| > \epsilon$), for both successful and unsuccessful iterations. With slight modification by replacing $\left|\beta_i\|s_d\|^2\right| \le \hat{B}\|s_d\|^2$ with $\|\mathcal{T}_i[s_d]^2\| \le \hat{B}\|s_d\|_W^2$, techniques in Lemma 3.3.1 give that the lower bound on the step is

$$\|s_d^{(i)}\|_W > \min\left\{(\hat{B}+L_H)^{-1/2}\epsilon^{1/2}, \frac{1}{2}d_i^{-1/3}\epsilon^{1/3}\right\}$$

where ϵ represents the prescribed first-order optimality tolerance

Moreover, the upper and lower bounds for d_i are

$$d_i > \frac{\hat{B}+L_H}{6}\|s_d^{(i)}\|_W^{-1} > 0, \qquad d_i \le d_{\max} := \gamma(\hat{B}+L_H)^{3/2}\epsilon^{-1/2}.$$

In practice, we can increase d_i until $d_i > \frac{\hat{B}+L_H}{6}\|s_d^{(i)}\|_W^{-1} > 0$ is satisfied. This procedure is well-defined since $\|s_d^{(i)}\|_W = O(d_i^{-1/3})$ as d_i increases, and hence $\|s_d^{(i)}\|_W^{-1} = O(d_i^{1/3})$. Consequently, the right-hand

side grows only like $d_i^{1/3}$, whereas the left-hand side grows linearly in d_i, so the inequality is eventually satisfied. The full proofs are given in Lemma A.2.4, Theorem A.2.5 and Lemma A.2.5 for completeness. The rationale behind upper bound for d_i is that whenever the regularization parameter surpasses $d_{\max}$, $M_d(s^{(i)}, s)$ overestimates the value of m_3 at the minimizer of M_d (as stated in Theorem A.2.5). This overestimation implies that any reduction achieved in minimizing $M_d(s^{(i)}, s)$ is as a guaranteed lower bound on the decrease in m_3, thereby ensuring that each iteration is successful and does not need to further increase the regularization parameter.

Next, we show that the diagonal tensor model constructed in successful steps of Algorithm 4.4.2 yields a decrease in the value of m_3 by at least $\mathcal{O}(\epsilon^{3/2})$.

Theorem 4.4.1. ***(Bounding the m_3 decrease on successful iterations[1])*** *Suppose that Assumption 4.4.1 holds and Algorithm 4.4.2 is used. For $\alpha \in (0, \frac{1}{2})$ and $\eta \in (0, 1]$, assume that $i+1$ is a successful iteration which does not meet the termination condition, i.e., $\|g_{i+1}\| > \epsilon$ and $\|g_i\| > \epsilon$. Then,*

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\|s_d^{(i)}\|_W^3 \geq \kappa_s \epsilon^{3/2} \tag{4.49}$$

where $\kappa_s = \frac{\alpha\eta}{192\gamma}(\hat{B} + L_H)^{-3/2}$.

Lastly, we establish the bound for unsuccessful iterations and total iterations, thereby giving the complexity bound for Algorithm 4.4.2.

Theorem 4.4.2. ***(Complexity bound for Algorithm 4.4.2[2])*** *Let m_{low} be a lower bound on $m_3(s)$ for $s \in \mathbb{R}^n$. Suppose that Assumption 4.4.1 holds. Then, there exists a positive constant κ_s (see Theorem 4.4.1), κ_c such that Algorithm 4.4.2 requires at most*

$$\frac{m_3(s^{(0)}) - m_{low}}{\kappa_s \epsilon^{3/2}} + \kappa_c + \log \epsilon^{-1/2} = \mathcal{O}(\epsilon^{-3/2}).$$

function evaluations (i.e., f_i) and at most

$$\frac{m_3(s^{(0)}) - m_{low}}{\kappa_s \epsilon^{3/2}} + 1 = \mathcal{O}(\epsilon^{-3/2}).$$

derivatives evaluations (i.e., g_i, H_i, T_i) to compute an iterate s_ϵ such that $\|g_i\| \leq \epsilon$ or $\|g_{i+1}\| \leq \epsilon$.

Improved Complexity for Special Cases: Although Algorithm 4.4.2 achieves a function value reduction of $\mathcal{O}(\epsilon^{3/2})$ for a general nonconvex m_3, in certain special cases, the DTM algorithm or its variants exhibit improved complexity bounds during specific iterations. One such instance is when the *tensor term is sufficiently small*. If $\|T_i\| \leq \epsilon^{1/3}$, and we entries of $\mathcal{T}_i$ with absolute value less than $\epsilon^{1/3}$, then the decrease in value of m_3 in such an iteration is $\mathcal{O}(\epsilon^{4/3})$. The analysis for this case follows similarly to (3.46)

Secondly, if m_3 has a *diagonal tensor*, we can set $\mathcal{T}_0 = T$ and $d_0 = 0$, (Diagonal Tensor Model) exactly recovers (AR3 Model). As a result, Algorithm 4.4.1 converges in one iteration. Lastly, Theorem 4.3.1 indicates that if (AR3 Model) has *a tensor with rank $P \leq n$* as illustrated in (4.31), then by setting $\tilde{g}$, $\tilde{H}$, the diagonal tensor $\mathcal{T}$ and $\tilde{W}$ as indicated in Theorem 4.3.1, we can minimize $m_3^{\mathbb{P}}(s)$ by minimizing the following diagonal tensor model

$$\tilde{s}_* = \min_{\tilde{s}\in\mathbb{R}^n} f_0 + \tilde{g}^T\tilde{s} + \frac{1}{2}\tilde{s}^T\tilde{H}\tilde{s} + \frac{1}{6}\mathcal{T}[\tilde{s}]^3 + \frac{\sigma}{4}\|\tilde{s}\|_{\tilde{W}}^4 \tag{4.50}$$

and recover the minimizer by $s_* = \min_{s\in\mathbb{R}^n} m_3^{\mathbb{P}} = C^{-1}\tilde{s}_*$. A slight modification to Algorithm 4.4.1 can yield an algorithm for minimizing (AR3 Model) with a tensor rank $P \leq n$. By adjusting the inputs of Algorithm 4.4.1 as follows: set $d_0 = 0$, $f_0 := m_3(s^{(0)})$, $g_0 := C^{-T}\nabla m_3(s^{(0)})$, $H_0 := C^{-T}\nabla^2 m_3(s^{(0)})C^{-1}$, and $\tilde{W} := C^{-T}WC^{-1}$. Also, modify Step 2 as $s_d^{(i)} = C^{-1}\operatorname{argmin}_{s\in\mathbb{R}^n} M_d(s^{(i)}, s)$. With these adjustments, Algorithm 4.4.1 converges to the minimizer of m_3 in a single iteration, i. The modified variant of Algorithm 4.4.1 for minimizing $m_3^{\mathbb{P}}$ with low-rank tensor is outlined in Algorithm 4.5.1 with numerical examples in Section 4.5.1.

[1]The proof of Theorem 4.4.1 is adapted from Lemma 3.3.1 and provided in Section A.2.3.1.
[2]The proof of Theorem 4.4.2 is adapted from Theorem 3.3.4 and provided in Section A.2.3.2.

4.5 Numerical Implementation and Preliminary Results

In this section, we offer two options for updating $\mathcal{T}_i$ in Algorithm 4.5.1, thereby completing the structure of Algorithm 4.4.1. The first option for updating $\mathcal{T}_i$ is suitable for a general m_3 or m_3 with a diagonal tensor. The second option for updating $\mathcal{T}_i$ is suitable for $m_3^{\mathbb{P}}$ with a low-rank tensor (rank $P \leq n$) and converges to the minimizer of $m_3^{\mathbb{P}}$ in a single iteration.

Algorithm 4.5.1: An Implementable Diagonal Tensor Method (DTM) Variant for Minimizing $m_3(s)$

Set a fixed bound $\hat{B} := 10^6$. Initialize and proceed through Algorithm 4.4.1 with specifications as follows. In Steps 2 and 4, update $\mathcal{T}_i$ and W using one of the following rules:

1. **Diagonal Rule**: Set $\mathcal{T}_i = \text{diag}\{t_j^{(i)}\}_{1\leq j\leq n}$, where $t_j^{(i)} = T_i[j,j,j]$ are the diagonal entries of T_i. If $T_i[j,j,j] > \hat{B}$, set $t_j^{(i)} := \text{sign}(T_i[j,j,j])\hat{B}$.

2. **Low Rank Rule**: Compute a rank-P approximation of T_i, denoting $\hat{C} := \left[\boldsymbol{a}^{(1)}, \ \ldots, \ \boldsymbol{a}^{(P)}\right]^T$. Set

$$C := \left[\hat{C}, \ \hat{C}_\perp\right]^T, \quad \tilde{g}_i = C^{-T} g_i, \quad \tilde{H}_i = C^{-T} H_i C^{-1}, \quad \tilde{W}_i = C^{-T} W C^{-1}, \quad \mathcal{T}_i = \text{diag}\{t_j\}_{1\leq j\leq n}$$

Recover $s_d^{(i)} = C^{-1}\tilde{s}_d^{(i)}$.

Remark 4.5.1. *(Role of $\hat{B}$) In Algorithm 4.5.1, the parameter $\hat{B}$ prevents the diagonal terms $|t_j^{(i)}|$ from becoming excessively large, thereby preventing potential numerical issues.*

Remark 4.5.2. *(Tensor-free nature) Note that in Algorithm 4.5.1, T_i is the full third-order derivative, represented as an $n \times n \times n$ tensor. $\mathcal{T}_i$ is the diagonal tensor used in the DTM to approximate T_i, and it has a sparse structure. The Diagonal Rule is tensor-free as we only access the n diagonal elements of T_i. All tensor-vector, tensor matrix multiplication is evaluated using $\mathcal{O}(n)$ operations, $\mathcal{T}_i[s] = \boldsymbol{t}^{(i)} \cdot s$, $\mathcal{T}_i[s]^2 = \boldsymbol{t}^{(i)} \cdot s \cdot s$, and $\mathcal{T}_i[s]^3 = \boldsymbol{t}^{(i)} \cdot s \cdot s \cdot s$, where $\cdot$ represents the entrywise dot product. The Low-Rank Rule obtains the rank-P approximation of T_i using `Eigenproblems` Functionality in Tensor Toolbox [12; 13]. Besides the step of extracting low-rank approximation, the rest of the Low-Rank Rule also does not involve tensor operations.*

We give the flow chart of the DTM framework in Figure 4.1. Algorithm 4.4.1 is the main DTM framework. Algorithm 4.4.1 calls Algorithm 4.5.1 as a subroutine to update the diagonal tensor $\mathcal{T}_i$, and calls Algorithm 4.3.1 as a subroutine to compute the step by minimizing the DTM model. Algorithm 4.4.2, by contrast, is a more detailed DTM variant used exclusively for establishing worst-case global evaluation complexity bounds in Section 4.4.1.

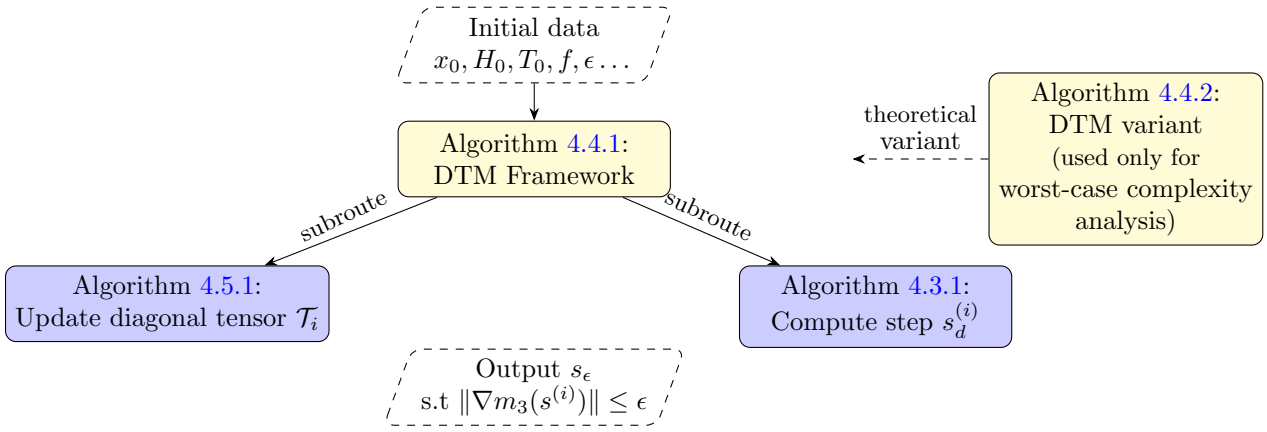
4.5.1 Minimizing m_3 with Diagonal, Low Rank and Full Tensor

The natural test cases for the DTM algorithmic framework (Algorithm 4.4.1) with diagonal updates or low-rank updates (Algorithm 4.5.1 as subroute) are, respectively, the minimisation of (AR3 Model) with diagonal tensor terms and the minimisation of (AR3 Model) with low-rank tensor terms. Therefore, in this subsection, we test the DTM algorithm to these test sets, along with a standard test set of m_3 with a full-rank tensor term.

1. **Diagonal tensor test set**: The objective polynomial is $m_3(s) = g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}\sum_{j=1}^n t_j s_j^3 + \frac{\sigma}{4}\|s\|^4$ where g, H, and t_j are generated as follows

$$g = \texttt{a*randn(n, 1)}, \quad H = \texttt{b*symm(randn(n, n))}, \quad t = [t_1, \ldots, t_n]^T = \texttt{c*randn(n, 1)} \quad (4.51)$$

Figure 4.1: Flow chart of the DTM framework.



2. **Low rank test set**: The objective polynomial is $m_3(s) = g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}\sum_{k=1}^{P}(\boldsymbol{a}^{(k)^T} s)^3 + \frac{\sigma}{4}\|s\|^4$. Specifically, g, H are generated the same as the Diagonal tensor test set and $\boldsymbol{a}^{(k)} =$ `c*randn(n, 1)` for $j = 1, \ldots, p$.

3. **AR3 subproblem test set**: The objective polynomial is the same as (AR3 Model). Specifically, g, H are generated the same as the Diagonal tensor test set, and $T =$ `c*symm(randn(n, n, n))` is generated using the Tensor Toolbox [12; 13].

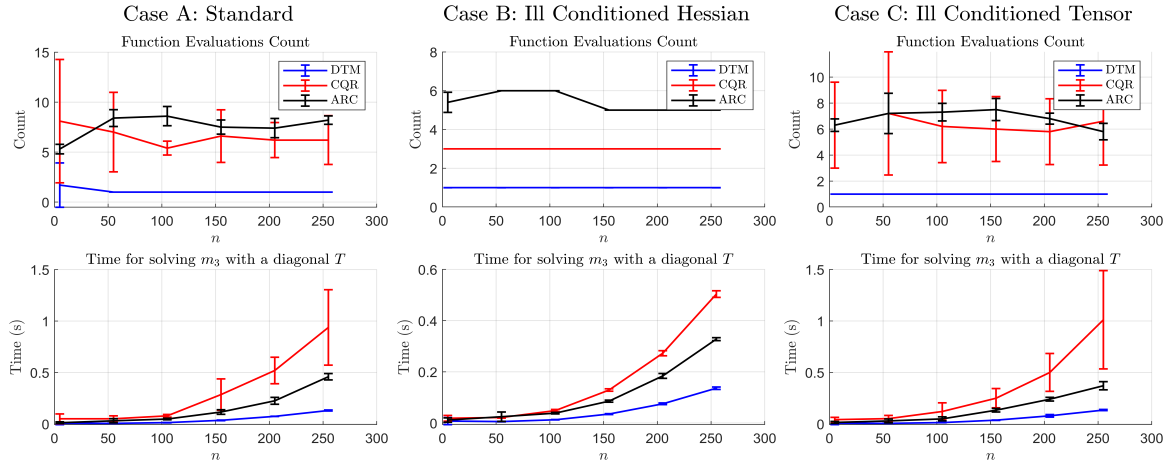
Here, `n` represents the dimension of the problem ($s \in \mathbb{R}^n$), and `symm(randn())` denotes a symmetric matrix with entries following a normal distribution with mean zero and variance one. The parameters `a`, `b`, `c`, and σ are selected differently to test the algorithm's performance under various scenarios. We occasionally assign specific values to H and t to test the algorithm's performance on ill-conditioned Hessians and tensors. The code is written in double precision. Unless otherwise stated, the parameters are set at $\eta = 0.1, \eta_1 = 2, \gamma_2 = 0.5, \gamma = 2.0$. In our numerical experiments, unless otherwise specified, we set the tolerance and stopping criterion as $\|\nabla^2 m_3(s_\epsilon)\| \leq$ `TOL`, where `TOL` $= 10^{-4}$ ($\epsilon_l = 10^{-8}$, $\epsilon_\kappa = 10^{-7}$). After obtaining s_ϵ, we tested both first-order and second-order criticality conditions ($\nabla^2 m_3(s_\epsilon) \succeq -$`TOL`). In all our test examples, the algorithm converged to an approximate second-order minimum of (AR3 Model).

Diagonal tensor test set: In Figure 4.2, we tested the DTM algorithmic framework (Algorithm 4.4.1) with diagonal updates (Algorithm 4.5.1 as subroute) on the diagonal tensor test set for problem dimensions up to 250 and compared it to the state-of-the-art Adaptive Regularization with Cubics (ARC) method [46][3] and Cubic Quartic Regularization (CQR) method Chapter 3. We tested three subsets of diagonal tensor in (4.51): (A) randomly generated g, H, and t; (B) ill-conditioned Hessian; and (C) ill-conditioned tensors. For each subcase, 10 randomized trials were performed, and the averages across these trials with error-bar are displayed in Figure 4.2. Since M_d exactly approximates our test examples, DTM algorithm consistently converges in a single iteration (requiring one evaluation of the function and derivatives), while the ARC and CQR algorithm typically requires 5–10 function evaluations. Also, DTM achieves the lowest computational time, particularly for large dimensions. We also observed that, for ARC (and CQR), more than 50% of the total time is spent evaluating the function and its derivatives. Thus, since ARC (and CQR) requires 5–10 function and derivative evaluations while DTM requires only one, this is the main reason its computational time becomes larger than that of DTM, especially for larger n. In terms of the total number of Cholesky factorizations, DTM outperforms ARC in cases A and C, and ARC outperforms DTM in case B. In all cases, the second-order local optimality condition is satisfied, as well as the necessary global optimality condition outlined in Theorem 4.2.1. Among the three methods, CQR usually takes more evaluations than DTM but fewer than ARC. This behaviour is expected. In CQR, the tensor term is approximated using a single parameter β. When the tensor has only one dominant diagonal entry—or when a single

[3]The code for the ARC algorithm can be found at URL: https://www.galahad.rl.ac.uk/packages/

direction in the tensor is large—CQR performs well. However, when there are n significant diagonal entries, a single *beta* cannot adequately capture all tensor directions, and consequently, more function evaluations are needed. More details for the CQR method can be found in Chapter 3. The need to more effectively capture multiple tensor directions was one of the motivations behind developing DTM as a more generalized version of CQR. We also experimented with a tighter tolerance, such as 10^{-9}, and the same behaviour was observed.

Figure 4.2: **Diagonal Tensor Test Set**: Averages over 10 random trials are plotted, with error bars showing one standard deviation. The same legend applies to all subplots. Standard Set: `a = 20, b = 20, c = 20`, $\sigma = 100$ in (4.51).
Ill-Conditioned Hessian Set: `a = 10, c = 5`, H is diagonal matrix uniformly distributed in $[10^{-6}, 10^{3}]$, $\sigma = 500$ in (4.51).
Ill-Conditioned Tensor Set: `a = 10, b = 20`, t has entries uniformly distributed in $[10^{-6}, 10^{2}]$, $\sigma = 500$ in (4.51).

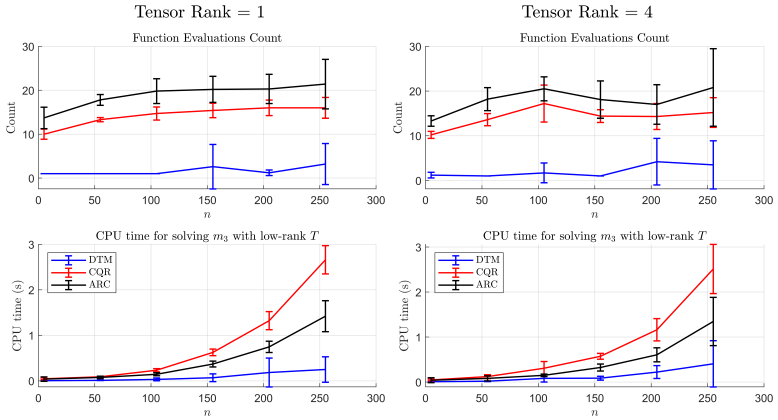


Low tensor rank test set: In Figure 4.3, we tested the DTM algorithmic framework (Algorithm 4.4.1) with low-rank updates (Algorithm 4.5.1 as subroute) on the low-rank tensor test set for tensor ranks $P = 1$ and $P = 4$. We compare DTM with both ARC algorithm and the CQR algorithm. Similar performance patterns were observed, with DTM requiring significantly fewer function evaluations and computation time than ARC/CQR in all cases. We give a brief justification of why this is the case. When the tensor has $P = 1$ or $P = 4$, we use the change of basis described in Theorem 3.1:

$$T^{\mathbb{1}}[s]^3 = \left(a_1^{(1)}s_1 + \cdots + a_n^{(1)}s_n\right)^3 \quad \underset{\text{change of basis}}{\rightsquigarrow} \quad \sum_{j=1}^{P} \tilde{s}_j^3.$$

Under this transformation, the tensor essentially becomes a sparse diagonal tensor with a few nonzero entries, $T_{1:P,1:P,1:P} = 1$ where $T_{i,j,k} = 0$ for all other indices. Consequently, the DTM subproblem can be solved in essentially one iteration, matching the behaviour observed when the tensor is already diagonal (Figure 4.2). Yet, the computation time is slightly longer than in the diagonal test set, because our computation time includes the cost of performing the change of variables. We notice that for $n \leq 100$, the performance of DTM and ARC is largely comparable, but for $n > 100$, a clear distinction emerges. This is because, when n is small, the Cholesky factorization dominates the computation time. While for larger n, the cost of function and derivative evaluations dominates. Therefore, the advantage of DTM—performing a single change of basis and solving the transformed problem in one or only a few iterations—becomes pronounced. Lastly, we note that in some DTM iterations, when solved to high accuracy, Algorithm 4.3.1 may encounter numerical issues due to the norm $\|\cdot\|_W$ being ill-conditioned, prompting us to revert to the ARC algorithm as a safeguard. This results in DTM converging in slightly more iterations, typically 3-4 iterations. Further improvements in Algorithm 4.3.1 to enhance scalability are deferred to future work.

Figure 4.3: **Low-Rank Tensor Test Set**: Parameter configurations are as follows. `a = 10, b = 20, c = 20`, $\sigma = 400$, with rank $P = 1$ and $P = 4$, respectively. The change of basis is implemented via the Low-Rank Rule for DTM in Algorithm 4.5.1 and calculated into the computation time. No change of basis for other algorithms, as their algorithmic frameworks do not naturally adapt this feature.



AR3 subproblem test set: In Table 4.1, we tested DTM algorithmic framework (Algorithm 4.4.1) with diagonal updates (Algorithm 4.5.1 as subroute) on the full-rank AR3 subproblem with no dominant tensor direction and compared it with the ARC and CQR methods. DTM performed competitively with ARC in terms of both iterations and function evaluations, although it required slightly more iterations and evaluations than the CQR method. Generally, if the quadratically regularized polynomial is low-rank or has a diagonal tensor term, DTM is preferred. On the other hand, for polynomials with a full-rank tensor and no dominant tensor direction, ARC or CQR is preferred. Note that in this chapter, we only provide a preliminary numerical illustration of applying the DTM method for minimizing the AR3 subproblem, particularly those with low-rank or diagonal tensor terms. The practical implementation of the AR3 methods to minimize the objective function is work in itself, and we refer the reader to [54] for a full implementation of the AR3 algorithm. In [54], the AR3 solver is tested on the Moré, Garbow, and Hillstrom Test Problem Set, and acceleration techniques such as pre-rejection and pre-acceptance are applied to accelerate the convergence of the AR3 solver.

Table 4.1: **AR3 Subproblem test set**: `a = b = c = 80, TOL` $= 10^{-3}$, stopping criterion as $\|\nabla^2 m_3(s_\epsilon)\| \leq$ `TOL`. The results are averaged over 10 randomly generated test functions Algorithm 4.4.1 with the Diagonal Rule in Algorithm 4.5.1.

Method	Iteration (i.e., Successful Iter.)					Evaluations (i.e., Total Iter.)				
	$n = 15$	$n = 25$	$n = 50$	$n = 75$	$n = 100$	$n = 15$	$n = 25$	$n = 50$	$n = 75$	$n = 100$
ARC	10.5	12.9	15.4	15.1	18.3	8.6	10.9	13.4	13.1	15.9
CQR	7.3	11	10.8	11.6	13.5	6.6	8.9	9.8	10.4	12
DTM	8.1	9.5	11.6	12.7	15.7	8.1	9.5	11.6	12.7	15.7

Magnitude of $\|T\|$ or σ and Global Optimality Characterization: In Figure A.6 in Section A.2.4.1, we examine how the magnitude of $\|T\|$ or σ influences global optimality characterization. Figure A.6 illustrates that as σ or $\|T\|$ increases, the likelihood that the minimum generated by the DTM algorithm satisfies the global sufficient condition in Theorem 4.2.2 also increases, indicating that the global minimum of m_3 is achieved. This finding aligns with the theoretical results in Section 4.2.3, which demonstrate that the global necessary and sufficient conditions for the AR3 subproblem coincide when σ is sufficiently large or $\|T\|$ is sufficiently small.

Applying Algorithm 4.4.1 directly to m_3: As discussed in Section 4.4, the tensor-free structure of the DTM offers advantages in terms of computational time and storage. Note that Algorithm 4.4.1 can be set directly with $\mathcal{T}_i = T_i$, where T_i denotes the full third-order derivative rather than a diagonal tensor. In Figure A.7 in Section A.2.4.2, we compare the computational time of Algorithm 4.4.1 with $\mathcal{T}_i = T_i$ (the full third-order derivative) against the DTM framework where $\mathcal{T}_i$ is a sparse diagonal

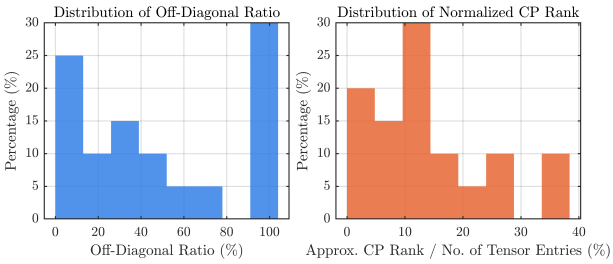


Figure 4.4: **Left:** Distribution of the off-diagonal ratio of the initial third-order tensors $T := \nabla^3 f(x_0)$, expressed as a percentage of the total Frobenius norm $\|T\|_F$. **Right:** Distribution of the approximate CP rank normalized by the number of tensor entries.

tensor. As the dimension increases, the variant with $\mathcal{T}_i = T_i$ (the full third-order derivative) becomes significantly slower. This slowdown is primarily due to the costly tensor–vector operations required by the full third-order derivative when neither tensor-free nor sparsity structures are employed.

4.5.2 Numerical Examples with Moré–Garbow–Hillstrom test set

We test our method on a subset of the Moré–Garbow–Hillstrom (MGH) test set [129] and provide preliminary numerical observations in this subsection. All problem initializations, as well as the first-, second-, and third-order derivatives, were generated following the specification in [21][4]. We compare the performance of the following algorithms:

- `ARC`: Adaptive Regularization with Cubics [46].
- `AR3 + ARC`: Third-order method with `ARC` as the inner solver [53].
- `AR3 + QQR`: Third-order method with `QQR-v2` as the inner solver Algorithm 2.5.2.
- `AR3 + CQR`: Third-order method with `CQR Choice 2` as the inner solver Algorithm 3.4.1.
- `AR3 + DTM`: AR3 with DTM subproblem solver Algorithm 4.4.1 with diagonal rule in Algorithm 4.5.1.

We first examine whether the third order derivative $T_0 := \nabla^3 f(x_0)$ in this test set exhibit diagonal or low-rank structure; the results are presented in Figure 4.4. In our analysis of the MGH problems, we observe a broad range of tensor structures. Approximately 20% of the problems have off-diagonal ratios below 7%, where offdiag_ratio $= \frac{\|T_{\text{off}}\|_F}{\|T\|_F}$ where T_{off} denotes off-diagonal entries. Several other problems, however, are dense. About 20% of the problems have an off-diagonal ratio below 7%. The CP rank measures how many outer-product components are required to reconstruct a tensor. The approximate CP rank is computed as the smallest r such that $\|T_r - T\|_F / \|T\|_F < 10^{-2}$, where T_r denotes a rank-r approximation, and is then normalized by r over number of tensor entries. For the initial third-order tensors, we find that roughly 10% of the problems are rank 1 and around 30% are rank 2. Note that the size of the MGH test set is relatively small, with dimensions ranging from $n = 2$ to $n = 11$. Whether real-world large-scale problems exhibit similar diagonal or low-rank structure remains an open question for future investigation.

The detailed numerical results for the problems in the MGH test set and comparisons is given Table 4.2. The outer-loop termination tolerance was set to $\|\nabla f\| < 10^{-3}$, and the maximum iteration count at 3000. If an inner iteration was used, its tolerance was fixed at 10^{-4} and the maximum inner iteration count at 3000. In performance tables, `eval.` refers to derivative evaluations (successful iterations only) and `iter.` denotes the total number of function evaluations (successful and unsuccessful). Performance profiles were used to compare the algorithms across the test set, following the framework of [69; 54; 79]. We adopt the notation of [20]: $\Gamma_{\hat{m}}(\tau_m)$ is the proportion of problems for which method $\hat{m}$ attained an ϵ_g-approximate solution within a computational cost no more than τ_m times the cost of

[4]The original implementations were in Fortran; our tests use a Matlab version obtained via a Fortran–Matlab converter [54].

the most efficient method, where the cost is measured in iteration or evaluation counts[5]. Formally, if $t_{\hat{m}\hat{j}}(\epsilon_g)$ denotes the cost incurred by method $\hat{m}$ on problem $\hat{j}$ to reach an ϵ_g-approximate solution, we define $t^{\hat{j}}_{\min}(\epsilon_g) := \min_{\hat{m}} t_{\hat{m}\hat{j}}(\epsilon_g)$. where $\hat{m} \in \{\texttt{ARC}, \texttt{AR3+ARC}, \texttt{AR3+QQR}, \texttt{AR3+CQR}, \texttt{AR3+DTM}\}$. If a method fails to satisfy the accuracy requirement within the iteration limit, we set $t_{\hat{m}\hat{j}}(\epsilon_g) = +\infty$.

Across the test set, all methods consistently converged to the same minimizer whenever convergence occurred, and each computed solution satisfied the second-order necessary conditions. As shown in Table 4.2, third-order methods generally require no more outer function or derivative evaluations than the second-order method, and in some cases require even fewer. Between the third-order methods, `AR3 with DTM` requires the least number of iterations on average[6], `AR3+DTM` requires 421 outer iterations—fewer than `ARC` (507 iters) and comparable to `AR3+ARC` (426 iters), `AR3+CQR` (463 iters), and `AR3+QQR` (438 iters). `AR3+QQR` requires least outer evaluations (only 293 evaluations) —fewer than `ARC` (505 iters), `AR3+ARC` (414 iters), `AR3+CQR` (446 iters), and `AR3+QQR` (410 iters). Among all methods, `ARC` remains the fastest in terms of overall computation time. For each subproblem, the DTM solver is efficient: it typically converges in 1–3 inner iterations/evaluations, whereas `AR3+ARC` generally requires 7–9 inner iterations/evaluations per subproblem. This advantage arises because each subproblem in DTM uses a diagonal tensor model, which more accurately captures the dominant diagonal structure of the AR3 subproblem tensor. This effect is evident in approximately diagonal problems—for example, Problems 2, 14, and 17—where `AR3+DTM` demonstrates stronger performance. On the other hand, for problems with a single dominant tensor entry, the CQR algorithm typically performs well. For instance, in Problem 4 (the Brown badly scaled problem), the tensor term contains a dominant entry of magnitude 10^8, while all other entries are relatively small, typically of order 1. For this problem, `AR3 with ARC` requires 1421 evaluations, whereas `AR3 with CQR` requires only 12 evaluations. Problem 10 and Problem 20 are not solved by any of the three methods within the maximum limit of 3000 iterations; this behaviour of Problem 10 is also reported in [20; 54]. In Problem 20, when the 3000-iteration limit is reached, all solvers satisfy $\|\nabla f(x_k)\| \approx 10^{-2}$ and are clearly converging but have not yet reached high accuracy. Increasing the maximum iteration limit to 6000 allows third-order methods to converge fully, but `ARC` still fails to satisfy the gradient tolerance.

Table 4.2: Comparison across ARC, AR3+ARC, AR3+QQR, AR3+CQR, and AR3+DTM over the 20 CUTEst problems, including total iteration counts, function evaluations, and per-subproblem efficiency. The average excludes Problem 10. Problems marked with − are excluded from the average count. − indicates that the algorithm failed to make progress after 3000 iterations. † indicates that the algorithm timed out: $\|\nabla f\|$ was reduced below 10^{-2} but did not reach the prescribed outer-loop tolerance of 10^{-3}.

Prob.	CP Rank	Off-Diag.	ARC		AR3+ARC				AR3+QQR				AR3+CQR				AR3+DTM			
			Iter	Eval	Iter	Eval	Iter/Sub	Eval/Sub	Iter	Eval	Iter/Sub	Eval/Sub	Iter	Eval	Iter/Sub	Eval/Sub	Iter	Eval	Iter/Sub	Eval/Sub
1	1	8.0%	2322	2322	2354	2311	4.4	4.1	2490	1874	1.5	1.9	3000	3000	1.0	1.0	2356	2312	2.39	1.25
2	1	0.3%	113	113	58	57	2.9	2.9	122	113	1.4	1.5	58	57	7.3	2.8	61	58	1.59	1.67
3	2	14.3%	26	13	31	12	7.4	18.2	31	12	3.9	9.3	31	12	4.2	5.3	31	12	2.10	5.33
4	3	100.0%	1421	1421	32	31	10.9	6.0	32	31	2.1	2.1	15	12	2.1	2.1	32	31	11.16	2.10
5	3	32.3%	66	66	58	57	2.6	2.7	70	51	1.4	1.9	58	57	3.4	2.0	58	57	1.40	1.42
6	2	7.9%	104	104	119	99	15.0	16.3	122	89	4.8	5.7	131	99	34.2	40.8	119	99	15.95	3.23
7	3	51.3%	782	781	717	716	2.2	2.2	738	609	1.3	1.6	717	716	1.4	1.2	717	716	1.19	1.19
8	3	49.8%	745	745	726	725	2.0	2.0	805	423	1.0	1.9	726	725	1.1	1.0	726	725	1.01	1.01
9	5	52.9%	31	31	23	12	5.3	10.3	27	13	1.3	2.6	27	13	1.4	2.8	27	13	1.26	2.62
10†	5	100.0%	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
11	3	100.0%	16	13	60	49	20.6	25.2	57	47	1.2	1.4	57	47	1.2	1.5	57	47	1.25	1.51
12	6	30.3%	42	42	26	25	2.8	2.9	26	25	1.5	1.6	27	25	3.0	2.2	26	25	1.42	1.48
13	2	30.5%	223	223	208	206	2.2	2.3	212	191	1.2	1.3	273	206	4.1	4.0	209	207	1.19	1.19
14	2	3.2%	116	116	119	63	13.5	21.9	117	63	9.2	15.0	119	63	24.8	20.3	123	64	4.45	7.80
15	8	98.0%	50	50	22	21	3.0	3.1	22	21	1.1	1.1	22	21	5.8	2.0	22	21	1.14	1.19
16	3	16.7%	91	91	92	85	3.5	3.5	92	85	3.1	3.1	153	85	6.5	7.4	92	85	4.78	2.15
17	2	3.2%	400	387	173	148	13.4	14.9	173	148	4.7	4.2	180	155	59.3	7.6	173	148	2.71	3.11
18	20	100.0%	42	42	17	16	3.2	3.4	17	16	2.1	2.2	17	16	2.0	2.1	17	16	1.94	2.06
19	4	94.0%	38	38	251	238	28.8	30.4	159	146	1.1	1.2	159	146	1.1	1.2	159	146	1.11	1.21
20†	20	67.9%	3000	3001	3001	3001	2.0	2.0	3001	1611	1.0	1.9	3001	3001	1.0	1.0	3001	3001	1.01	1.01
Average			**507**	**505**	**426**	**414**	**7.7**	**9.2**	**438**	**293**	**2.4**	**3.2**	**463**	**446**	**8.7**	**5.7**	**421**	**410**	**3.1**	**2.2**

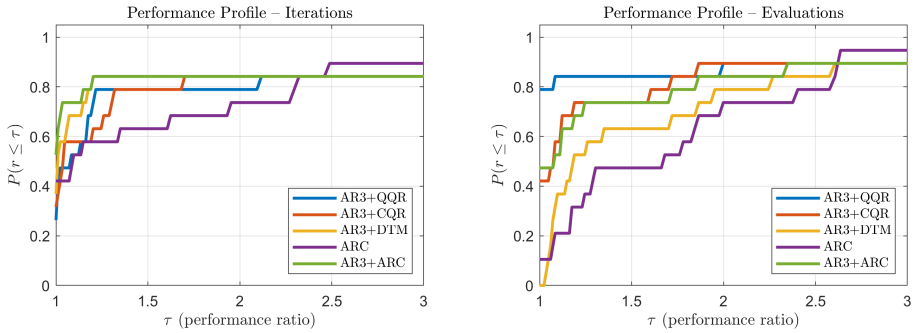
4.5.3 A Low Rank Example: Packing Problem

We consider the problem of packing N identical small spheres with radius $\tilde{r} > 0$ within a large sphere of radius $\tilde{R} \geq \tilde{r}$. By packing, we mean that the small spheres must be placed within the large sphere without overlapping ([27; 25; 24]). Following the formulation presented by Birgin et al, the problem

[5]Some comments regarding CPU time are given at the end of the section.

[6]The average excludes Problem 10.

Figure 4.5: Performance profile plots for ARC, AR3+ARC, AR3+QQR, AR3+CQR, and AR3+DTM.

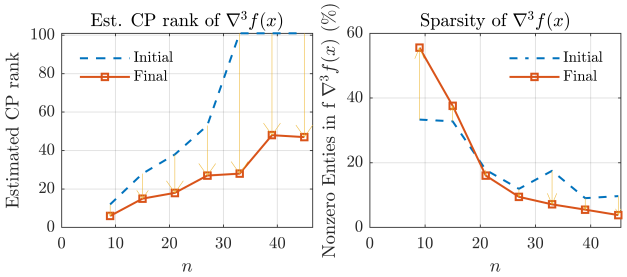


can be modeled as follows

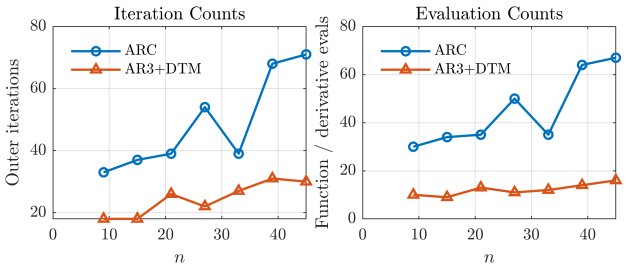
$$\min_{x\in\mathbb{R}^n} f_{pp}(x) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} \underbrace{\max\left\{0,\ (2\tilde{r})^2 - \|c_i - c_j\|^2\right\}^4}_{\text{overlapping between the small spheres}} + \sum_{i=1}^{N} \underbrace{\max\left\{0,\ \|c_i\|_2^2 - (\tilde{R} - \tilde{r})^2\right\}^4}_{\text{fitting within the large sphere}}. \tag{4.52}$$

where $n = 3N$, $x = (c_1^T, \ldots, c_N^T)^T$ represents N the small balls are the centers $c_1, \ldots, c_N \in \mathbb{R}^3$. A solution of the packing problem corresponds to a solution of problem (4.52) at which $f_{pp}(x)$ is approximately minimized to zero. Following [26, §13.1], the packing formulation considered here is central to the approach implemented in Packmol [126], widely used for initializing molecular dynamics systems. The radii $\tilde{r}$ and $\tilde{R}$ are related through $\tilde{R} \approx \tilde{r}\sqrt[3]{N}/0.3$, meaning that the small spheres occupy approximately 30% of the volume of the large sphere.

We initialize the centres of the small spheres near the origin using $x_0 = 0.1\mathcal{N}(0, I_n)$ and examine how the CP rank and sparsity of the third-order tensor $\nabla^3 f(x)$ evolve throughout the optimization process. As illustrated in Figure 4.6a, the tensors arising from such random initializations are not low rank. However, as the spheres gradually redistribute within the enclosing region and the iterates are driven toward a non-overlapping arrangement that fills the volume with radii $\tilde{R}$, the tensors computed at the final iterates begin to exhibit lower CP rank. An observation is that, for increasing dimension n, the estimated CP rank grows approximately linearly with n, rather than with the full n^3 degrees of freedom in the tensor. We also quantify sparsity by the proportion of nonzero tensor entries, and the results indicate that the tensor becomes sparser as the problem size increases. The sparsity structure becomes more pronounced as the iterates approach the minimizer. These preliminary findings suggest that third-order tensors arising in realistic optimization models may possess exploitable structure, such as approximate low rank or sparsity, even when such structure is absent at initialization. We compare the performance of `ARC` and `AR3+DTM` on this class of packing problems, with the results summarized in Figure 4.6b. While `ARC` remains the fastest method in terms of computation time, `AR3+DTM` requires substantially fewer outer iterations and derivative evaluations to reach a minimizer of f_{pp}. These experiments are preliminary numerical illustrations, highlighting the potential benefits of exploiting tensor structure within higher-order models. Note that we employed the standard `CP_ALS` algorithm [13] to obtain an estimate of the best CP rank of a tensor via an alternating least-squares procedure. More advanced low-rank update strategies tailored to third-order methods remain an active area of research. Recent developments, such as the quasi-Newton–type updates proposed in [183] for approximating third derivatives from Hessian evaluations, may offer significant improvements in computational efficiency. A scalable, low-rank implementation of DTM is beyond the scope of the present work and constitutes a direction for our future research.



(a) Approximate CP rank and sparsity of $\nabla^3 f(x)$ at initialization and after convergence.



(b) Iteration and evaluation counts for ARC and AR3+DTM.

Figure 4.6: Performance of `ARC` and `AR3+DTM` on the sphere packing problem (right), and the structure of the third-order tensor $\nabla^3 f(x)$ before and after optimization (left). In all experiments, the small-sphere radius is fixed at $r = 1$, and the $R = r\,(N/0.3)^{1/3}$. The problem dimension is $n = 3N$, initialized at $x_0 = 0.1\mathcal{N}(0, I_n)$. Each method is terminated once $\|f_{pp}(x)\| < 10^{-4}$. The approximate CP rank is computed as the smallest r such that $\|T_r - T\|_F / \|T\|_F < 5\%$.

Chapter 5

Global Optimization Techniques for Regularized Polynomial Subproblems

5.1 Introduction

Through the investigation of iterative solvers for AR3 subproblems (such as QQR, CQR, and DTM), we are gradually approaching the core of a key theoretical question,

> "*Does there exist an efficient and tractable algorithm for globally minimizing a sufficiently regularized ARp subproblem ($p \geq 3$)?*"

Verifying the existence of a global minimum is closely tied to determining whether a polynomial is nonnegative over $\mathbb{R}^n$. However, testing whether a general polynomial is nonnegative is an NP-hard problem, even for polynomials of relatively low degrees or with a small number of variables. It has been proven that finding the global minimum of general even-degree polynomials with a degree of at least 4 is NP-hard [1; 5; 8; 133]. Also, Nesterov showed that minimizing a cubic polynomial over the Euclidean ball is NP-hard [143].

It is well known that determining whether a polynomial can be expressed as a SoS can be reformulated as a semidefinite programming (SDP) problem. Hilbert's 17th problem famously established that not all nonnegative polynomials admit an SoS representation. Hilbert also identified a few special classes in which nonnegativity and SoS are equivalent, such as univariate polynomials, quadratic polynomials, and bivariate quartic polynomials. In this chapter, we extend this equivalence to several new subclasses of multivariate quartically regularized polynomials and investigate the scope and limitations of exact SoS certification for these structured models. Namely, we consider the global optimization of multivariate symmetric cubic polynomials regularised by weighted quartic powers of the Euclidean norm (i.e., the AR3 subproblem). These special-structure polynomials arise as iterative subproblems in high-order tensor methods for nonconvex optimisation problems. We consider shifting these polynomials by their global optimum to make them nonnegative, and show that for sufficiently high regularization and under mild assumptions, these polynomials admit a SoS representation. We also identify several structured subclasses of quartically-regularized cubic polynomials for which global optimality of the model implies that its nonnegativity is certified by an SoS decomposition, for all values of the regularization parameter (such as quadratic-quartic polynomials and quartic polynomials containing a special cubic term that can be decomposed into a product of a quadratic norm and a linear form). We provide counterexamples using quartic separable norms that showcase the crucial role of the Euclidean norm in these representations. We illustrate how to use our SoS-based certificates for the Taylor subproblems inside high-order tensor methods for nonconvex optimization, with encouraging results.

Recall that in the case of $p = 3$, (ARp Subproblem) becomes a quartically regularized cubic polynomial in n variables,

$$\min_{s\in\mathbb{R}^n} m_3(s) = f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}T[s]^3 + \frac{\sigma}{4}\|s\|^4, \qquad \text{(Quartic Regularized Cubic Model)}$$

where $f_0 \in \mathbb{R}$, $g \in \mathbb{R}^n$, $H \in \mathbb{R}^{n\times n}$, and $T \in \mathbb{R}^{n\times n\times n}$ denote the constant, linear, quadratic, and cubic coefficients, respectively. The regularization term is a quartic-degree Euclidean norm, that is, $\frac{\sigma}{4}\|s\|^4 = \frac{\sigma}{4}(s_1^2 + \cdots + s_n^2)^2$. Since $\sigma > 0$, the model $m_3(s)$ is bounded below and the optimization problem is well defined, and we denote the global minimum value of m_3 as m_*, and the global minimum is attained (see Remark 5.3.1). In this work, we aim to address the following fundamental question

> *When* (Quartically Regularized Cubic Model) *is sufficiently regularized (i.e., when $\sigma > 0$ is large enough), is the shifted polynomial $m_3 - m_*$ a SoS?*

Once a polynomial can be written as a sum of squares, this not only certifies its global nonnegativity but also allows the associated global polynomial optimization problem to be reformulated as a semidefinite program (SDP) at the zeroth level of the Lasserre hierarchy [108; 158]. Such SDPs have polynomial size $\binom{n+4}{4}$ and can be solved in polynomial time using, for example, interior-point methods [145]. Moreover, the global minimum value can be approximated to arbitrary prescribed accuracy through the moment–SoS hierarchy. Under additional rank conditions on the optimal moment matrix, a global minimizer may also be extracted [112; 109].

In this chapter, we identify structural regimes in which quartically regularized polynomials admit exact SoS certificates. The contributions of the chapter are the following.

Regularization threshold for SoS: Under mild assumption, we establish an explicit bound on the regularization parameter σ such that, if σ exceeds this threshold, the shifted quartically regularized polynomial $(m_3 - m_*)$ is a SoS. When H is not positive semidefinite (i.e., when m_3 is not locally convex), the SoS property remains uncharacterized in the existing literature. To the best of our knowledge, this is the first work to establish SoS results for a generally nonconvex quartically regularized cubic polynomial. Building an understanding of how regularization strength governs convexification, SoS certification, and global optimality.

Special polynomial classes with equivalence between global optimality and SoS certificates: Hilbert's 17th problem famously established that not all nonnegative polynomials admit an SoS representation [95]. Hilbert also identified a few special classes in which nonnegativity and SoS are equivalent, such as *univariate polynomials*, *quadratic polynomials*, and *bivariate quartic polynomials*. In this chapter, we extend this equivalence to several new variants of multivariate quartically regularized polynomials. Specifically, (i) quadratic–quartic polynomials with any number of variables, and (ii) quartic polynomials containing a special cubic term that can be decomposed into a product of a quadratic norm and a linear form. These cases constitute a rare subclass of nonconvex polynomials that are SoS-certifiable, establishing the relationship between global optimality conditions and SoS representability.

Discussion of quartic norms and trust-region formulations: Lastly, we identify a subtle yet important distinction between different types of regularization norms and trust-region formulations. Several alternative regularization norms and seminorms have been proposed in the literature. For instance, Schnabel, Robert, and Chow [169] introduced a quartic seminorm formulation for unconstrained optimization of the form $\frac{1}{4}\sum_{j=1}^{k}\sigma_j(a_j^T s)^4$, where $\sigma_j > 0$ and $a_j \in \mathbb{R}^n$. The separable quartic norm (SQR) $\|s\|_4^4 = \sum_{i=1}^{n} s_i^4$. We use explicit examples to illustrate that when the regularization term is not the Euclidean quartic norm $\|s\|^4$ but the separable quartic norm $\|s\|_4^4$, even for arbitrarily large σ, the corresponding nonnegative separable quartic polynomial need not be SoS. We provide detailed analysis and examples on the choice of regularization norm or trust-region formulation that leads to SoS representability.

The chapter is organized as follows. Section 5.2 provides background and definitions on the SoS framework, the Lasserre hierarchy, and related results by Ahmadi, Lasserre, Nesterov, and Parrilo. Section 5.3 presents our main theoretical results, including explicit bounds on σ under which the quartically regularized cubic polynomial becomes SoS. We further discuss how this bound differs between locally convex cases ($H \succ 0$) and locally nonconvex cases. Section 5.4 analyzes several special classes of quartically regularized polynomials, identifying which forms are SoS and which are not, and highlighting the distinctions arising from different regularization norms. Section 5.5 provides detailed examples illustrating how various regularization norms lead to different SoS behaviors.

5.2 Preliminaries and Background on Nonnegative Polynomial and SoS

Let $\mathcal{P}_n^{\deg(h)}$ denote the class of real polynomials $h(s) : \mathbb{R}^n \to \mathbb{R}$ with n variables and degree $\deg(h)$ and $h(s) : \mathbb{R}^n \to \mathbb{R}$ be a (multivariate) real polynomial in variables $s = [s_1, \ldots, s_n]^T \in \mathbb{R}^n$. The degree of a polynomial, $\deg(h)$, is defined to be the highest degree of its component monomials[1]. If the global minimum exists, we denote it by $h^* := \min_{s \in \mathbb{R}^n} h(s)$ and s^* is a global minimizer. A homogeneous polynomial is a polynomial where all the monomials have the same degree. A polynomial $h(s)$ is said to be nonnegative if $h(s) \geq 0$ for all $s \in \mathbb{R}$.

Definition 5.2.1. *(SoS polynomial [1; 7; 102]) $h(s)$ is SoS if there exist polynomials $\tilde{h}_1, \ldots, \tilde{h}_r : \mathbb{R}^n \to \mathbb{R}$, for some $r \in \mathbb{N}$, such that $h(s) = \sum_{j=1}^r \tilde{h}_j(s)^2$ for all $s \in \mathbb{R}^n$. We refer to a problem as being* SoS-exact at level 0 *if $h(s) - h^*$ is SoS.*

Definition 5.2.1 implies that the degree of an SoS polynomial $h(s)$ has to be even, $\deg(h) = 2d$, and that the maximum degree of each $\tilde{h}_j$ is $\frac{\deg(h)}{2} = d$. Let $\mathbb{R}[s]^{n \times n}$ be the real vector space of $n \times n$ real polynomial matrices, where each entry of such a matrix is a polynomial with real coefficients.

Definition 5.2.2. *(SoS-convex, [7, Def. 2.4]) $h(s)$ is SoS-convex if its Hessian $\nabla^2 h(s)$ is an SoS-matrix. A symmetric polynomial matrix $\check{H}(s) \in \mathbb{R}[s]^{n \times n}$ is an SoS-matrix if there exists a polynomial matrix $\check{V}(s) \in \mathbb{R}[s]^{n_1 \times n}$ for some $n_1 \in \mathbb{N}$, such that $\check{H}(s) = \check{V}(s)^T \check{V}(s)$.*

Clearly, if $h(s) : \mathbb{R}^n \to \mathbb{R}$ is SoS-convex, then $h(s)$ is convex [7; 102]. Recently, Ahmadi, Chaudhry, and Zhang [1] proved that for a locally strongly convex quartically regularized polynomial ($H \succeq \delta I_n \succ 0$), there exists a sufficiently large regularization parameter σ such that the polynomial becomes SoS-convex (see Theorem 5.2.1). An SoS-convex polynomial can be globally minimized to arbitrary prescribed accuracy in polynomial time through its associated semidefinite programming formulation.

Theorem 5.2.1. *([1], Lemma 3 and Thm 3) If $H \succeq \delta I_n \succ 0$, then the following sum of squares programming*

$$\bar{\sigma} := \min_{\sigma} \ \sigma \qquad s.t. \qquad \text{(Quartically Regularized Cubic Model) } \textit{is SoS-convex}$$

is feasible. Moreover, this program can be reformulated as an SDP of size that is polynomial in n.

By choosing σ under this setup, the resulting polynomial is strongly convex, its global minimizer exists and is unique. Using the SoS-convexity, we deduce that the unique global minimizer can be obtained by solving an SDP whose size is polynomial in $N = \binom{n+2d}{2d}$. Therefore, for locally convex regularized polynomials, both determining such a regularization parameter σ and globally minimizing the resulting SoS-convex polynomial reduce to solving semidefinite programs in polynomial time. The following lemma from [93] connects nonnegative SoS-convex polynomials to SoS polynomials, and therefore ensures that a sufficiently regularized, nonnegative, locally strongly convex polynomial is SoS.

Lemma 5.2.1 (SoS-convexity and non-negativity imply SoS; Lemma 8 in [93])**.** *Let $h : \mathbb{R}^n \to \mathbb{R}$ be an SoS-convex polynomial. Suppose that h attains its global minimum at s^*, and define $q(s) := h(s) - h(s^*)$. Then $q(s) \geq 0$ for all $s \in \mathbb{R}^n$, with $q(s^*) = 0, \nabla q(s^*) = 0$. Since q is also SoS-convex, Lemma 8 in [93] implies that q is SoS. Hence $h(s)$ is SoS.*

While convexity often leads to tractable global optimization [155; 7], local nonconvexity makes the problem markedly more difficult. In 1992, the question of determining whether a polynomial defines a convex function was posed as one of the seven major open problems in the complexity theory of numerical optimization [153]. Ahmadi, Olshevsky, and Tsitsiklis later proved that this decision problem is strongly NP-hard even for quartic polynomials [5]. A natural and fundamental question, first posed and (partially) answered by Hilbert, is to determine for which dimensions n and degrees $\deg(h)$, nonnegative polynomials or forms can always be expressed as sums of squares.

[1]The degree of a monomial s^α where $\alpha = [\alpha_1, \ldots, \alpha^n]$ is equal to $\alpha_1 + \ldots + \alpha_n$

Theorem 5.2.2 (Hilbert, [95]). *For the space of real polynomials $\mathcal{P}_n^{\deg(h)}$, nonnegativity and SoS coincide if and only if one of the following holds:*

1. $n = 1$ *(univariate polynomials of any degree);*
2. $\deg(h) = 2$ *(quadratic polynomials in any number of variables);*
3. $n = 2$ *and* $\deg(h) = 4$ *(bivariate quartic polynomials).*

For all other combinations of dimension and degree, Hilbert showed that nonnegativity does not imply a SoS representation. However, within the class of quartic polynomials, the SoS behavior of quartically regularized models is largely unexplored. Explicit examples of nonnegative but non-SoS polynomials appeared in the 1960s, most notably the Motzkin polynomial $M(s_1, s_2) = s_1^4 s_2^2 + s_1^2 s_2^4 - 3s_1^2 s_2^2 s_3^2 + s_3^6$, a nonnegative trivariate polynomial of degree six [131]. A more recent example appears in [3, Thm. 3.3], a nonnegative trivariate quartic polynomial which is not SoS,

$$\tilde{m}_{\mathcal{A}}(s) = \frac{9}{4} + 2\|s\|^2 + 8(s_1 s_2 + s_1 s_3 + s_2 s_3) + (s_1^4 + s_2^4 + s_3^4). \tag{5.1}$$

Using this example, [3] proved that testing the nonnegativity of separable quartic–quadratic polynomials is NP-hard. The distinction between (5.1) and (Quartically Regularized Cubic Model): the former uses a separable quartic norm regularization, whereas the latter employs a Euclidean quartic norm. This leads to subtle but important differences in the characterization of their SoS representations. Also, Nesterov [143] proved that even very structured nonconvex polynomial optimization problems are NP-hard to solve globally. In particular, he showed that the following problems are NP-hard.

Theorem 5.2.3. *([143, Thm 4], [120]) The maximization of homogeneous cubic or quartic form over the Euclidean ball, and the optimization of a quartic homogeneous polynomial under quadratic constraints are NP-hard*[2]*.*

5.2.1 Connection to SDP and Matrix Generalizations

As mentioned earlier, the computational advantage of sums of squares is the connection to semidefinite programming. It is well known that a polynomial is SoS if and only if it admits a positive semidefinite Gram matrix representation.

Theorem 5.2.4. *(Parrilo [156]) Let $h \in \mathcal{P}_n^{2d}$ and $\phi_d(s)$ be the vector of all monomials of degree at most d. A polynomial $h(s)$ is SoS if and only if there exists a symmetric matrix Q such that (i) $h(s) = \phi_d(s)^T Q \phi_d(s)$ for all $s \in \mathbb{R}^n$, (ii) $Q \succeq 0$. Note that Q has size $\binom{n+2d}{2d}$.*

Example 5.2.1. *For quartic polynomials ($d = 2$), ϕ_2 is a list of monomial basis with highest degree 2,*

$$\phi_2(v) := [1, v_1, \ldots, v_n, v_1^2, \ldots, v_n^2, v_1 v_2, \ldots, v_1 v_n, v_2 v_3, \ldots, v_2 v_n, \ldots, v_{n-1} v_n]^T \in \mathbb{R}^{\frac{1}{2}(n+2)(n+1)}. \tag{5.2}$$

In this chapter, we use simplified notation. Let $\omega \in \mathbb{R}^{\frac{1}{2}(n+2)(n+1)}$

$$\omega := [1, v, u, z]^T = [1, \underbrace{v_1, \ldots, v_n}_{:=v \in \mathbb{R}^n}, \underbrace{u_1, \ldots, u_n}_{:=u \in \mathbb{R}^n}, \underbrace{z_{1,2}, \ldots, z_{n-1,n}}_{:=z \in \mathbb{R}^{\frac{n(n-1)}{2}}}]^T. \tag{5.3}$$

where $v = [v_1, \ldots, v_n]^T$, $u = [v_1^2, \ldots, v_n^2]$, $z = \{z_{ij}\}_{1 \le i < j \le n} = [v_1 v_2, \ldots, v_1 v_n, v_2 v_3, \ldots, v_2 v_n, \ldots, v_{n-1} v_n]$.

If the nonnegative polynomial $h(s) - h^*$ admits an SoS representation, then—by the Lasserre moment–SoS hierarchy [1; 110; 158]—two semidefinite programs coexist in duality. The SoS primal formulation [110] takes the form of a single convex semidefinite program (SDP) with dimension $N = \binom{n+2d}{2d}$.

$$\min_{h^*,\, Q \succeq 0} \; h^* \quad \text{s.t.} \quad h(s) - h^* = \phi_d(s)^T Q \, \phi_d(s), \tag{5.4}$$

[2]The problem formulations are given in (5.46)–(5.48).

where $\phi_d(s)$ denotes the vector of all monomials in s of degree $\leq d$, and $Q \succeq 0$ is the associated Gram matrix. The global minimum h^* is computable in polynomial time in the problem size. If the optimal moment matrix satisfies the rank condition, then s^* can be recovered from the moment dual SDP [110; 158; 148]. Exactness of the SoS relaxation guarantees recovery of the global optimal value, whereas extraction of a global minimizer additionally requires an appropriate rank or flat-extension condition on the associated optimal moment matrix.

As the resulting SDPs have size $N = \binom{n+2d}{2d}$, they could suffer from the curse of dimensionality, making SoS programming impractical for larger-scale problems ($n, d \geq 10$) without exploitable structure. When sparsity is present, sparsity-adapted SDP hierarchies can be used [39; 111; 128; 177]. Scalable variants such as DSoS/SDSoS and BSoS [4; 113; 177; 182; 189] further reduce SDP size by leveraging structural properties.

Lastly, we remark that when the nonnegative polynomial $h(s) - h^*$ is not SoS, Lasserre [108] showed that polynomial optimization problems can be approximated through a hierarchy of increasingly tight SDP relaxations. Under the standard Archimedean assumption, this hierarchy converges asymptotically to the global optimal value. Moreover, Nie [149] proved finite convergence when the constraint qualification, strict complementarity, and second-order sufficiency conditions hold at every global minimizer. Scheiderer [167] proved that there is no uniform bound on the number of relaxation layers for finite convergence.

Recently, Ahmadi and Zhang [8] established polynomial-time checkable necessary and sufficient conditions for local minimality of cubic polynomials. Building on this, Silina and Zhang [172] proposed an unregularized third-order Newton method in which each iteration computes a local minimizer of the third-order Taylor model via an SDP in polynomial size. This unregularized approach coincides with the ARp model when the regularization parameter is set to zero. Recently, [38] gives a globally convergent framework, Adaptive Levenberg-Marquardt Third-Order Newton Method (ALMTON). A comprehensive overview of methods involving third- and higher-order derivatives is provided in [44, Ch. 4].

5.3 Nonnegative Quartically Regularized Polynomial is SoS for Large Regularization

This section presents the main theoretical proof showing that the quartically regularized polynomial is SoS for sufficiently large σ. We begin by outlining the structure of the proof and the organisation of this section. First, we express the quartic polynomial

$$q(v) := m_3(s^* + v) - m_3(s^*)$$

where m_3 is defined in (1.2) via its fourth-order Taylor expansion at s^* (see Lemma 5.3.1). The key idea is that, under mild assumption ($\|g\| \neq 0$), the regularization term $\sigma\|s^*\|^2 I_n$ becomes dominant as $\sigma \to \infty$, eventually guaranteeing that $B(s^*) \succeq 0$. To make this precise, in Section 5.3.2 we derive upper and lower bounds on $\|s^*\|$ and show that $\|s^*\|$ scales as $O(\sigma^{-1/3})$ as $\sigma \to \infty$, implying that $\sigma\|s^*\|^2 \to \infty$ as $\sigma \to \infty$. Finally, in Section 5.3.3 (see Theorem 5.3.3), when $\|g\| \neq 0$, we use these bounds to derive an explicit condition on σ that ensures $B(s^*) \succeq 0$, and consequently guarantees that the quartically regularized polynomial is SoS. When $\|g\| \neq 0$, we are able to identify several subcases in which $m_3(s) - m_3(s^*)$ remains SoS Remark 5.3.2. Determining, generally, whether $m_3(s) - m_3(s^*)$ is SoS when $g = 0$ is left for future research.

Theorem 5.3.1 and Theorem 5.3.3 are the two central results of this section. A detailed case-by-case analysis for the locally convex and locally nonconvex settings is provided in Section 5.3.3.1 and Section 5.3.3.2. These sections build the connections between SoS representations, convexification, SoS-convexity, and global optimality, and they characterize the resulting behaviour of the global minimizer in each regime.

5.3.1 SoS Representation of Shifted AR3 Subproblem and Conditions for the Global Minimum

According to global optimality conditions of m_3 in Chapter 4, we have the following Lemma to express $q(v)$

Lemma 5.3.1. *(Adapted from Lemma 4.2.2) Let $q(v) := m_3(s^* + v) - m_3(s^*)$ where m_3 is defined in (1.2). Then the fourth-order Taylor expansion of $q(v)$ at the point s^* is exact and yields, for any vector $v \in \mathbb{R}^n$,*

$$\begin{aligned} q(v) &= \nabla m_3(s^*)^T v + \frac{1}{2}\,\mathcal{G}(s^*)[v]^2 + \frac{1}{6}\,T[v]^3 + \frac{\sigma}{4}\big(\|s^* + v\|^2 - \|s^*\|^2\big)^2 \\ &= \nabla m_3(s^*)^T v + \frac{1}{2}\,\nabla^2 m_3(s^*)[v]^2 + \frac{1}{6}\,T[v]^3 + \frac{\sigma}{4}\|v\|^4 + \sigma\, s^{*T} v\, \|v\|^2, \end{aligned} \tag{5.5}$$

where $\mathcal{G}(s) := H + T[s] + \sigma\|s\|^2 I_n$. Note that the first- and second-order derivatives of m_3 are

$$\nabla m_3(s) = g + Hs + \frac{1}{2}\,T[s]^2 + \sigma\|s\|^2 s, \qquad \nabla^2 m_3(s) = H + T[s] + \sigma\big(2ss^T + \|s\|^2 I_n\big).$$

where $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix.

Next, we present a technical lemma that rewrites the homogeneous cubic term $T[v]^3$, as a sum of linear polynomials multiplied by quadratic polynomials.

Lemma 5.3.2. *Any homogeneous cubic polynomial, $c_3 : \mathbb{R}^n \to \mathbb{R}$ can be expressed as*

$$c_3(v) = \left(\sum_{i=1}^{n} u_i\right)\big(t^T v\big) + \left[\sum_{1\le i<j\le n} z_{ij}\big(v^T b^{(ij)}\big)\right],$$

where u_i and z_{ij} are quadratic terms, t and $b^{(ij)}$ are vectors in $\mathbb{R}^n$ based on the coefficients of $c_3(v)$.

Proof. For a homogeneous cubic polynomial, $c_3(v) = \frac{1}{6}T[v]^3$, we can write it in expanded form as $c_3(v) := \sum_{1\le i\le j\le k\le n} t_{ijk} v_i v_j v_k$. Note that coefficients in the expanded form $\sum_{1\le i\le j\le k\le n} t_{ijk} v_i v_j v_k$ correspond to the tensor entries $\{T\}_{ijk}$ in tensor form $\frac{1}{6}T[v]^3$ as follows, $t_{ijk} = T_{ijk}$ for all distinct ijk, $t_{iij} = \frac{1}{2}T_{iij}$ and $t_{iii} = \frac{1}{6}T_{iii}$. We can further express $c_3(v)$ as a sum of linear polynomials times quadratic polynomials

$$\begin{aligned} c_3(v) := \sum_{i,j,k=1, i\le j\le k}^{n} t_{ijk} v_i v_j v_k &= \left(\sum_{i=1}^{n} v_i^2\right)\big(t^T v\big) + \left[\sum_{1\le i<j\le n} v_i v_j \big(v^T b^{(ij)}\big)\right] \\ &= \left(\sum_{i=1}^{n} u_i\right)\big(t^T v\big) + \left[\sum_{1\le i<j\le n} z_{ij}\big(v^T b^{(ij)}\big)\right] \end{aligned} \tag{5.6}$$

where $t = \{t_{iii}\}_{1\le i\le n} \in \mathbb{R}^n$. For $1 \le i < j \le n$, $b^{(ij)} = [t_{1ij}, \ldots, t^*_{iij} \ldots, t^*_{ijj} \ldots, t_{ijn}]^T \in \mathbb{R}^n$ with $t^*_{iij} = t_{iij} - t_{jjj}$ and $t^*_{ijj} := t_{ijj} - t_{iii}$. □

Theorem 5.3.1 provides a sufficient condition under which $q(v) := m_3(s^* + v) - m_3(s^*)$ can be expressed as a SoS polynomial, and it also gives an explicit SoS representation when the condition is satisfied.

Theorem 5.3.1. *Assume $\sigma > \nu > 0$. If s^* satisfies $\nabla m_3(s^*) = 0$ and*

$$\begin{aligned} B(s^*) = H + T[s^*] + \sigma\|s^*\|^2 I_n - \nu\big(\|s^*\|^2 I_n - s^* s^{*T}\big) - \frac{2}{\sigma-\nu}\,tt^T - 2(s^* t^T + t s^{*T}) \\ - \sum_{1\le i<j\le n}\left(\tilde{s}^{(ij)} b^{(ij)T} + b^{(ij)}\tilde{s}^{(ij)T} + \frac{1}{\nu} b^{(ij)} b^{(ij)T}\right) \succeq 0. \end{aligned} \tag{5.7}$$

where $\tilde{s}^{(ij)} = [0, \ldots s^*_j, \ldots, s^*_i, \ldots 0] \in \mathbb{R}^n$ has s^*_j on ith entry and s^*_i on jth entry and all other zeros. $b^{(ij)}$ and t depends on tensor coefficients and are defined in (5.6). Then, $q(v) := m_3(s^* + v) - m_3(s^*)$ is SoS in v, and consequently, s^* is a global minimizer of m_3.

Proof. Since $\nabla m_3(s^*) = 0$, it follows from Lemma 5.3.1 that

$$2q(v) = \nabla^2 m_3(s^*)[v]^2 + \frac{1}{3}T[v]^3 + 2\sigma\|v\|^2 v^T s^* + \frac{\sigma}{2}\|v\|^4. \tag{5.8}$$

We express the terms in (5.8) using $\omega = [1, v, u, z]^T$, where $u_i = v_i^2$ and $z_{ij} = v_i v_j$.

Fix $\nu \in (0, \sigma)$. The regularization-induced cubic term can be decomposed as

$$\begin{aligned}\sigma v^T s^* \|v\|^2 &= (\sigma-\nu)v^T s^* \|v\|^2 + \nu \sum_{i=1}^n s_i^* v_i^3 + \nu \sum_{1\le i<j\le n} v_i v_j (v_i s_j^* + v_j s_i^*) \\ &= (\sigma-\nu)\left(\sum_{i=1}^n u_i\right)(v^T s^*) + \nu \sum_{i=1}^n s_i^* v_i u_i + \nu \sum_{1\le i<j\le n} z_{ij}\big(v^T \tilde{s}^{(ij)}\big). \end{aligned} \tag{5.9}$$

Moreover, the homogeneous quartic term satisfies

$$\sigma\|v\|^4 = \sigma\left(\sum_{i=1}^n v_i^2\right)^2 = \sigma\sum_{i=1}^n u_i^2 + 2(\sigma-\nu)\sum_{1\le i<j\le n} u_i u_j + 2\nu \sum_{1\le i<j\le n} z_{ij}^2. \tag{5.10}$$

Combining (5.6), (5.8), (5.9), and (5.10), we obtain

$$\begin{aligned} 2q(v) = \nabla^2 m_3(s^*)[v]^2 &\\ &+ \underbrace{\frac{\sigma-\nu}{2}\sum_{i=1}^n u_i^2 + (\sigma-\nu)\sum_{1\le i<j\le n} u_i u_j + 2\left(\sum_{i=1}^n u_i\right)(t^T v) + 2(\sigma-\nu)\left(\sum_{i=1}^n u_i\right)(v^T s^*)}_{\mathcal{I}_1} \\ &+ \underbrace{\nu \sum_{1\le i<j\le n} z_{ij}^2 + 2\sum_{1\le i<j\le n} z_{ij}(v^T b^{(ij)}) + 2\nu \sum_{1\le i<j\le n} z_{ij}(v^T \tilde{s}^{(ij)})}_{\mathcal{I}_2} \\ &+ \underbrace{2\nu\sum_{i=1}^n s_i^* v_i u_i + \frac{\nu}{2}\sum_{i=1}^n u_i^2}_{\mathcal{I}_3}. \end{aligned} \tag{5.11}$$

Here, the quartic terms arise from (5.10), while the terms involving s^* arise from (5.9). The remaining cubic terms arise from the decomposition of $T[v]^3$ in (5.6). Completing squares gives

$$\mathcal{I}_1 = \underbrace{\frac{\sigma-\nu}{2}\left[\sum_{i=1}^n u_i + 2v^T s^* + \frac{2t^T v}{\sigma-\nu}\right]^2}_{\mathcal{SI}_1} - \underbrace{\left[\frac{2}{\sigma-\nu}(t^T v)^2 + 4(v^T s^*)(t^T v)\right]}_{\mathcal{BI}_1} - 2(\sigma-\nu)(v^T s^*)^2,$$

$$\mathcal{I}_2 = \underbrace{\nu\sum_{1\le i<j\le n}\left[z_{ij} + v^T\tilde{s}^{(ij)} + \frac{v^T b^{(ij)}}{\nu}\right]^2}_{\mathcal{SI}_2} - \underbrace{\sum_{1\le i<j\le n}\left[2(v^T\tilde{s}^{(ij)})(v^T b^{(ij)}) + \frac{(v^T b^{(ij)})^2}{\nu}\right]}_{\mathcal{BI}_2} - \nu\sum_{1\le i<j\le n}(v^T\tilde{s}^{(ij)})^2,$$

$$\mathcal{I}_3 = \underbrace{\frac{\nu}{2}\sum_{i=1}^n (u_i + 2s_i^* v_i)^2}_{\mathcal{SI}_3} - 2\nu\sum_{i=1}^n (s_i^* v_i)^2.$$

We next collect the final terms in the three expressions above:

$$2(\sigma-\nu)(v^T s^*)^2 + \nu\sum_{1\le i<j\le n}(v_i s_j^* + v_j s_i^*)^2 + 2\nu\sum_{i=1}^n (s_i^* v_i)^2 = 2\sigma(v^T s^*)^2 + \underbrace{\nu\sum_{1\le i<j\le n}(v_i s_j^* - v_j s_i^*)^2}_{\mathcal{BI}_3}. \tag{5.12}$$

Furthermore,

$$\begin{aligned}
\mathcal{BI}_3 &= \nu v^T\big(\|s^*\|^2 I_n - s^* s^{*T}\big)v,\\
\mathcal{BI}_1 &= v^T\left[\frac{2}{\sigma-\nu}tt^T + 2\big(s^*t^T + ts^{*T}\big)\right]v,\\
\mathcal{BI}_2 &= v^T\left[\sum_{1\le i<j\le n}\left(\tilde{s}^{(ij)}b^{(ij)T} + b^{(ij)}\tilde{s}^{(ij)T} + \frac{1}{\nu}b^{(ij)}b^{(ij)T}\right)\right]v.
\end{aligned}$$

Substituting these identities into (5.11) yields

$$\begin{aligned}
2q(v) &= \nabla^2 m_3(s^*)[v]^2 - 2\sigma(v^T s^*)^2 - \mathcal{BI}_1 - \mathcal{BI}_2 - \mathcal{BI}_3 + \mathcal{SI}_1 + \mathcal{SI}_2 + \mathcal{SI}_3\\
&= B(s^*)[v]^2 + \mathcal{SI}_1 + \mathcal{SI}_2 + \mathcal{SI}_3. \qquad (5.13)
\end{aligned}$$

Indeed, using

$$\nabla^2 m_3(s^*) = H + T[s^*] + \sigma\|s^*\|^2 I_n + 2\sigma s^* s^{*T},$$

the quadratic part in the first line of (5.13) is precisely $B(s^*)[v]^2$, where $B(s^*)$ is defined in (5.7).

By assumption, $B(s^*) \succeq 0$, and hence the quadratic form $B(s^*)[v]^2$ is a sum of squares of linear functions of v. Moreover, each of $\mathcal{SI}_1$, $\mathcal{SI}_2$, and $\mathcal{SI}_3$ is explicitly a sum of squares. Therefore, (5.13) shows that $q(v)$ is SoS. Consequently,

$$m_3(s^* + v) - m_3(s^*) = q(v) \ge 0 \qquad \text{for all } v \in \mathbb{R}^n,$$

and hence s^* is a global minimizer of m_3. □

Corollary 5.3.1. *($\mathcal{Q}$ matrix formulation) Let $\mathbf{1}^T = [1,\dots,1] \in \mathbb{R}^n$. We construct a symmetric matrix of dimension $\binom{n}{2} \times \binom{n}{2}$, such that*

$$\mathcal{Q} = \begin{bmatrix} 0 & 0 & 0 & 0\\ 0 & \nabla^2 m_3(s^*) & D(s) & C(s)\\ 0 & D(s)^T & U & 0\\ 0 & C(s)^T & 0 & \nu I_N \end{bmatrix} \qquad (5.14)$$

- *The u-u block has size $n \times n$ and $U = \left(\frac{\sigma}{2} - \frac{\nu}{2}\right)\mathbf{1}\mathbf{1}^T + \frac{\nu}{2}I_n$ which has $\frac{\sigma}{2}$ on the diagonal and $\frac{\sigma-\nu}{2}$ off-diagonal.*
- *The z-z block has size $N \times N$ is an identity matrix νI_N.*
- *The v-u block has size $n \times n$, $D(s) = \big[(\sigma-\nu)s^* + t\big]\mathbf{1}^T + \nu\,\mathrm{diag}(s^*)$.*
- *The v-z block has size $n \times N$ with the ij-th column as $\nu\tilde{s}^{(ij)} + b^{(ij)}$.*

Then, if the conditions in Theorem 5.3.1 are satisfied, then $\mathcal{Q} \succeq 0$.

Proof. Using $\omega := [1, v, u, z]^T$ as defined in (5.3) we analyze $\omega^T\mathcal{Q}\omega$ term by term. Using $\mathbf{1}^T u = \sum_{i=1}^n u_i$, the u-u block yields

$$u^T U u = \frac{\sigma}{2}\sum_{i=1}^n u_i^2 + (\sigma-\nu)\sum_{1\le i<j\le n} u_i u_j.$$

The z-z block gives $\nu\sum_{1\le i<j\le n} z_{ij}^2$. The v-u block gives

$$u^T D(s) v = 2\left(\sum_{i=1}^n u_i\right)\big(t^T v\big) + 2(\sigma-\nu)\left(\sum_{i=1}^n u_i\right)\big(v^T s^*\big) + 2\nu\sum_{i=1}^n s_i^* v_i u_i.$$

The v-z block gives

$$u^T C(s) z = \nu\sum_{1\le i<j\le n} z_{ij}\big(v^T\tilde{s}^{(ij)}\big) + \sum_{1\le i<j\le n} z_{ij}\big(v^T b^{(ij)}\big).$$

By adding the terms from these blocks and comparing $\omega^T \mathcal{Q} \omega$ with (5.11), we obtain $\omega^T \mathcal{Q} \omega = 2q(v)$. If the conditions of Theorem 5.3.1 are satisfied, then $q(v)$ is SoS. By Theorem 5.2.4, we conclude that $\mathcal{Q} \succeq 0$. □

Example 5.3.1. *We write the full matrix for the case of $n = 2$:*

$$2q(v) = \begin{bmatrix} 1 \\ v_1 \\ v_2 \\ u_1 \\ u_2 \\ z_{12} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \nabla^2 m_3(s^*) & & \sigma s_1 + t_{111} & (\sigma-\nu)s_1 + t_{111} & \nu s_2 + t^*_{112} \\ 0 & & & (\sigma-\nu)s_2 + t_{222} & \sigma s_2 + t_{222} & \nu s_1 + t^*_{122} \\ 0 & * & * & \frac{\sigma}{2} & \frac{\sigma}{2} - \frac{\nu}{2} & 0 \\ 0 & * & * & \frac{\sigma}{2} - \frac{\nu}{2} & \frac{\sigma}{2} & 0 \\ 0 & * & * & 0 & 0 & \nu \end{bmatrix} \begin{bmatrix} 1 \\ v_1 \\ v_2 \\ u_1 \\ u_2 \\ z_{12} \end{bmatrix}.$$

The symbol "$*$" denotes symmetric entries in the corresponding positions. By examining the quadratic form $\mathcal{Q}[\tilde{\omega}]^2$, with $\tilde{\omega} \in \mathbb{R}^{\frac{1}{2}(n+2)(n+1)}$, we find that $\mathcal{Q}[\tilde{\omega}]^2 \geq 0$ holds whenever $\nabla m_3(s^*) = 0$ and $B(s^*) \succeq 0$. It follows that $\mathcal{Q} \succeq 0$, and consequently the corresponding polynomial $q(v)$ is SoS. Lastly, we note that the expression of $\mathcal{Q}$ is not unique. Let ν, δ, and β be arbitrary constants representing the degrees of freedom in choosing the matrix $\mathcal{Q}$.

$$\mathcal{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \nabla^2 m_3(s^*) & & \sigma s_1 + t_{111} & (\sigma-\nu)s_1 + t_{122} + \beta & \nu s_2 + t_{112} + \delta \\ 0 & & & (\sigma-\nu)s_2 - \delta & \sigma s_2 + t_{222} & \nu s_1 - \beta \\ 0 & * & * & \frac{\sigma}{2} & \frac{\sigma}{2} - \frac{\nu}{2} & 0 \\ 0 & * & * & \frac{\sigma}{2} - \frac{\nu}{2} & \frac{\sigma}{2} & 0 \\ 0 & * & * & 0 & 0 & \nu \end{bmatrix}.$$

5.3.2 Bound on Global Minimizer

To effectively certify the positive semidefiniteness of $B(s^*)$, we require upper and lower bounds on s^* expressed in terms of the coefficients of m_3. Note that we define the vector norm, the matrix norm and tensor norm as such

$$\|H\| := \max_{s \in \mathbb{R}^n, \|s\|=1} |H[s]^2| = \max_i \{|\lambda_i|\}, \qquad \lambda_* := \max\{-\lambda_{\min}[H], 0\} \tag{5.15}$$

$$\|T\| := \max_{s \in \mathbb{R}^n, \|s\|=1} |T[s]^3|, \qquad \|T\|_F := \sum_{i,j,k} T^2_{ijk}, \qquad \|g\| = (\sum_{i=1}^n g_i^2)^{1/2}. \tag{5.16}$$

Clearly, $\|T\|_F \geq \|T\|$ and more details are provided in Section A.3.1.2.

Theorem 5.3.2. ***(Bound for s^*)*** *Assume that $\|g\| \neq 0$, let s^* satisfies $\nabla m_3(s^*) = 0$ and $m_3(s^*) \leq m_3(0)$, and σ satisfies*

$$\sigma \geq \max\left\{ \frac{9\|H\|^3}{\|g\|^2}, \left(\frac{3\|T\|^3}{8\|g\|}\right)^{1/2} \right\}, \tag{5.17}$$

then,

$$\left(\frac{\|g\|}{3\sigma}\right)^{1/3} \leq \|s^*\| \leq 2\left(\frac{\|g\|}{\sigma}\right)^{1/3}. \tag{5.18}$$

Proof. **To prove the upper bound for** $\|s^*\|$, we rewrite m_3 at s^* as

$$\begin{aligned} m_3(s^*) - m_3(0) &= \left(g^T s + \frac{\sigma}{8}\|s^*\|^4\right) + \left(H[s^*]^2 + \frac{\sigma}{16}\|s^*\|^4\right) + \left(-\frac{1}{6}T[s]^3 + \frac{\sigma}{16}\|s^*\|^4\right) \\ &\geq \left(-\|g\| + \frac{\sigma}{8}\|s^*\|^3\right)\|s^*\| + \left(-\frac{\lambda_*}{2} + \frac{\sigma}{16}\|s^*\|^2\right)\|s^*\|^2 + \left(-\frac{\|T\|}{6} + \frac{\sigma}{16}\|s^*\|\right)\|s^*\|^3 \end{aligned}$$

where the last inequality comes from $g^T s \leq \|g\|\|s\|$, $H[s^*]^2 \geq -\lambda_*\|s^*\|$ and $T[s^*]^3 \geq \|T\|\|s^*\|^3$. Assume that (5.17) holds and assume for contradiction the global minimum of $m_3(s)$ happens at $\|s^*\| > r_c := \left(\frac{8\|g\|}{\sigma}\right)^{1/3}$. Then,

$$m_3(s^*) - m_3(0) > \underbrace{\left[-\|g\| + \frac{\sigma}{8}r_c^3\right]}_{=0}\|s^*\| + \underbrace{\left[-\frac{\lambda_*}{2} + \frac{\sigma}{16}r_c^2\right]}_{\geq 0}\|s^*\|^2 + \underbrace{\left[-\frac{\|T\|}{6} + \frac{\sigma}{16}r_c\right]}_{\geq 0}\|s^*\|^3 \geq 0. \quad (5.19)$$

However, we have $m_3(s^*) \leq m_3(0)$, we arrives at a contradiction. Note that in (5.19), the first term is zero using $r_c = \left(\frac{8\|g\|}{\sigma}\right)^{1/3}$. The second and third term is positive using $\sigma \geq \frac{9\|H\|^3}{\|g\|^2}$ and $\sigma \geq \left(\frac{3\|T\|^3}{8\|g\|}\right)^{1/2}$ from condition (5.17),

$$\begin{aligned}
\frac{\sigma}{16}r_c^2 &= \left(\frac{1}{4}\|g\|^{2/3}\right)\sigma^{1/3} \underset{(5.17)}{\geq} \left(\frac{1}{4}\|g\|^{2/3}\right)9^{1/3}\|H\|\|g\|^{-2/3} > \frac{\|H\|}{2} > \frac{\lambda_*}{2}, \\
\frac{\sigma}{16}r_c &= \left(\frac{1}{8}\|g\|^{1/3}\right)\sigma^{2/3} \underset{(5.17)}{\geq} \left(\frac{1}{8}\|g\|^{1/3}\right)\left(\frac{8}{3}\right)^{1/3}\|T\|\|g\|^{-1/3} > \frac{\|T\|}{6}.
\end{aligned}$$

To prove the lower bound for $\|s^*\|$,

$$\begin{aligned}
\|g\| &= \|\nabla m_3(0)\| = \|\nabla m_3(s^*) - \nabla m_3(0)\| \leq \left\|H[s^*] + \frac{1}{2}T[s^*]^2 + \sigma\|s^*\|^2 s^*\right\| \\
&\leq \|H\|\|s^*\| + \frac{1}{2}\|T\|\|s^*\|^2, +\sigma\|s^*\|^3. \quad (5.20)
\end{aligned}$$

Therefore, we have

$$\frac{\|g\|}{3} \leq \|H\|\|s^*\|, \quad \text{or} \quad \frac{\|g\|}{3} \leq \frac{1}{2}\|T\|\|s^*\|^2, \quad \text{or} \quad \frac{\|g\|}{3} \leq \sigma\|s^*\|^3.$$

This gives

$$\|s^*\| \geq \min\left\{\frac{\|g\|}{3\|H\|}, \left(\frac{2\|g\|}{3\|T\|}\right)^{1/2}, \left(\frac{\|g\|}{3\sigma}\right)^{1/3}\right\}. \quad (5.21)$$

From (5.17), we have

$$\begin{aligned}
\sigma \geq \frac{9\|H\|^3}{\|g\|^2}, \qquad &\Rightarrow \qquad \left(\frac{\|g\|}{3\sigma}\right)^{1/3} \leq \frac{\|g\|}{3\|H\|} \\
\sigma \geq \left(\frac{3\|T\|^3}{8\|g\|}\right)^{1/2} \qquad &\Rightarrow \qquad \left(\frac{\|g\|}{3\sigma}\right)^{1/3} \leq \left(\frac{2\|g\|}{3\|T\|}\right)^{1/2}.
\end{aligned}$$

Therefore, under (5.17), (5.21) simplifies to $\|s^*\| \geq \left(\frac{\|g\|}{3\sigma}\right)^{1/3}$. □

Remark 5.3.1. *(Global Minimum Attained) For any $\sigma > 0$, the model $m_3(s)$ is coercive and $\|s\| \to \infty$ gives $m_3(s) \to +\infty$. The argument used to derive an upper bound on $\|s^*\|$ implies that $m_3(s) \leq m_3(0)$ can only occur when $\|s\| \geq r_c$. Consequently, the constrained minimization problem $s_r^* := \arg\min_{\|s\| \leq r_c} m_3(s)$ admits a global minimizer. Since any global minimizer s^* of m_3 must satisfy $m_3(s^*) \leq m_3(0)$, we conclude that $s^* = s_r^*$, and hence a global minimizer of m_3 is attained for all $\sigma > 0$.*

5.3.3 Behaviour of $B(s^*)$ when σ large

We are now ready to prove the main theoretical result of the chapter. In Theorem 5.3.3, we show that when σ is sufficiently large, the matrix $B(s^*)$ becomes positive semidefinite, and consequently $m_3(s^*+v)-m_3(s^*)$ is a SoS polynomial. Moreover, we provide an explicit bound on σ, expressed solely in terms of the coefficients of m_3, that guarantees the SoS representation of $m_3(s^*+v) - m_3(s^*)$. In Section 5.3.3.1 and Section 5.3.3.2, we provide a case-by-case analysis showing how the bound simplifies for locally convex and nonconvex m_3.

Theorem 5.3.3. *($B(s^*)$ is PSD for large σ) Assume that $\|g\| \neq 0$, let s^* satisfies $\nabla m_3(s^*) = 0$ and $m_3(s^*) \leq m_3(0)$, and σ satisfies*

$$\sigma \geq \max\left\{\frac{9\|H\|^3}{\|g\|^2}, \left(\frac{3\|T\|^3}{8\|g\|}\right)^{1/2}, \left(\frac{3\cdot 32^3\|T\|^3}{\|g\|}\right)^{1/2}, \frac{3^2\cdot 6^3\lambda_*^3}{\|g\|^2}\right\} = \max\left\{\frac{9\cdot 6^3\|H\|^3}{\|g\|^2}, \left(\frac{3\cdot 32^3\|T\|^3}{\|g\|}\right)^{1/2}\right\} \quad (5.22)$$

where the norms and λ_* are defined in (5.15) – (5.16). Then $B(s^*)$ defined in (5.7) with $\nu = \frac{\sigma}{2}$ is positive semi-definite. Moreover, then, $m_3(s^*+v) - m_3(s^*)$ is SoS and consequently, s^* is a global minimizer.

Proof. When $\nu = \frac{\sigma}{2}$ in (5.7), we have

$$B(s^*) = H + T[s^*] + \frac{\sigma}{2}\|s^*\|^2 I_n + \frac{\sigma}{2}s^*s^{*T} - \frac{4tt^T}{\sigma} - 2\left(s^*t^T + ts^{*T}\right) - \sum_{1\leq i<j\leq n}\left[\tilde{s}^{(ij)}b^{(ij)T} + b^{(ij)}\tilde{s}^{(ij)T} + \frac{2b^{(ij)}b^{(ij)T}}{\sigma}\right]$$

We give some lower bounds for the eigenvalues of the terms in $B(s^*)$, using the inequalities

$$\begin{aligned} -2\|t\|\cdot\|s^*\|I_n \preceq s^*t^T + ts^{*T} &\preceq 2\|t\|\cdot\|s^*\|I_n, \qquad 0 \preceq tt^T \preceq \|t\|^2 I_n, \qquad (5.23)\\ 0 \preceq \sum_{1\leq i<j\leq n} b^{(ij)}b^{(ij)T} &\preceq \sum_{1\leq i<j\leq n}\|b^{(ij)}\|^2 I_n \preceq \|T\|_F^2 I_n, \quad 0 \preceq s^*s^{*T},\\ -2\|T\|_F\|s^*\|I_n \preceq \sum_{1\leq i<j\leq n}\left[\tilde{s}^{(ij)}b^{(ij)T} + b^{(ij)}\tilde{s}^{(ij)T}\right] &\preceq 2\sum_{1\leq i<j\leq n}\|\tilde{s}^{(ij)}\|\cdot\|b^{(ij)T}\|I_n \preceq 2\|T\|_F\|s^*\|I_n. \end{aligned}$$

Using the upper bounds in (5.23), and definition of norms in (5.15) – (5.16), we have

$$\begin{aligned} \lambda_{\min}\left[B(s^*)\right] &\geq -\lambda_* - \|T\|\|s^*\| + \frac{\sigma}{2}\|s^*\|^2 - \frac{4}{\sigma}\|t\|^2 - 4\|t\|\|s^*\| - 2\|T\|_F\|s^*\| - \frac{2}{\sigma}\|T\|_F^2, \quad (5.24)\\ &\geq \frac{\sigma}{2}\|s^*\|^2 - 7\|T\|_F\|s^*\| - \frac{6}{\sigma}\|T\|_F^2 - \lambda_*, \quad (5.25) \end{aligned}$$

where in the second inequality, we used $\|T\|_F \geq \|T\|$ and $\|T\|_F \geq \|t\|$, and re-arrangements. Thus, $\lambda_{\min}\left[B(s^*)\right] \geq 0$ if the right-hand side of (5.25) is non-negative, which in turn is equivalent to $\|s_*\|$ being greater or equal to the positive root of the right-hand side of (5.25) (as a function of $\|s_*\|$), namely, $\|s^*\| \geq \frac{7}{\sigma}\|T\|_F + \frac{1}{\sigma}\sqrt{2\sigma\lambda_* + 61\|T\|_F^2}$. In particular, the latter is achieved if $\|s^*\| \geq \max\left\{\frac{32}{\sigma}\|T\|_F, 6\left(\frac{\lambda_*}{\sigma}\right)^{1/2}\right\}$. We now recall the lower bound in (5.18), namely, $\|s^*\| \geq \left(\frac{\|g\|}{3\sigma}\right)^{1/3}$, which we easily check that it satisfies $\left(\frac{\|g\|}{3\sigma}\right)^{1/3} \geq \max\left\{\frac{32}{\sigma}\|T\|_F, 6\left(\frac{\lambda_*}{\sigma}\right)^{1/2}\right\}$ when σ satisfies (5.22). □

The bound in (5.22) contains $\|g\|$ in the denominator. When the quartically regularized polynomial is used as a subproblem within a higher-order method, the stopping condition is typically chosen as $\|g\| \geq \epsilon_g$, meaning that the algorithm terminates once an approximate critical point of the subproblem has been identified. As shown in [53], even an approximate critical point is sufficient to guarantee convergence and complexity of the outer algorithm. Therefore, the bound in (5.22) is well defined in the context of high-order optimization methods. For completeness, we also discuss the behaviour in the degenerate case $\|g\| = 0$ in Remark 5.3.2.

Remark 5.3.2. *We give a discussion on what would happen if $\|g\| = 0$.*

- *If $H \succeq \delta I_n \succ 0$, then $m_3(s) - m_3(s^*)$ is both SoS and SoS-convex for any $\sigma > \frac{8\|T\|^2}{\delta}$; see Section 5.3.3.1.*
- *If $T = 0$, or if T has special structure $T[s]^3 = t^T s\,\|s\|^2$, then $m_3(s) - m_3(s^*)$ is SoS for every $\sigma > 0$; see Section 5.4.2.*
- *If H is not PSD and T is general, determining whether $m_3(s) - m_3(s^*)$ becomes SoS for sufficiently large $\sigma > 0$ remains an open question. We leave this analysis to future work.*

Beyond the behavior of $B(s^*)$, we also remark that if s^* is any second-order local minimizer of the model $m_3(s)$, then $\|s^*(\sigma)\|$ is a non-increasing function of σ. Further details are provided in Section A.3.1.3 and using Cauchy analysis to give upper bound for the global minimum $m(s^*)$ can be found in Section A.3.1.4.

5.3.3.1 Locally Strongly Convex m_3: Convexification, SoS convex and Unique global minimizer

If m_3 is locally strongly convex at $s = 0$, i.e., $H = \nabla^2 m_3(0) \succeq \delta I_n \succ 0$, then for sufficiently large σ, the model m_3 remains strongly convex and is, in fact, SoS convex for all $s \in \mathbb{R}^n$ [1]. A detailed discussion on how convexification relates to SoS and SoS-convexity can be found in Section 4.2.3.2. Here we highlight the key arguments. If m_3 is strongly convex at $s = 0$, then Lemma 4.2.4 shows that choosing σ such that

$$\sigma > \frac{\|T\|^2}{4\delta} \tag{5.26}$$

guarantees m_3 is strongly convex for all $s \in \mathbb{R}^n$. A sufficient condition for a unique global minimizer is the global strong convexity of m_3, i.e., $\nabla^2 m_3(s) \succeq \delta I_n$ for all s and some $\delta > 0$. Then, (5.26) ensures m_3 has a unique global minimizer. Moreover, if we increase σ further so that

$$\sigma > \frac{8\|T\|^2}{\delta}, \tag{5.27}$$

then by [1], m_3 is SoS-convex. Then, it follows from Lemma 5.2.1 that $q(s) = m_3(s) - m_3(s^*)$ is SoS.

5.3.3.2 Locally Nonconvex m_3: Global Optimality and Size of σ

The size of σ is closely related to the idea of convexification of m_3 with large σ. If m_3 is locally nonconvex at $s = 0$, then no matter how large $\sigma > 0$ is chosen, it is not possible to make m_3 globally convex for all $s \in \mathbb{R}^n$. However, by selecting σ sufficiently large, m_3 becomes convex everywhere except within a small neighborhood of the origin. In this section, we show that for sufficiently large σ, the stationary points (may be non-unique) lie outside the small locally nonconvex region and are global minimizers of the nonconvex model m_3. Moreover, in this regime, the global necessary and sufficient optimality conditions established in Chapter 4 coincide (more details are discussed in Section A.3.1.5). $m_3(s)$ is SoS exact with zero levels of relaxation. If the global minimizer is unique, the minimum can then be found in polynomial time.

Lemma 5.3.3. *(Adapted from Lemma 4.2.3) If $\lambda_{\min}[H] \leq 0$, let $s_0 > 0$ be any scalar, and suppose that σ satisfies*

$$\sigma > 2\max\left\{\lambda_* s_0^{-2}, \quad \|T\| s_0^{-1}\right\}, \tag{5.28}$$

then $m_3(s)$ is convex for all $\|s\| \geq s_0$.

Corollary 5.3.2. *Assume that m_3 is locally nonconvex at $s = 0$, that is, $H = \nabla^2 m_3(0)$ is not positive definite. Let $s^* \in \mathbb{R}^n$ satisfies $\nabla m_3(s^*) = 0$. If σ large enough such that (A.27) holds, then $m_3(s)$ is convex for all $\|s\| \geq \sqrt{\frac{2}{3}}\,\|s^*\|$.*

Proof. Suppose that (A.27) holds. Then, by Result 1 in Theorem A.3.3, we have

$$\sigma \geq 2\max\left\{\lambda_*\left\|\sqrt{\frac{2}{3}}\,s^*\right\|^{-2}, \|T\|\left\|\frac{2}{3}s^*\right\|^{-1}\right\} > 2\max\left\{\lambda_*\|s_0\|^{-2}, \|T\|\,\|s_0\|^{-1}\right\},$$

where $s_0 = \sqrt{\frac{2}{3}}\,s^*$. Therefore, by Lemma 5.3.3, $m_3(s)$ is convex for all $\|s\| \geq \|s_0\| = \sqrt{\frac{2}{3}}\,\|s^*\|$. □

Remark 5.3.3. *(Region of convexification) If σ is sufficiently large so that (5.22) holds, then any stationary point s_* of m_3 with $m_3(s_*) \leq m_3(0)$ is a global minimizer (Theorem 5.3.3 and Theorem 5.3.1). In this regime, m_3 is nonconvex only in a small neighborhood of the origin and is convex outside the region $\{ s \in \mathbb{R}^n : \|s\| \leq \|s_0\| = \sqrt{\frac{2}{3}} \|s^*\| \}$. Moreover, any global minimizer lies outside this nonconvex region, and no stationary point with $m_3(s_*) \leq m_3(0)$ exists within it. This follows because, when (A.27) holds, every stationary point of m_3 with $m_3(s_*) \leq m_3(0)$ is a global minimizer, and the global minimizer necessarily lies in the convex region, implying that no stationary point with $m_3(s_*) \leq m_3(0)$ can exist within it.*

Remark 5.3.4. *(Condition for uniqueness of the global minimizer) As shown in this section, there are no stationary points with $m_3(s_*) \leq m_3(0)$ inside the inner nonconvex region. To establish the uniqueness of the global minimizer, one must additionally guarantee that the stationary point outside the nonconvex region is unique. In that case, this unique stationary point is also the unique global minimizer. Without such additional conditions, we can ensure existence of a global minimizer, but not necessarily uniqueness.*

Remark 5.3.5. *(Comparing different σ bound) Section 4.2.3.1 (also summarized in this chapter's notation in Theorem A.3.3) provides a bound under which the necessary and sufficient conditions coincide. The bounds in Theorem A.3.3 and Theorem 5.3.3 are comparable in magnitude; both depend only on the coefficients of m_3, and both are of order[3] $\mathcal{O}(\|g\|^{-2})$. In the locally nonconvex regime, a stationary point that achieves a decrease in the model value is the global minimizer, and the necessary and sufficient conditions coincide. Since $m_3(s) - m_3(s^*)$ is a SoS polynomial, if the minimizer is unique, the global minimization of m_3 can be reformulated as a semidefinite program (SDP), from which the global minimum can be recovered in polynomial time.*

5.4 Classes of SoS Quartically Regularized Polynomials

In this section, we present several special cases and variants of the quartically regularized cubic model. In some of these cases, it is possible to remove the lower bound on σ in (5.22), and still ensure that $m_3(s) - m_3(s^*)$ is SoS for all $\sigma > 0$ where s^* denotes the global minimizer. Namely,

- Univariate quartically regularized cubic model ($n = 1$), see Section 5.4.1.
- Quadratic quartically regularized model (QQR), where the cubic tensor is zero ($T = 0$), see Theorem 5.4.3.
- A special tensor model where $T[s]^3 = (t^T s)\,\|s\|^2$ with $t \in \mathbb{R}^n$, see Section 5.4.2.

We also investigate the Schnabel model introduced in [169] for unconstrained optimization (see Section 5.4.3). The model is based on a fourth-order regularization term involving the separable quartic norm $\sum_{i=1}^n s_i^4$. For this separable quartic norm, we find that there exist quartic polynomials that cannot be expressed as SoS, even in the limit as $\sigma \to \infty$. This observation is consistent with the example given in [3, Thm 3.3]. A summary of all cases is given in Table 5.1.

5.4.1 Univariate Quartically Regularized Cubic Mode

Lemma 5.4.1. *For $n = 1$, s^* satisfies*

$$\nabla m_3(s^*) = 0 \qquad \text{and} \qquad H + \frac{1}{3} T[s^*] + \sigma \|s^*\|^2 \geq \frac{T^2}{18\sigma}, \tag{5.29}$$

if and only if s^* is the global minimum.

[3] In iterative algorithms for minimizing f, We solve a sequence of cubic subproblems m_3, with a stopping criterion $\|g\| \geq \epsilon_g$. Therefore, in both bounds we have $\sigma = O(\epsilon_g^{-2})$.

Table 5.1: Summary of SoS conditions for $q(v) = m_3(s^* + v) - m_3(s^*)$ and their relation to global optimality.

Polynomial Type	**Condition for $q(v)$ to be SoS**	**Is $m_3(s) - m_3(s^*)$ SoS?**
General m_3 (Section 5.3.3)	$\nabla m_3(s^*) = 0$ and $B(s^*) \succeq 0$ (5.7)	Yes, if σ is large enough.
Univariate m_3 (Section 5.4.1)	$\nabla m_3(s^*) = 0$ and $H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2 \geq \frac{T^2}{18\sigma}$	Yes, for all $\sigma > 0$.
QQR model ($T = 0$) (Theorem 5.4.3)	$\nabla m_3(s^*) = 0$ and $H + \sigma\|s^*\|^2 I_n \succeq 0$	Yes, for all $\sigma > 0$.
Special T model $T[s]^3 = t^T s\|s\|^2$ (Section 5.4.2)	$\nabla m_3(s^*) = 0$ and $B_T(s^*) \succeq 0$ (5.31)	Yes for all $\sigma > 0$.
Schnabel Model (Section 5.4.3)	$\nabla m_{\mathcal{S}}(s^*) = 0$ and $B_{\mathcal{S}}(s^*) \succeq 0$ (5.35)	Sometimes no, even if σ is large.

Proof. Let $q(v) := m_3(s^* + v) - m_3(s^*)$, using (5.5), we have

$$\begin{aligned} 2q(v) &= 2\underbrace{\nabla m_3(s^*)^T}_{=0} v + \frac{1}{2}v^T\left[\nabla^2 m_3(s^*)[v]^2 + \frac{1}{3}T[v]^3 + 2\sigma s^{*T}v + \frac{\sigma}{2}\|v\|^2\right]v \\ &\underset{\text{use } n=1}{=} \frac{1}{2}v^T\left[\left(H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2\right) + 2\sigma s^{*2} + \frac{2}{3}T[s^*] + \frac{1}{3}T[v] + 2\sigma s^* v + \frac{\sigma}{2}v^2\right]v \\ &= \frac{1}{2}v^T\left[H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2 - \frac{T^2}{18\sigma}\right]v + \frac{\sigma}{4}\left(v + 2s^* - \frac{T}{3\sigma}\right)^2 v^2. \end{aligned}$$

'$\Rightarrow$' Clearly, if (5.29) is satisfied, $m_3(s^* + v) - m_3(s^*) \geq 0$ for all v. Therefore, s^* is the global minimum.

'$\Leftarrow$' If s^* is a global minimum, then it is a stationary point, therefore $\nabla m_3(s^*) = 0$. Since s^* is the global minimum $m_3(s^* + v) - m_3(s^*) \geq 0$ for all $v \in \mathbb{R}^n$. By setting $v = -(2s^* + \frac{1}{3\sigma}T)$, we derive that $H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2 \geq \frac{T^2}{18\sigma}$. □

Hilbert [95] proved that every nonnegative univariate polynomial is SoS. Here we provide an alternative proof of this result using global optimality conditions.

Theorem 5.4.1. *For $n = 1$, if s^* is the global minimizer, then $q(v) = m_3(s^* + v) - m_3(s^*)$ is SoS.*

Proof. Since s^* is the global minimum, we have $\nabla m_3(s^*) = 0$, according to (5.5), we can write

$$m_3(s^* + v) - m_3(s^*) = \frac{1}{2}\begin{bmatrix} 1 \\ v \\ v^2 \end{bmatrix}\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \nabla^2 m_3(s^*) & \sigma s^* + \frac{T}{6} \\ 0 & \sigma s^* + \frac{T}{6} & \frac{\sigma}{2} \end{bmatrix}}_{Q_a}\begin{bmatrix} 1 \\ v \\ v^2 \end{bmatrix}.$$

Since s^* is the global minimizer, by Lemma 5.4.1, (5.29) holds and

$$\nabla^2 m_3(s^*) = H + T[s^*] + 3\sigma\|s^*\|^2 \underset{(5.29)}{\geq} 2\sigma\|s^*\|^2 + \frac{2}{3}Ts^* + \frac{T^2}{18\sigma}.$$

Note that the term in $3\sigma\|s^*\|^2$ is only true for $n = 1$ case, in $n \geq 2$, the term will be $\sigma(\|s^*\|^2 I_n + ss^T)$. Therefore, The determinant of the submatrix in Q_a,

$$\frac{\sigma}{2}\nabla^2 m_3(s^*) - \left(\sigma s^* + \frac{T}{6}\right)^2 \geq \sigma^2\|s^*\|^2 + \frac{\sigma}{3}Ts^* + \frac{T^2}{36} - \left(\sigma s^* + \frac{T}{6}\right)^2 = 0.$$

We deduce that $Q_a \succeq 0$, therefore $m_3(s^* + v) - m_3(s^*)$ is SoS. □

Remark 5.4.1. *When $T = 0$, (5.29) becomes $\nabla m_3(s^*) = 0$ and $H + \sigma\|s^*\|^2 I_n \succeq 0$ which coincides with the necessary and sufficient global optimality condition for the QQR polynomial*

$$\nabla m_3(s^*) = 0, \qquad B(s^*) = H + \sigma\|s^*\|^2 I_n \succeq 0. \tag{5.30}$$

Remark 5.4.2. *Consider global optimality condition in Theorem 4.2.1 and Theorem 4.2.2, for $n = 1$, (Necessary cond.) gives to $H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2 \geq -\frac{1}{3}T[s^*] + \frac{|T|}{3}\|s^*\| \geq 0$ and (Sufficient cond.) gives to $H + \frac{1}{3}T[s^*] + \sigma\|s^*\|^2 \geq \frac{\hat{T}^2}{18\sigma} - \frac{1}{3}T[s^*] + \frac{|T|}{3} \geq \frac{\hat{T}^2}{18\sigma}$. note that (5.29) lies between (Necessary cond.) and (Sufficient cond.), such that*

$$\text{(Sufficient cond.)} \Rightarrow (5.29) \Rightarrow \text{(Necessary cond.)}.$$

Lemma 5.4.1 provides a tighter necessary and sufficient condition for global optimality in the $n = 1$ case.

5.4.2 Quartically Regularized Cubic Model with Special T

We can also obtain tight necessary and sufficient global optimality conditions for the quartically regularized problem when the cubic tensor has the special structure $T[s]^3 = t^T s\,\|s\|^2$.

Lemma 5.4.2. *If for any $s \in \mathbb{R}^n$, T satisfies $T[s]^3 = t^T s\|s\|^2$ for some $t \in \mathbb{R}^n$ then s^* satisfies*

$$\nabla m_3(s^*) = 0 \qquad \text{and} \qquad B_T(s^*) := H + T[s^*] + \sigma\|s^*\|^2 I_n - \frac{1}{3}(s^* t^T + t s^{*T}) - \frac{tt^T}{18\sigma} \succeq 0, \tag{5.31}$$

if and only if s^* is the global minimum.

Proof. Let $q(v) := m_3(s^* + v) - m_3(s^*)$. Using the expression of $B_T(s^*)$, we have

$$\nabla^2 m_3(s^*) = B_T(s^*) + 2\sigma s^* s^{*T} + \frac{1}{3}(s^* t^T + t s^{*T}) + \frac{tt^T}{18\sigma} I_n$$

Substituting into (5.5), we have

$$\begin{aligned}
2\big[q(v) - \nabla m_3(s^*)^T v\big] &= v^T \Big[\underbrace{B_T(s^*) + 2\sigma s^* s^{*T} + \frac{1}{3}(s^* t^T + t s^{*T}) + \frac{tt^T}{18\sigma} I_n}_{=\nabla^2 m_3(s^*)} + \frac{1}{3}T[v] + 2\sigma s^{*T} v + \frac{\sigma}{2}\|v\|^2 I_n \Big] v \\
&= B_T(s^*)[v]^2 + 2\sigma(v^T s^*)^2 + \frac{2}{3}(v^T s^*)(t^T v) + \frac{(v^T t)^2}{18\sigma} + \frac{1}{3}(t^T v)\|v\|^2 + 2\sigma s^{*T} v\|v\|^2 + \frac{\sigma}{2}\|v\|^4 \\
&= B_T(s^*)[v]^2 + \frac{\sigma}{2}\left(\|v\|^2 + 2v^T s^* + \frac{v^T t}{3\sigma}\right)^2 \\
&= B_T(s^*)[v]^2 + \frac{\sigma}{2}\left[\left\|v + \left(s^* + \frac{t}{6\sigma}\right)\right\|^2 - \left\|s^* + \frac{t}{6\sigma}\right\|^2\right]^2
\end{aligned}$$

'$\Rightarrow$' Clearly, if (5.31) is satisfied, $m_3(s^* + v) - m_3(s^*) \geq 0$ for all v. Therefore, s^* is the global minimum.

'$\Leftarrow$' If s^* is a global minimum, then it is a stationary point, therefore $\nabla m_3(s^*) = 0$. Let $w := v + 2s^* + \frac{t}{3\sigma}$. Assume $w^T v = 0$, since s^* is the global minimum, we have $m_3(s^* + v) - m_3(s^*) \geq 0$ for all $v \in \mathbb{R}^n$, thus $B_T(s^*)[v]^2 \geq 0$ for v orthogonal to w.

Otherwise, assume $w := v + 2s^* + \frac{t}{3\sigma}$ and $s_c := s^* + \frac{t}{6\sigma}$ are not orthogonal, the line $s_c + \tilde{k}w$ intersects the ball centred at the origin of radius $\|s_c\|$ at two points, s_c and u_*, with $\|u_*\| = \|s_c\|$. We set $v = -s_c + u_* = -\big(s^* + \frac{t}{6\sigma}\big) + u_*$. Since s^* is the global minimum, we have $m_3(s^* + v) - m_3(s^*) \geq 0$ for all $v \in \mathbb{R}^n$, we arrive at $B_T(s^*)[v]^2 \geq 0$ for v not orthogonal to w. □

Theorem 5.4.2. *For $n \geq 2$, assume for any $s \in \mathbb{R}^n$, T satisfies $T[s]^3 = t^T s\|s\|^2$ for some $t \in \mathbb{R}^n$. If s^* is the global minimizer, then $q(v) = m_3(s^* + v) - m_3(s^*)$ is SoS.*

Proof. Since s^* is the global minimum, we have $\nabla m_3(s^*) = 0$, according to (5.5), we can write

$$m_3(s^*+v) - m_3(s^*) = \frac{1}{2}\omega \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \nabla^2 m_3(s^*) & (\sigma s^* + \frac{t}{6})\mathbf{1}_n^T & 0 \\ 0 & \mathbf{1}_n(\sigma s^* + \frac{t}{6})^T & \frac{\sigma}{2}\mathbf{1}_n\mathbf{1}_n^T & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{Q_c} \omega \tag{5.32}$$

where ϕ_2 defined as in (5.2), $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^n$ and $\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{n\times n}$ is a matrix with all entries equal to 1s. Let $\omega \in \mathbb{R}^{\frac{1}{2}(n+2)(n+1)}$ be any vector, we write $\omega = [1, v, u, z]^T$ as defined in (5.3). For any ω, or equivalently for any (u, v, z), we have

$$\begin{aligned} 2\omega^T Q_c(v)\omega &= \nabla^2 m_3(s^*)[v]^2 + 2v^T(\sigma s^* + \frac{t}{6})\sum_{i=1}^n u_i + \frac{\sigma}{2}\sum_{i=1}^n u_i^2 + \sigma \sum_{1\le i<j\le n} u_i u_j \\ &= \nabla^2 m_3(s^*)[v]^2 + 2v^T(\sigma s^* + \frac{t}{6})e^T u + \frac{\sigma}{2}(e^T u)^2 \\ &= \underbrace{\nabla^2 m_3(s^*)[v]^2 - 2\sigma\left[v^T(s^* + \frac{t}{6\sigma})\right]^2}_{:\mathcal{J}_1} + \frac{\sigma}{2}\left[\sum_{i=1}^n u_i + 2v^T(s^* + \frac{t}{6\sigma})\right]^2. \end{aligned} \tag{5.33}$$

Since s^* is the global minimum, Lemma 5.4.2 proves that (5.31) holds. Rearranging (5.31) gives $\nabla^2 m_3(s^*)[v]^2 \ge \frac{1}{3}(s^* t^T + t s^{*T}) + \frac{tt^T}{18\sigma} + 2\sigma s^* s^{*T}$ for all $v \in \mathbb{R}^n$. Consequently,

$$2\omega^T Q_c(v)\omega \ge \mathcal{J}_1 \ge v^T\left[\frac{1}{3}(s^* t^T + t s^{*T}) + \frac{tt^T}{18\sigma} I_n + 2\sigma s^* s^{*T}\right]v - 2\sigma v^T\left[(s^* + \frac{t}{6\sigma})(s^* + \frac{t}{6\sigma})^T\right]v \ge 0$$

for all $v \in \mathbb{R}^n$. Thus, we obtained that $Q_c \succeq 0$ and therefore $m_3(s^*+v) - m_3(s^*)$ is SoS. □

Remark 5.4.3. *(Extending to W-norm) By changing I_n to W-norm with $W \succeq 0$, if the model is bounded below or $W \succ 0$, then Theorem 5.4.2 and Lemma 5.4.2 can be extended to allow $T[s]^3 = (t^T s)(s^T W s)$ by using the change of $\tilde{s} = \hat{D}^{1/2}\hat{L}s$ where $\hat{L}^T\hat{D}\hat{L} = W$.*

Quadratic quartically regularized model. By setting $t = 0$, Lemma 5.4.2 can be simplified to Lemma 5.4.3, which coincides with the result in [44, Theorem 8.2.8]. Also, Theorem 5.4.2 can be simplified to Theorem 5.4.3.

Lemma 5.4.3. *s^* satisfies $\nabla m_3(s^*) = 0$ and $B_{QQR} := H + \sigma\|s^*\|^2 I_n \succeq 0$ if and only if s^* is the global minimum.*

Theorem 5.4.3. *For $n \ge 2$, if $T = 0$, s^* is the global minimum of m_3, then $q(v) = m_3(s^*+v) - m_3(s^*)$ is SoS. Proof can be found in Section A.3.2.1*

A discussion for $n = 2$ case. Hilbert proved that every nonnegative bivariate quartic polynomial is SoS (Section A.3.2.2, [95]). Therefore, when $n = 2$, the polynomial $q(v) = m_3(s^*+v) - m_3(s^*)$ is SoS. Notice that in this case $T[v]^3 = t_1 v_1^3 + t_2 v_1^2 v_2 + t_3 v_1 v_2^2 + t_4 v_2^3$, and there are no cross terms of the form $t_{i,j,k} v_i v_j v_k$ with distinct indices (i, j, k). This structure resembles the simplified tensor form $t^T v \|v\|^2$ in Theorem 5.4.2. We further conjecture that, for $n = 2$, a necessary and sufficient global optimality condition (with no relaxation gap) holds for any tensor T. If this conjecture is true, the result may extend to any dimension $n \ge 2$ in which the tensor term $T[v]^3$ contains no cross terms $v_i v_j v_k$ with all distinct (i, j, k), yielding corresponding global optimality certificates. We leave this investigation to future work. [95; 166; 159]. Unlike our proof, this proof does not use the optimality conditions for the quartically regularized subproblem and is universal for any nonnegative bivariate quartic polynomial. We also provide a detailed comparison between the proof technique used there and the one developed in the literature. The proof for binary quartic forms [95; 166; 159] cannot be directly generalized

to higher dimensions in Section A.3.2.2 due to insufficient degrees of freedom for cubic cancellation. For $n = 3$, a homogeneous cubic terms involves $\binom{3+3-1}{3} = 10$ monomial coefficients, whereas the expression $t^T s\|s\|^2$ (or $t^T s\|s\|_W^2$) where $t^T s$ is linear and $\|s\|$ (or $\|s\|_W^2$) is quadratic, has only three to six independent coefficients. Hence, in general one cannot find $t^T s\|s\|^2$ that exactly reproduces an arbitrary homogeneous cubic terms. The lack in degree of freedom prevents the complete elimination of the cubic component.

5.4.3 Schnabel Model

Schnabel, Robert, and Chow [169] proposed a formulation for unconstrained optimization based on a fourth-order model of the objective function f. The model is given by

$$m_{\mathcal{S}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}\sum_{j=1}^{k}(a_j^T s)^2(b_j^T s) + \frac{1}{4}\sum_{j=1}^{k}\sigma_j(a_j^T s)^4 \qquad \text{(Schnabel Model)}$$

where $\sigma_j > 0$, $a_j, b_j \in \mathbb{R}^n$ are specific vectors which correspond to the low–rank approximations of the third and fourth-order derivatives and are built by interpolating function and gradient information from past iterates. Schnabel, Robert, and Chow showed that although (Schnabel Model) is a fourth-order polynomial in n variables, it can be reduced to the minimization of a fourth-order polynomial in k variables plus a quadratic in $n-k$ variables. Note that the quartic term in $m_{\mathcal{S}}$ does not necessarily ensure that $m_{\mathcal{S}}(s) \to +\infty$ as $\|s\| \to \infty$ in all directions. Therefore, the Schnabel model may not be bounded below and cannot guarantee the existence of a global minimizer of the model function. In [169; 170], Schnabel, Robert, and Chow used this model within a trust-region framework to obtain global convergence of the overall algorithm.

Let $q_{\mathcal{S}}(v) := m_{\mathcal{S}}(s+v) - m_{\mathcal{S}}(s)$. We begin with a technical lemma that provides alternative equivalent expressions for $q_{\mathcal{S}}(v)$. This lemma is a simplified version of the result in Lemma 5.3.1.

Lemma 5.4.4. *For any vector $s, v \in \mathbb{R}^n$, let $q_{\mathcal{S}}(v) := m_{\mathcal{S}}(s+v) - m_{\mathcal{S}}(s)$. We have*

$$q_{\mathcal{S}}(v) := \nabla m_{\mathcal{S}}(s)^T v + \frac{1}{2}\nabla^2 m_{\mathcal{S}}(s)[v]^2 + \sum_{j=1}^{k}\left[\sigma_j(a_j^T v)^2(s^T a_j) + \frac{1}{6}(a_j^T v)^2(b_j^T v) + \frac{\sigma_j}{4}(a_j^T v)^4\right]. \quad (5.34)$$

where $\nabla^2 m_{\mathcal{S}}(s) = H + \sum_{j=1}^{k}(a_j a_j^T)(b_j^T s) + \sum_{j=1}^{k} 3\sigma_j(a_j a_j^T)(a_j^T s)^2$.

Previously, in Lemma 5.4.2, we established necessary and sufficient conditions for the global minimizer for m_3. For (Schnabel Model), however, there is an inherent gap between necessary and sufficient conditions. Theorem 5.4.4 proves the sufficient conditions for global optimality and its relationship to SoS characterization. Theorem 5.4.5 proves the necessary conditions for global optimality

Theorem 5.4.4. *(SoS and Global Sufficient Optimality) For $n \geq 2$, assume s^* satisfies*

$$\nabla m_{\mathcal{S}}(s^*) = 0 \qquad \text{and} \qquad B_{\mathcal{S}}(s^*) := \nabla^2 m_{\mathcal{S}}(s^*) - 2\sum_{i=1}^{k}\sigma_j\left[(a_j a_j^T)s^* + \frac{b_j}{6\sigma_j}\right]\left[(a_j a_j^T)s^* + \frac{b_j}{6\sigma_j}\right]^T \succeq 0 \quad (5.35)$$

then $q_{\mathcal{S}}(v) = m_{\mathcal{S}}(s^*+v) - m_{\mathcal{S}}(s^*)$ is SoS. Also, if s^* satisfies (5.35), then, s^* is the global minimum.

Proof. Using Lemma 5.4.4, (5.34) and (5.35), we have

$$\begin{aligned} 2q_{\mathcal{S}}(v) &= v^T\left[B_{\mathcal{S}}(s^*) + 2\sum_{i=1}^{k}\sigma_j\left((a_j a_j^T)s^* + \frac{b_j}{6\sigma_j}\right)\left((a_j a_j^T)s^* + \frac{b_j}{6\sigma_j}\right)^T\right]v \\ &\quad + 2\sum_{i=1}^{k}\left[\sigma_j(a_j^T v)^2 s^T(a_j a_j^T)v + +\frac{1}{2}(a_j^T v)^2(b_j^T v) + \frac{\sigma_j}{4}(a_j^T v)^4\right] \\ &= \underbrace{B_{\mathcal{S}}(s^*)}_{\succeq 0}[v]^2 + \sum_{j=1}^{k}\frac{\sigma_j}{2}\left[(a_j^T v)^2 + 2(a_j^T s^*)(a_j^T v) + \frac{b_j^T v}{3\sigma_j}\right]^2. \end{aligned} \quad (5.36)$$

Note that the deduction of (5.36) is similar to the proof of Lemma 5.4.1. Since (5.36) is symmetric and positive semidefinite, the quadratic form $B_{\mathcal{S}}[v]^2$ is SoS. Together with the remaining sum of squares terms, this implies that $q_{\mathcal{S}}(v)$ is itself a SoS polynomial. To prove the second statement, $q_{\mathcal{S}}(v) := m_{\mathcal{S}}(s^* + v) - m_{\mathcal{S}}(s^*)$ is SoS and thus nonnegative for all v. Therefore, s^* is the global minimum. □

Remark 5.4.4. *Using the expression of $\nabla^2 m_{\mathcal{S}}$ in Lemma 5.4.4, the alternative formulation of $B_{\mathcal{S}}(s^*)$ is*

$$B_{\mathcal{S}}(s) := H + T[s] + \sum_{j=1}^{k} \sigma_j (a_j a_j^T)(a_j^T s)^2 - \sum_{j=1}^{k} \frac{a_j a_j^T}{3}\left[s b_j^T + b_j s^T\right] - \sum_{j=1}^{k} \frac{b_j b_j^T}{18\sigma_j} \tag{5.37}$$

where $T = \sum_{j=1}^{k} a_j \circ a_j \circ b_j$. This condition can be viewed as a generalization of (5.31). From Theorem 5.4.1–Theorem 5.4.4, we observe that the SoS conditions are derived in more general forms.

Remark 5.4.5. ***(Separable Regularization Norm)*** *To ensure that (Schnabel Model) is bounded below. Set $k = n$ and, without loss of generality, assume that $\sigma_j = \sigma$ and $a_j = e_j$, where e_j denotes the unit vector with 1 in the j-th entry and 0 elsewhere. Note that any other choice of σ_j, a_j can be reduced to this form through an appropriate change of variables. Let $T = \sum_{j=1}^{n} e_j \circ e_j \circ b_j$, (Schnabel Model) simplifies to*

$$m_{\mathcal{S}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}T[s]^3 + \frac{\sigma}{4}\|s\|_4^4, \qquad \textit{where} \qquad \|s\|_4 = \Big(\sum_{j=1}^{n} s_j^4\Big)^{1/4},$$

The key difference between this model and the quartically regularized polynomial model m_3 lies in the regularization term: here we employ a quartically separable norm *$\|s\|_4^4 = \sum_{j=1}^{n} s_j^4$, whereas m_3 uses a* quartically regularized Euclidean norm *$\|s\|_2^4 = (s_1^2 + \ldots s_n^2)^2$. Note that under this construction, $B_{\mathcal{S}}(s)$ has the expression,*

$$B_{\mathcal{S}}(s) := H + T[s] + \sigma\,\mathrm{diag}\{s_j^2\}_{1\le j\le n} - \sum_{j=1}^{n} \frac{e_j e_j^T}{3}\big(s b_j^T + b_j s^T\big) - \sum_{j=1}^{n} \frac{b_j b_j^T}{18\sigma_j}. \tag{5.38}$$

Theorem 5.4.5. *(Global Necessary Optimality) Let $Im(A) := \mathrm{span}\{a_1, \ldots, a_k\}$ and its orthogonal complement space $Im(A)^{\perp} := \mathrm{span}\{a_{k+1}, \ldots, a_n\}$. Then, if a global minimum exist and is attained and let s^* be thae global minima, then $\nabla m_{\mathcal{S}}(s^*) = 0$ and*

$$B_{\mathcal{S}}(s^*)[v]^2 \ge 0 \qquad \textit{for all} \qquad v \in Im(A)^{\perp}.$$

If $\{a_1, \ldots, a_k\}$ are orthogonal vectors, then we also have

$$B_{\mathcal{S}}(s^*)[a_j]^2 \ge 0 \qquad \textit{for all} \qquad a_j \in Im(A).$$

Proof. If a global minimum exists and s^* is a global minimum, then it is a stationary point; therefore $\nabla m_{\mathcal{S}}(s^*) = 0$. From (5.36),

$$2q_{\mathcal{S}}(v) = B_{\mathcal{S}}(s^*)[v]^2 + \sum_{j=1}^{k} \frac{\sigma_j}{2}\left[\left\|a_j a_j^T v + a_j a_j^T s^* + \frac{b_j}{6\sigma_j}\right\|^2 - \left\|a_j a_j^T s^* + \frac{b_j}{6\sigma_j}\right\|^2\right]^2. \tag{5.39}$$

Note that the deduction of (5.39) is similar to is similar to the proof of Lemma 5.4.1.

1. If $v \in \mathrm{Im}(A)^{\perp}$, then $a_j a_j^T v = 0$ for all $j = 1, \ldots, k$. Then, we have $B_{\mathcal{S}}(s^*)[v]^2 \ge 0$ for $v \in \mathrm{Im}(A)^{\perp}$.

2. Otherwise, for each unit vector $v = \tilde{k}\frac{a_j}{\|a_j\|} \in \mathrm{Im}(A)$, since $a_j^T a_\iota = 0$ for $\iota \neq 0$. Therefore, let $s_c := a_j a_j^T s^* + \frac{b_j}{6\sigma_j}$,

$$2q_{\mathcal{S}}(v) = \tilde{k}^2\|a_j\|^{-2}B_{\mathcal{S}}(s^*)[a_j]^2 + \frac{\sigma_j}{2}\left[\left\|\tilde{k}a_j + s_c\right\|^2 - \left\|s_c\right\|^2\right]^2. \tag{5.40}$$

If $w_j(\tilde{k}) := \tilde{k}a_j + s_c$ and s_c are orthogonal, the (5.40) becomes $2q_{\mathcal{S}}(v) = \tilde{k}^2\|a_j\|^{-2}B_{\mathcal{S}}(s^*)[a_j]^2 \geq 0$ and we deduce that $B_{\mathcal{S}}(s^*)[a_j]^2 \geq 0$. On the other hand, if $w_j(\tilde{k}) := \tilde{k}a_j + s_c$ and s_c are not orthogonal, the line $s_c + w_j(k)$ intersects the ball centred at the origin of radius $\|s_c\|$ at two points, s_c and $u_{j,*}$. Set $\tilde{k} := k^*$ such that $\|\tilde{k}a_j + s_c\|^2 = \|s_c\|^2$. Since s^* is the global minimum, we have $m_{\mathcal{S}}(s^* + v) - m_{\mathcal{S}}(s^*) \geq 0$ for all $v \in \mathbb{R}^n$, we arrive at $B_{\mathcal{S}}(s^*)[a_j]^2 \geq 0$.

□

A natural question arises: is $B_{\mathcal{S}}(s^*)$ positive semidefinite when σ is sufficiently large? Equivalently, if σ is large enough, can $m_{\mathcal{S}}(s) - m_{\mathcal{S}}(s^*)$ be represented as a SoS? Unfortunately, unlike the case of m_3, the answer is *no*. An illustrative counterexample is provided in Section 5.5.1. If we attempt to extend the analysis from Theorem 5.3.2 and Theorem 5.3.3 to the current setting, we find that the bound $\|s^*\| = O(\|g\|^{1/3})$ in Theorem 5.3.2 still holds—with minor adjustments replacing $\|s^*\|$ by $\|s^*\|_4$. However, difficulties arise in the proof of (5.24) in Theorem 5.3.3. In m_3, the regularization term of $B(s^*)$ is $\sigma\|s^*\|^2 I_n$, whose minimum eigenvalue equals $\sigma\|s^*\|^2$. In contrast, the regularization term in $B_{\mathcal{S}}(s^*)$ is $\sigma\,\mathrm{diag}\{s_j^2\}_{1\leq j\leq n}$ as given in (5.38), whose minimum eigenvalue is $\sigma\min_{1\leq j\leq n}\{s_j^2\}$. The step-size bound $\|s^*\| = O(\epsilon_g^{1/3})$ does not directly control this minimum component $\min_j |s_j|$, and hence does not guarantee that the regularization term in $B_{\mathcal{S}}(s^*)$ dominates the remaining terms. Consequently, $B_{\mathcal{S}}(s^*)$ may fail to be PSD. This demonstrates that even a subtle change in the form of the regularization norm can fundamentally alter the SoS properties of the model as $\sigma \to \infty$.

Difference between the special T model and the Schnabel model Under special cases, (Schnabel Model) can be transformed into our special T with a W semi-norm ($W \succeq 0$). If $b_1 = b_2 = \ldots = b_k$ and $\sigma_1 = \sigma_2 = \ldots = \sigma_k = \frac{3}{2}\sigma$, we can set $b_1 := \frac{t}{3}$ and $\sigma_1 := \sigma$,

$$m_{\mathcal{S}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}\Bigg[s^T \underbrace{\Big(\sum_{j=1}^{k} a_j a_j^T\Big)}_{:=W} s\Bigg](t^T s) + \frac{\sigma}{4}\|s\|_W^4. \tag{5.41}$$

Therefore, we have $T[s]^3 = (t^T s)(s^T W s)$ and clearly $W \succeq 0$. Under this specific constructions, if the model is bounded below, then $q_{\mathcal{S}}(v) = m_{\mathcal{S}}(s^*+v) - m_{\mathcal{S}}(s^*)$ is SoS (see Lemma 5.4.2 and Remark 5.4.3).

Remark 5.4.6. *(Difference between the regularization norm) There is no matrix transform / linear change of basis that converts $\sum_{j=1}^k s_j^4$ into the Euclidean norm $\|s\|^4 = \left(\sum_{j=1}^k s_j^2\right)^2$ for $j \geq 2$. More generally, the regularization term in* (Schnabel Model) *takes the form $\sum_{j=1}^k (a_j^T s)^4 = \sum_{j=1}^k (s^T a_j a_j^T s)^2 = \sum_{j=1}^k \|s\|_{W_j}^4$, $W_j = a_j a_j^T$. Here, $\|s\|_{W_j} := \sqrt{s^T W_j s}$ is the matrix-induced quadratic form. Since each W_j is rank-one and positive semidefinite, $\|\cdot\|_{W_j}$ is not a norm on the whole space $\mathbb{R}^n$, but only a* seminorm*: it vanishes for any $s \perp a_j$, even when $s \neq 0$. Only in the case $k = 1$ does the regularization reduce to a single quartic seminorm, i.e.,*

$$(a_1^T s)^4 = \|s\|_{W_1}^4,$$

which is in a form analogous to the quartic regularization norm in the special T model (i.e., $\|s\|_W^4$ or $\|s\|^4$). The quartic semi-norm does not necessarily ensure that $m_{\mathcal{S}}$ is bounded below. For instance, if $s \in \mathrm{Im}(A)^\perp$ and $H[s]^2 < 0$, then $m_{\mathcal{S}}(s) \to -\infty$ as $\|s\| \to \infty$.

Remark 5.4.7. *When $k = 1$, $m_{\mathcal{S}}(s) := f_0 + g^T s + \frac{1}{2}H[s]^2 + \frac{1}{6}(a_1^T s)^2(b_1^T s) + \frac{\sigma_1}{4}(a_1^T s)^4$. In this case, the global necessary optimality condition in Theorem 5.4.5 becomes $\nabla m_{\mathcal{S}}(s^*) = 0$ and $B_{\mathcal{S}}(s^*) \succeq 0$. Hence,*

if a global minimizer exists, the necessary and sufficient conditions coincide and $m_{\mathcal{S}}(s) - m_{\mathcal{S}}(s^*)$ is SoS. This corresponds to the special case in (5.41), which can be transformed into the special T-model with a weighted quartic semi-norm. For $k > 1$, however, the presence of multiple distinct seminorms $\|s\|_{W_j}$ prevents such a change of basis and reduction; the equivalence no longer holds.

5.5 Examples and Preliminary Numerical Results

In this section, we begin by examining several well known quartic examples due to Nesterov and Ahmadi (Section 5.5.1 and Section 5.5.2). These problems are not SoS-exact at the zeroth level of the Lasserre hierarchy. We compare these examples with their quartically regularized counterparts and analyze—both theoretically and numerically—how different regularization norms and problem constraints affect SoS representability. Further numerical results and a high-order algorithm that incorporates the SoS framework for the Taylor model are presented in a later chapter; see Section 6.5.

5.5.1 Examples with Alternative Quartic Norms

In the special T-model or the QQR model, which uses a quartic Euclidean norm, $\|s\|^4 = \left(\sum_{i=1}^n s_i^2\right)^2$, the shifted polynomial $m_3(s) - m_3(s^*)$ is SoS for all $\sigma > 0$. However, if we replace the Euclidean norm with the quartic ℓ_4-norm, $\|s\|_4^4 = \sum_{i=1}^n s_i^4$, then the shifted polynomial need not remain SoS for all $\sigma > 0$. We illustrate using a modified example by Ahmadi, Dibek and Hall in [3, Thm. 3.3]. Consider a trivariate quartic polynomial,

$$m_{\mathcal{A}}(s) = 2\|s\|^2 + 8(s_1s_2 + s_1s_3 + s_2s_3) + \frac{\sigma}{4}(s_1^4 + s_2^4 + s_3^4) = \hat{H}[s]^2 + \frac{\sigma}{4}\sum_{i=1}^3 s_i^4, \qquad \sigma > 0. \tag{5.42}$$

whose eigenvalues are $-4, -4$, and 20, the Hessian is indefinite at the origin. Hence, $m_{\mathcal{A}}$ is nonconvex for every $\sigma > 0$.[4]

Lemma 5.5.1. *For any $\sigma > 0$, let $m_{\mathcal{A}}(s)$ be defined as in (5.42). Denote its global minimum value by $m_{\mathcal{A}}(s^*)$. Then, for every $\sigma > 0$, the polynomial $m_{\mathcal{A}}(s) - m_{\mathcal{A}}(s^*)$ is nonnegative but not a SoS.*

Proof. For the specific case of $\sigma = 4$, we denote

$$m_{\mathcal{A},\sigma_0}(s) := 2\|s\|^2 + 8(s_1s_2 + s_1s_3 + s_2s_3) + (s_1^4 + s_2^4 + s_3^4) = \hat{H}[s]^2 + \sum_{i=1}^3 s_i^4.$$

Note that $m_{\mathcal{A},\sigma_0}(s)$ is bounded below and possesses six global minimizers[5]. We denote the global minimizers and global minimum as $(s^*_{(\sigma_0)}, m^*_{(\sigma_0)})$. [3, Thm 3.3] proves that, the shifted polynomial $m_{\mathcal{A},\sigma_0}(s) - m^*_{(\sigma_0)}$ is not SoS in s. We prove that this result extends to all $\sigma > 0$. Take any fixed $\sigma > 0$, consider the change of variables $\tilde{s} = \frac{\sigma^{1/2}}{2}s$. Then, from (5.42), we have

$$m_{\mathcal{A}}(s) \;=\; \hat{H}[s]^2 + \frac{\sigma}{4}\sum_{i=1}^3 s_i^4 = 4\sigma^{-1}\left[\hat{H}\big[\frac{\sigma^{1/2}}{2}s\big]^2 + \sum_{i=1}^3 \big(\frac{\sigma^{1/2}}{2}s_i\big)^4\right] = 4\sigma^{-1}m_{\mathcal{A},\sigma_0}(\tilde{s}).$$

By this scaling, $m_{\mathcal{A}}(s)$ attains the global minimizers and global minimum at $(2\sigma^{-1/2}s^*_{(\sigma_0)}, 4\sigma^{-1}m^*_{(\sigma_0)})$. As scaling preserve non-SoSness, given $m_{\mathcal{A},\sigma_0}(\tilde{s}) - m^*_{(\sigma_0)}$ is not SoS, we deduce that $4\sigma^{-1}\big[m_{\mathcal{A},\sigma_0}(\tilde{s}) - m^*_{(\sigma_0)}\big]$ is also not SoS in s. Consequently, $m_{\mathcal{A}}(s) - m_{\mathcal{A}}(s^*) = m_{\mathcal{A}}(s) - 4\sigma^{-1}m^*_{(\sigma_0)}$ is not SoS for any $\sigma > 0$. □

[4] Equivalently, restricting $m_{\mathcal{A}}$ to a line in the direction of an eigenvector associated with the eigenvalue -4 yields a univariate quartic whose second derivative at the origin is negative. Thus, the nonconvexity can also be verified from this univariate restriction.

[5] Numerically, we obtain $m^*_{(\sigma_0)} = -2.1443$, and $s^*_{(\sigma_0)} = \pm(-1.1498, 0.6674, 0.6674)$ and their permutations.

Lemma 5.5.1 can be verified numerically. Using an SoS relaxation (e.g., `SoSTOOLS`) with $\sigma = 4$, the moment-SDP solver returns a lower bound but not a global minimum, indicating that the SoS relaxation *fails* to recover the true optimum[6].

Connection to the Schnabel model. For any $\sigma > 0$, the polynomial (5.42) can be written in the form (Schnabel Model) by choosing $a_j = e_j, b_j = 0$, for $j = 1, 2, 3$ $g = 0$, and $H = \hat{H}$. Using $s^* = 2\sigma^{-1/2} s^*_{(\sigma_0)}$ and $s^*_{(\sigma_0)} = [-1.1498, 0.6674, 0.6674]^T$. We derive from (5.37), $B_{\mathcal{S}}(s^*)$ for $m_{\mathcal{A}}$ has the expression

$$B_{\mathcal{S}}(s^*) = H + \operatorname{diag}(s^*_{(\sigma_0)}{}^2) = \begin{bmatrix} 9.2878 & 8 & 8 \\ 8 & 5.7819 & 8 \\ 8 & 8 & 5.7815 \end{bmatrix} \tag{5.43}$$

for any $\sigma > 0$. $B_{\mathcal{S}}(s^*)$ has eigenvalues $[-2.2, 0, 23.1]$ and is therefore indefinite for all $\sigma > 0$. Thus, the sufficient SoS condition in Theorem 5.4.4 fails. Yet, we verify that s^*, the necessary global condition in Theorem 5.4.5 is satisfied: $B_{\mathcal{S}}(s^*)[e_j]^2 \geq 0, j = 1, 2, 3$.

Replacing the separable quartic norm. Let us now consider the Euclidean quartic regularization,

$$m_{\mathcal{E}}(s) = 2\|s\|^2 + 8(s_1 s_2 + s_1 s_3 + s_2 s_3) + \sigma\|s\|^4 = \hat{H}[s]^2 + \sigma\|s\|^4 \qquad \sigma \geq 0. \tag{5.44}$$

$m_{\mathcal{E}}(s)$ is a nonconvex quadratic quartically regularized (QQR) polynomial. By Theorem 5.4.3, we know that $m_{\mathcal{E}}(s) - m_{\mathcal{E}}(s^*)$ is SoS for every $\sigma \geq 0$. In Lemma 5.5.2, we also derived the explicit expression for its global minimizers.

Lemma 5.5.2. *Let $s^* \in \mathbb{R}^3$ satisfy $s^*_1 + s^*_2 + s^*_3 = 0, \|s^*\| = \sigma^{-1/2}$. Then s^* is a global minimizer of $m_{\mathcal{E}}$, and its minimum value is $m_{\mathcal{E}}(s^*) = -\sigma^{-1}$.*

Proof. $m_{\mathcal{E}}(s) + \sigma = \sigma\big(\|s\|^2 - \sigma^{-1}\big)^2 + 4(s_1 + s_2 + s_3)^2$, which is nonnegative and SoS for all s. The lower bound $-\sigma^{-1}$ is attained precisely when $s_1 + s_2 + s_3 = 0$ and $\|s\| = \sigma^{-1/2}$. □

Lemma 5.5.2 shows that $m_{\mathcal{E}}$ has infinitely many global minimizers: every point on the circle $\{s \in \mathbb{R}^3 : s_1 + s_2 + s_3 = 0,\ \|s\| = \sigma^{-1/2}\}$. Although $m_{\mathcal{E}}$ is SoS, the moment relaxations numerical solvers (e.g., `yalmip`) return a rank-deficient moment matrix. This is because (5.44) has infinitely many global minimizers—the rank condition for finite relaxation cannot be satisfied. From (5.30), we have

$$B_{QQR}(s^{*2}) = H + \sigma \operatorname{diag}(s^{*2}) = \begin{bmatrix} 4 + \sigma\|s^*\|^2 & 8 & 8 \\ 8 & 4 + \sigma\|s^*\|^2 & 8 \\ 8 & 8 & 4 + \sigma\|s^*\|^2 \end{bmatrix} = \begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix}. \tag{5.45}$$

$B_{QQR}(s^{*2})$ has eigenvalue $0, 8, 24$ and $B_{QQR}(s^{*2}) \succeq 0$ for all $\sigma > 0$. The subtle difference between (5.43) and (5.45) $B_{\mathcal{S}}(s^*)$, the separable quartic norm does not guarantee positive definiteness, whereas the Euclidean quartic norm does.

Remark 5.5.1 (Coercivity and attainment of a global minimum)**.** *The quartic regularization term $\|s\|^4$ is coercive. Consequently, for any $\sigma > 0$, it dominates the lower-degree terms in the model m_3, so that $m_3(s) \to +\infty$ as $\|s\| \to \infty$. Since m_3 is continuous, coercivity ensures that its global minimum is attained. Boundedness from below alone does not guarantee the existence of a global minimizer. For example, consider $f(x_1, x_2) = x_1^4 + (x_1 x_2 - 1)^2$. This function is nonnegative but not coercive. Indeed, along the sequence $x_1^{(k)} = \frac{1}{k}, x_2^{(k)} = k$, we have $f\big(x_1^{(k)}, x_2^{(k)}\big) = \frac{1}{k^4} \to 0$. Thus, $\inf f = 0$. However, this value is not attained. This example illustrates the role of coercivity in guaranteeing attainment of the global infimum.*

Robustness: the effect of perturbations on global minimizers and on SoS representability. We introduce a small perturbation to (5.44), either by perturbing a single coefficient,

$$\hat{m}_{\mathcal{E},1}(s) = \frac{\sigma}{4}\|s\|^4 + 2\|s\|^2 + 8\Big[(1 + 10^{-5})s_1 s_2 + s_1 s_3 + s_2 s_3\Big].$$

[6]For very large σ (e.g., $\sigma \geq 2500$), the solver reports a global minimum close to the origin, but this is due to numerical errors, not because the shifted polynomial becomes SoS.

Alternatively, we can add small random perturbations to several entries

$$\hat{m}_{\mathcal{E},2}(s) = \sigma\|s\|^4 + 2\|s\|^2 + 8\Big[(1+10^{-6}\,\xi_{12})s_1s_2 + (1+10^{-6}\,\xi_{13})s_1s_3 + (1+10^{-6}\,\xi_{23})s_2s_3\Big],$$

where each $\xi_{ij} \sim \mathcal{N}(0,1)$ represents an independent Gaussian random variable. When we apply the moment-relaxation solver `yalmip` to both perturbed polynomials, it returns a successful status of `1`, indicating that the ring of infinite global minimizers breaks—after perturbation—into one or at most two global minimizers. Consequently, the rank condition is satisfied. Small perturbations in the data (i.e., in the coefficients) can change the numerical outcome returned by `yalmip`. This behaviour observed arises due to the lack of robustness of this specific example.

Perturbing the cubic term. We also consider perturbing the cubic term (5.42),

$$m_{\mathcal{A},2}(s) = 2\|s\|^2 + 8(s_1s_2 + s_1s_3 + s_2s_3) + 0.1(s_1^3 + s_2^3 + s_3^3) + \sigma(s_1^4 + s_2^4 + s_3^4)$$

for $\sigma = 4$. This polynomial attains its global minimum at $s^* = [-1.1722,\ 0.6700,\ 0.6700]^T, m_{\mathcal{A},2}(s^*) = -2.2409$. Using `SoSTOOLS`, we obtain a lower bound $\gamma^* = -2.3990$, which is strictly less than the true global minimum, implying that $m_{\mathcal{A},2}(s) - \gamma^*$ is still not SoS. In contrast, if we replace the separable quartic term by the Euclidean quartic term, we obtain the variant

$$m_{\mathcal{E},2}(s) = 2\|s\|^2 + 8(s_1s_2 + s_1s_3 + s_2s_3) + 0.1(s_1^3 + s_2^3 + s_3^3) + \sigma\|s\|^4.$$

Using `SoSTOOLS`, we verify that $m_{\mathcal{E},2}(s) - m_{\mathcal{E},2}(s^*)$ is SoS for the tested values of $\sigma \gtrsim 10$, not not SoS for all $\sigma > 0$. This observation also verifies Theorem 5.3.3 and Theorem 5.4.3. Without the cubic term, $m_{\mathcal{E}}(s)$ has a QQR structure and is therefore SoS-exact with zero layer of relaxation for all $\sigma > 0$ (Theorem 5.4.3). After adding a small cubic perturbation, the polynomial $m_{\mathcal{E},2}(s)$ is no longer a quadratic–quartically regularized model but instead becomes a cubic–quartically regularized polynomial. In this case, the bound for σ follows the requirements of Theorem 5.3.3: σ must be sufficiently large to ensure that $B(s)$ is PSD and thus guarantee an SoS representation. Numerically, we observe that the threshold occurs around $\sigma \gtrsim 10$.

Remark 5.5.2.

5.5.2 Cubic and Quartic Maximization over the Euclidean Ball

Nesterov [143] proved that the following cubic or quartic polynomial optimization problems are NP-hard to solve globally.

Theorem 5.5.1 ([143; 120])**.** *The following polynomial optimization problems are NP-hard. Let $A_j \in \mathbb{R}^{n\times n}$ be positive semidefinite matrices.*

1. ***Quartic maximization over the Euclidean ball:***

$$\max_{s\in\mathbb{R}^n}\left\{ f_{\mathcal{N}}(s) := \sum_{j=1}^{k} \left(A_j[s]^2\right)^2 \ :\ \|s\| = 1 \right\}. \tag{5.46}$$

2. ***Cubic maximization over the Euclidean ball:*** *where $g_3(s)$ is a homogeneous cubic polynomial,*

$$\max_{s\in\mathbb{R}^n} \{ g_3(s) \ :\ \|s\| = 1 \}. \tag{5.47}$$

3. ***Quartic minimization under quadratic constraints:*** *where $g_4(s)$ is a homogeneous quartic polynomial,*

$$\min_{s\in\mathbb{R}^n} \{ g_4(s) \ :\ A_j[s]^2 \geq 1, \quad j = 1,\dots,k \}. \tag{5.48}$$

Note that this formulation is an alternative representation of (5.46).

We first study the quartic problem (5.46). Since $f_{\mathcal{N}}$ is 4-homogeneous, maximizing over the unit ball or over the sphere is equivalent: $\max_{\|s\|\leq 1} f_{\mathcal{N}}(s) = \max_{\|s\|=1} f_{\mathcal{N}}(s)$. A Lagrangian reformulation is given by

$$\mathcal{L}_1(s,\lambda) = \sum_{j=1}^{m} \left(A_j[s]^2\right)^2 + \lambda(\|s\|^2 - 1). \tag{5.49}$$

The quartic polynomial $\mathcal{L}(s,\lambda)$ is bounded below with respect to s, but not with respect to λ. The Lagrangian formulation (5.49) is fundamentally different from our quartically regularized polynomial model in the following ways.

Remark 5.5.3. *Several key distinctions explain why maximizing a quartic polynomial over the Euclidean ball is NP-hard, whereas sufficiently quartically regularized nonnegative polynomials can be SoS-exact.*

1. ***Unboundedness in the Lagrange multiplier.*** *The variable pair is (s,λ). While $\mathcal{L}(s,\lambda)$ is quartic in s, it contains* no *regularization term in λ, hence $\mathcal{L}(s,\lambda)$ is unbounded in λ.*

2. ***No freedom to choose a regularization parameter.*** *In quartic regularization, the coefficient $\sigma > 0$ is a user-chosen stabilizing parameter. In contrast, here λ^* is an* optimization variable *determined by the KKT system, not a tunable regularization parameter.*

3. ***Constraint active at global solutions.*** *Any KKT point (s^*,λ^*) corresponding to a global maximizer of (5.49) must satisfy the boundary constraint $\|s^*\| = 1$, implying $\lambda^* \neq 0$. Thus, the constraint term $\lambda(\|s\|^2 - 1)$* cannot *be ignored in the global analysis.*

4. ***Degeneracy when $\|g\| = 0$.*** *Consider the Lagrangian as a function of the joint variable (s,λ). Apart from the quartic terms in s, $\mathcal{L}(s,\lambda)$ contains only a homogeneous cubic term and a constant term, and no first order coefficient $\|g\| = 0$. As discussed in Remark 5.3.2, the case where the gradient vanishes is structurally special and requires careful, case-by-case analysis.*

Similarly, for the cubic maximization over the Euclidean ball in (5.47), we can write the corresponding Lagrange multiplier formulation

$$\mathcal{L}_2(s,\lambda) = g_3(s) + \lambda(\|s\|^2 - 1). \tag{5.50}$$

By writing the complementarity condition in quartic form, $\mathcal{L}(s,\lambda)$ becomes a quartic polynomial in s. However, the four key differences highlighted in Remark 5.5.3 also apply here: $\mathcal{L}(s,\lambda)$ is unbounded in λ (if $\lambda \to -\infty$ and $\|s\| < 1$, then $\mathcal{L}(s,\lambda) \to -\infty$); there is no freedom to choose a regularization parameter (since λ^* is an optimization variable, not a regularization weight); and the constraint is always active at a global maximizer because a cubic polynomial is unbounded; lastly, besides regularization, we have a homogeneous cubic term with $\|g\| = 0$. However, if we modify (5.50) as

$$\min_{s\in\mathbb{R}^n} \mathcal{L}_{c,2}(s) = g_3(s) + \sigma(\|s\|^4 - 1), \tag{5.51}$$

where $\sigma > 0$ is a regularization parameter that can be chosen a priori, then $\mathcal{L}_{c,1}(s)$ becomes a quartically regularized polynomial and is bounded below. In Figure 5.1, we plot the behaviour of the global minimizer $\|s^*\|$ as a function of the regularization parameter σ. For σ increasing from 0.1 to 100, we test whether $\mathcal{L}_c(s) - \mathcal{L}_c(s^*)$ is SoS using `GloptiPoly/MSDP`. Non-SoS cases are marked with red crosses, while SoS cases are plotted by blue circles with the corresponding norm $\|s^*\|$ against the regularization weight σ. The right panel also includes the curve $O(\sigma^{-1/3})$ for comparison which is the theoretical bound of $\|s^*\|$ given in Theorem 5.3.2. We observe that $\mathcal{L}_{c,2}(s) - \gamma$ becomes SoS exact with zero layer of relaxation as σ increases. The verifies our theoretical result in Theorem 5.3.3 that sufficiently regularized nonnegative quartic polynomials are SoS.

Remark 5.5.4. *Note that assume $A_j \succ 0$, a similar modification can be written for (5.49) with*

$$\min_{\tilde{s}\in\mathbb{R}^n} \mathcal{L}_{1,c}(\tilde{s},\lambda) = \sigma\|\tilde{s}\|^4 + (\tilde{s}^T A_j^{-1}\tilde{s} - 1). \tag{5.52}$$

where we use the change of variables $\tilde{s}^T\tilde{s} = A_j[s]^2$. We observe same numerical results with $\mathcal{L}_{1,c}$.

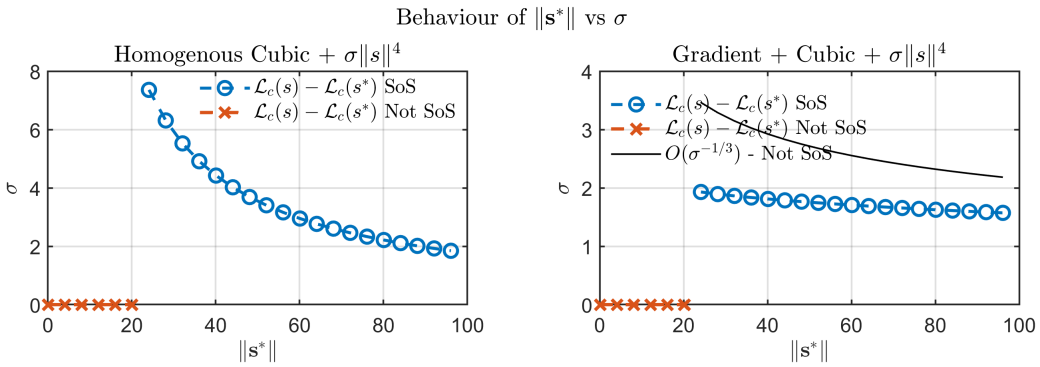


Figure 5.1: **Left:** Homogeneous cubic model $g_3(s) + \sigma\|s\|^4$ with random *negative* coefficients uniformly generated in $[0, -100]$ and zero gradient term. **Right:** Non-homogeneous model $g(s) = g^T s + g_3(s) + \sigma\|s\|^4$, where both the cubic and linear coefficients are randomly generated in $[0, -100]$.

We would like to highlight that (5.51) and (5.50) are not equivalent and do not correspond to the same optimization problem, nor are (5.52) and (5.49). The formulation in (5.50) is the Lagrange multiplier expression of the constrained cubic maximization over the Euclidean ball, for which any optimal solution necessarily satisfies $\|s^*\| = 1$. In contrast, (5.51) defines an unconstrained quartically regularized problem in s, with no restriction on the step size of its global minimizer. The subtle change in formulation alters the SoS representability and tractability of the resulting problem. This observation is consistent with known results in polynomial optimization: adding a constraint set can change the SoS representability of a problem. In the unconstrained case, many nonnegative polynomials are not SoS (as shown by Blekherman [29]), whereas Lasserre formulations often recover SoS certificates on compact domains [108]. Thus, different globalization mechanisms can indeed have a fundamental impact on the tractability of polynomial optimization problems.

In this chapter, we studied the interplay between regularization, convexification, and SoS representability for quartically regularized cubic polynomials arising from high-order adaptive regularization methods. We showed that, under sufficiently strong regularization, a nonconvex nonnegative quartically regularized polynomial admits an SoS representation. This establishes, for the first time, explicit SoS certification results for a family of nonconvex quartically regularized models. In addition, we identify several structured subclasses of quartically regularized cubic polynomials for which global optimality of the model implies that its nonnegativity is *certified* by a SoS decomposition, for all values of the regularization parameter σ. In this sense, our results extend Hilbert's classical characterizations of nonnegative polynomials to a new subfamily motivated by high-order tensor methods. We further propose an algorithmic framework that combines SoS-based Taylor models with adaptive regularization techniques. Several directions for future research naturally arise from this work. These include establishing global complexity guarantees for the AR3+SoS algorithmic framework, and investigating tight semidefinite relaxations with reduced dimensionality. In particular, recent work by [190] leverages global sufficient optimality conditions for cubic quartically regularized (CQR) polynomials (Chapter 3) to derive a tight SDP formulation of size $n + 1$. Another promising direction is to exploit additional structural properties to develop more scalable variants using alternatives [113; 177; 182; 189].

Chapter 6

Global Optimization Techniques for Regularized Polynomial Subproblem: Convexified Models

6.1 Introduction

In the previous chapter, we discussed the SoS property of AR3 subproblem. Recently, Ahmadi, Chaudhry, and Zhang [1] introduced an iterative algorithm based on SoS convex reformulations for minimizing the objective function f. This chapter integrates the SoS-convex Taylor model proposed in [1] with the adaptive regularization techniques in [23; 51; 53; 137] and introduces an algorithmic framework that combines the SoS convex Taylor model with adaptive regularization techniques. Each iteration minimizes a SoS-convex Taylor model, offering a polynomial cost per iteration. While [1] analyzed the local convergence rate and the global convergence of this method (for odd $p \geq 3$) for strongly convex objective functions. The global convergence and complexity for a general nonconvex objective function remain unknown prior to our studies. We show that, for general nonconvex functions, the worst-case evaluation complexity bound is $\mathcal{O}(\epsilon^{-2})$, while for strongly convex functions, an improved evaluation complexity bound of $\mathcal{O}(\epsilon^{-\frac{1}{p}})$ is established. To the best of our knowledge, this is the first global rate analysis for an adaptive regularization algorithm with a tractable high-order sub-problem in nonconvex smooth optimization, opening the way for further improvements.

In the adaptive SoS Convex Taylor method, we approximate f using a multivariate polynomial model $m_c : \mathbb{R}^n \to \mathbb{R}$, referred to as the SoS-convex Taylor model. This model comprises a pth-order Taylor expansion, a second-order perturbation term if the iteration is locally nonconvex (i.e., $\nabla_x^2 f(x_k)$ not positive definite), and a regularization term. In locally convex iterations (i.e., $\nabla_x^2 f(x_k) \succ 0$), the SoS-convex Taylor model precisely corresponds to the ARp sub-problem in (ARp Subproblem). Note that the regularization term for the SoS-convex Taylor model is designed to be sufficiently large to ensure that m_c is SoS-convex[1]. Given that determining the convexity (or otherwise) of polynomials of degree at least four is NP-hard [5], SoS-convexity, which determines a strict subclass of convex functions, offers an effective means of verifying and classifying convexity. Another important benefit of using the SoS-convex Taylor model is that both the process of finding the σ_k to make m_c SoS convex and the process of minimizing the corresponding SoS-convex polynomial m_c can be reformulated as semidefinite programmings (SDP) [1; 174]. SDP is a well-studied subclass of convex optimization problems that can be solved to arbitrary precision within polynomial time [174]. Algorithm 1 in [1] presents a minimization algorithm that employs an SoS-convex Taylor model at each iteration. The chapter presents two key contributions.

1. We present both global convergence and worst-case evaluation complexity results for Algorithm 6.2.1. We show that Algorithm 6.2.1 yields a monotonically decreasing sequence of iterates, $\{f(x_k)\}_{k\geq 0}$, and converges to an ϵ_g-approximate first-order minimizer of f. The function

[1]The definition of SoS polynomials and SoS-convex can be found in Definitions 5.2.1–5.2.2.

value reduction in each strongly convex iteration is $\mathcal{O}(\epsilon_g^{\frac{p+1}{p}})$ for odd $p > 5$, and $\mathcal{O}(\epsilon_g^{\frac{p+3}{p+1}})$ for even $p > 5$. Note that we have improved function value reduction for $p = 3, 5$ at $\mathcal{O}(\epsilon_g^{\frac{2}{p}})$ and for $p = 4$ at $\mathcal{O}(\epsilon_g^{\frac{4}{5}})$. In locally nonconvex cases, the error reduction per iteration is $\mathcal{O}(\epsilon_g^2)$. Therefore, for a general nonconvex function, the overall evaluation complexity bound[2] is $\mathcal{O}(\epsilon_g^{-2})$. However, for strongly convex functions $f(x)$, we establish an improved evaluation complexity bound for the ϵ_g-approximate minimum value of convex functions as $\mathcal{O}(\epsilon_g^{-\frac{1}{p}})$ for odd p and $\mathcal{O}(\epsilon_g^{-\frac{p+1}{p}})$ for even p. To the best of our knowledge, for a generally nonconvex objective function, this is the first to offer a global convergence and a global rate result for a tensor method with a tractable sub-problem; these bounds are not of optimal or best-known accuracy order for the respective classes, but open the way for such improvements.

2. The perturbation size for locally nonconvex iterations and its effect on the complexity bound remain unspecified in [1]. In this chapter, we use an ϵ_g-dependent parameter (denoted here as δ) to act as the perturbation for nonconvex iterations. We provide a detailed analysis of how to choose δ as a function of ϵ_g. Additionally, we explore how the complexity bound varies with the order of the Taylor approximation, p, and the magnitude of perturbation, δ.

The chapter is structured as follows. In Section 6.2, we introduce the SoS-convex Taylor model and the related adaptive regularization framework. In Section 6.3, we demonstrate the existence of an upper bound for the regularization parameter across iterations. In Section 6.4, we provide guaranteed bounds for function value reduction in successful iterations for the three convexity cases, along with the overall complexity bound for Algorithm 6.2.1. We also provide results demonstrating improved function value reduction specifically for the cases of $3 \leq p \leq 5$ and improved global convergence rate for strongly convex objective functions. In Section 6.5, we present our preliminary numerical illustration of theoretical bounds.

6.2 The SoS-convex Taylor Model with Adaptive Regularization

In this section, we introduce the SoS-convex Taylor model and outline the algorithmic framework that integrates the SoS-convex Taylor model with adaptive regularization. In the locally strongly convex and nonconvex iterations, the SoS-convex Taylor model is consistent with that in [1]. Here, we extend the model to the locally nearly strongly convex case, which needs to be addressed when optimizing general objectives. For notational simplicity, we denote $f_k = f(x_k) \in \mathbb{R}$, $g_k = \nabla_x f(x_k) \in \mathbb{R}^n$ and $H_k = \nabla_x^2 f(x_k) \in \mathbb{R}^{n \times n}$. In all three cases, we add the smallest non-negative multiple of the identity matrix to H_k to construct a new second-order term $\bar{H}_k$, such that the smallest eigenvalue of $\bar{H}_k$ is at least δ.

The pth order SoS-convex Taylor model

Set $p' := p + 1$ if p is odd and $p' := p + 2$ if p is even. Note that $p \geq 3$. Let $\delta = {\epsilon_g}^a > 0$ with $a \in [0, \frac{1}{2}]$. Note that $0 < \epsilon_g, \delta < 1$.

1. If H_k is positive definite, (i.e. the smallest eigenvalue of Hessian satisfies $\lambda_{\min}[H_k] \geq \delta > 0$), we say that f is **locally strongly convex** at x_k, and define the SoS-convex Taylor model as

$$\begin{aligned} m_c(x_k, s) &= T_p(x_k, s) + \frac{\sigma_k}{p'}\sigma_k\|s\|^{p'} \\ &= f_k + g_k^T s + \frac{1}{2}\underbrace{H_k}_{:=\bar{H}_k}[s]^2 + \sum_{j=3}^{p}\frac{1}{j!}\nabla_x^j f(x_k)[s]^j + \frac{\sigma_k}{p'}\|s\|^{p'}. \end{aligned} \tag{6.1}$$

[2]This overall bound matches in accuracy order that for high-order trust-region methods [44], and it is not surprising to obtain this bound here, as we shall see, due to the quadratic regularisation term added in the model in nonconvex iterations.

2. If H_k is not positive definite (i.e. $\lambda_{\min}[H_k] \leq 0$), we say that f is **locally nonconvex** at x_k, and define the SoS-convex Taylor model as

$$\begin{aligned} m_c(x_k, s) &= T_p(x_k, s) + \frac{1}{2}\Big(- \lambda_{\min}[H_k] + \delta \Big)\|s\|^2 + \sigma_k \|s\|^{p+1} \\ &= f_k + g_k^T s + \frac{1}{2}\underbrace{\Big(H_k - \lambda_{\min}[H_k] I_n + \delta I_n\Big)}_{:=\bar{H}_k}[s]^2 + \sum_{j=3}^{p} \frac{1}{j!}\nabla_x^j f(x_k)[s]^j + \frac{\sigma_k}{p'}\|s\|^{p'} \end{aligned} \tag{6.2}$$

3. If H_k is nearly strongly convex, (i.e. $0 < \lambda_{\min}[H_k] \leq \delta$), we say f is **locally nearly strongly convex** at x_k, and define the SoS-convex Taylor model as

$$\begin{aligned} m_c(x_k, s) &= T_p(x_k, s) + \frac{\delta}{2}\|s\|^2 + \frac{\sigma_k}{p'}\|s\|^{p'} \\ &= f_k + g_k^T s + \frac{1}{2}\underbrace{(H_k + \delta I_n)}_{:=\bar{H}_k}[s]^2 + \sum_{j=3}^{p} \frac{1}{j!}\nabla_x^j f(x_k)[s]^j + \frac{\sigma_k}{p'}\|s\|^{p'}. \end{aligned} \tag{6.3}$$

Remark 6.2.1 (A continuous construction of the modified Hessian)**.** *An alternative formulation of Case 3 is to define*

$$\bar{H}_k := H_k - \lambda_{\min}(H_k) I_n + \delta I_n, \qquad 0 < \lambda_{\min}(H_k) < \delta.$$

Under this choice, $\lambda_{\min}(\bar{H}_k) = \delta$. Moreover, this definition connects continuously with the constructions used in Cases 1 and 2. Indeed, when $\lambda_{\min}(H_k) = \delta$, the perturbation vanishes and $\bar{H}_k = H_k$, which coincides with the definition in Case 1. Similarly, when $\lambda_{\min}(H_k) = 0$, it reduces to

$$\bar{H}_k = H_k - \lambda_{\min}(H_k) I_n + \delta I_n,$$

which is the construction used in Case 2. Thus, the resulting modified Hessian may be written compactly as $\bar{H}_k = H_k + \max\{0, \delta - \lambda_{\min}(H_k)\} I_n$, so that its eigenvalues vary continuously with $\lambda_{\min}(H_k)$, provided that H_k itself varies continuously. Both this alternative construction and the formulation adopted in Case 3, $\bar{H}_k := H_k + \delta I_n$, ensure the uniform lower bound $\lambda_{\min}(\bar{H}_k) \geq \delta$. This inequality is the only property required in the convergence and evaluation-complexity analysis in Section 6.4. Consequently, the same convergence orders and complexity bounds remain valid under either construction. We use $\bar{H}_k := H_k + \delta I_n$ throughout this chapter because it yields a simpler perturbation term.

The key idea of the SoS-convex Taylor model with Adaptive Regularization is to choose $\sigma_k > 0$ such that $m_c(x_k, s)$ is SoS-convex. To facilitate this, we provide definitions related to SoS, SoS-matrix, and SoS-convexity, following [1; 7; 102]. Note that SoS polynomial and SoS-convex are defined in Definition 5.2.1 and Definition 5.2.2.

Definition 6.2.1. ***(SoS matrix)*** *A symmetric polynomial matrix $H(s) \in \mathbb{R}[s]^{n \times n}$ is an SoS-matrix if there exists a polynomial matrix $V(s) \in \mathbb{R}[s]^{n_1 \times n}$ for some $n_1 \in \mathbb{N}$, such that $H(s) = V(s)^T V(s)$ [7, Def. 2.4].*

If $p(s)$ is SoS-convex, as per Definition 5.2.2, its Hessian is an SoS-matrix which corresponds to a positive semidefinite matrix (as per Definition 6.2.1). Consequently, if $p(s) : \mathbb{R}^n \to \mathbb{R}$ is SoS-convex, it implies that $p(s)$ is a convex function [7; 102].

Choosing σ_k in the SoS-convex Taylor model

In all three cases, according to [1], $\sigma_k > 0$ is chosen according to the following SoS program.

$$\bar{\sigma}_k := \min \sigma_k, \qquad \text{s.t.} \qquad m_c(x_k, s) \qquad \text{SoS-convex.} \tag{6.4}$$

This problem can be reformulated as a semidefinite program (SDP). For fixed p, the size of this SDP grows polynomially in n. [1, Lemma 3 and Thm 3] and Equation (6.6) here ensure that such $\sigma_k > 0$ exists and is bounded above.

Remark 6.2.2. *The minimum in (6.4) is well-defined. Indeed, for fixed degree, the cone of SoS-convex polynomials is closed, since SoS-convexity is equivalent to the existence of a positive semidefinite Gram matrix satisfying linear coefficient-matching constraints. Hence the feasible set for σ_k is closed. Together with the existence of a finite feasible σ_k guaranteed by [1, Lemma 3 and Theorem 3], this implies that the infimum is attained. Therefore, we write* min *rather than* inf *in (6.4).*

Remark 6.2.3. ***(Role of δ):*** *In [1], the authors set $\delta = \epsilon_g$. Here, we introduce more flexibility by allowing δ to be chosen. Specifically, we set $\delta = {\epsilon_g}^a > 0$ with $a \in [0, \frac{1}{2}]$. We analyze this choice in later sections (Section 6.4, Theorem 6.4.2, and Remark 6.4.2) and discuss how to select δ as a function of ϵ_g. A discussion for the bound $a \geq \frac{1}{2}$ is provided in Remark 6.4.1 and Section A.4.2, respectively.*

Remark 6.2.4. ***(Exponent of regularization)*** *In the ARp framework [23; 41; 44; 51; 53], the Taylor model is typically regularized by $p' = p+1$ order. For even p, the ARp subproblem contains the regularization term $\|s\|^{p+1}$, which is not a polynomial function. As indicated by [1], the SoS-convex Taylor model uses a regularization of order $p' = p+2$ when p is even, to ensure that the model remains a polynomial and, with a suitable choice of σ_k, that the model m_p can be made SoS-convex.*

Remark 6.2.5. ***(Convexity of m_c)*** *Since σ_k chosen by (6.4) makes $m_c(x_k, s)$ SoS-convex, according to Definition 5.2.2, it is natural to deduce that $m_c(x_k, s)$ is convex for all $s \in \mathbb{R}^n$. For instance, if $\bar{H}_k \succeq \delta I_n$ is locally strongly convex, we can select σ_k large enough (depending on δ and Λ_j) to ensure a convex model m_c. However, SoS convex properties, leading to a strict subclass of convex polynomials, provides a tractable means to verify and categorize convexity.*

The minimization algorithm is described in Algorithm 6.2.1. Algorithm 6.2.1 uses the SoS-convex Taylor model as constructed in (6.1), (6.2) or (6.3) depending on the convexity case[3] together with an adaptive regularization framework.

Remark 6.2.6. ***(Minimization of m_c)*** *In Algorithm 6.2.1, we assume that we find a (global) minimizer of m_c, such that $s_k = \operatorname{argmin}_{s \in \mathbb{R}^n} m_c(x_k, s)$. Since m_c is strongly convex and bounded below by construction, the existence and uniqueness of the minimizer are guaranteed. Therefore, any first-order minimizer is also the global minimum. The minimizer can be found by solving a semidefinite program (SDP) of size polynomial in n [1, Thm 2] and can be solved to arbitrary precision within polynomial time. While our proof (in particular Theorem 6.4.1) assumes exact first-order minimization of $\nabla m_c(x_k, s) = 0$, similar bounds can be derived when considering approximate minimization of m_c with a step-termination condition, such as $\|\nabla m_c(x_k, s_k)\| \leq \theta \|s_k\|^{p'-1}$ for some $\theta \in (0, 1)$. Such step-termination conditions are also used in [20; 23; 44; 53], with alternative variants available [46].*

[3] For more on the choice of a which dictates the convexity case, see Remark 6.4.2 and Remark 6.4.3.

Algorithm 6.2.1: Adaptive pth-order Regularization Method with SoS-convex Taylor Models (ARp+SoS Convex)

Initialization: Initialize $x_0 = \mathbf{0} \in \mathbb{R}^n$, $k = 0$, $\sigma_k^r = 0$, $\sigma_{\min} > 0$, $\sigma_0^r = \sigma_{\min}$, $\eta > 0$, $\gamma_1 > 1 > \gamma_2 > 0$.

Input: Provided an order $p \geq 3$, set $p' := p + 1$ for odd p and $p' := p + 2$ for even p. Derivatives at $x_0 = \mathbf{0}$, $\{f(x_0), \nabla_x f(x_0), \nabla_x^2 f(x_0), \ldots, \nabla_x^p f(x_0)\}$, an accuracy level ϵ such that $0 < \epsilon < 1$, a and $\delta = \epsilon^a > 0$.

Step 1: Test for termination. If $\|\nabla_x f(x_k)\| \leq \epsilon_g$, terminate with the ϵ_g-approximate first-order minimizer $s_{\epsilon_g} = s_k$.

Step 2: Step computation.

Compute $\lambda_{\min}[\nabla_x^2 f(x_k)]$ to determine the convexity case. Construct m_c as (6.1), (6.2) or (6.3) depending on the convexity case. Compute $\bar{\sigma}_k$ using (6.4). Set $\sigma_k = \max\{\bar{\sigma}_k, \sigma_k^r\}$.

Compute $s_k = \operatorname{argmin}_{s \in \mathbb{R}^n} m_c(x_k, s)$.

Step 3: Acceptance of trial point. Compute $T_p(x_k, s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_p(x_k, s_k)}. \qquad \text{(Ratio Test)}$$

Step 4: Regularization parameter update. if $\rho_k > \eta$ **then**

  Successful iteration. $x_{k+1} := x_k + s_k$, decrease regularization $\sigma_{k+1}^r = \max\{\gamma_2 \sigma_k, \sigma_{\min}\}$.

else

  Unsuccessful iteration. $x_{k+1} := x_k$, increase regularization $\sigma_{k+1}^r = \gamma_1 \sigma_k$.

$k := k + 1$

6.3 An Upper Bound on the Regularization Parameter

In this section, we show that there exists an iteration independent uniform upper bound for $\sigma_k = \max\{\bar{\sigma}_k, \sigma_k^r\}$. Note that the parameter $\bar{\sigma}_k$ that is computed in (6.4) can be considered as the smallest possible value such that $m_c(x_k, s)$ is SoS-convex, making the sub-problem tractable, while σ_k^r is the regularization parameter generated in Step 4 of the adaptive regularization framework (Algorithm 6.2.1) that monitors sufficient decrease and overall progress of the algorithm towards convergence. To prove this result, we first introduce the following definitions, assumptions, and technical Lemmas.

Assumption 6.3.1. *$f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$ which means that f is p-times continuously differentiable, bounded below by a constant f_{low} and the pth derivative of f is globally Lipschitz continuous. Namely, there exists a constant $L > 0$ such that, for all $x, y \in \mathbb{R}^n$, $\|\nabla_x^p f(x) - \nabla_x^p f(y)\| \leq (p-1)!L\|x - y\|$.*

Note that we denote L_p in (1.5)–(1.6) by L in this chapter for notational simplicity. Assumption 6.3.1 is a standard assumption in the analysis of algorithms that utilize high-order derivative information [23; 45; 46; 53], see Lemma 1.2.1.

Definition 6.3.1. *Let $T \in \mathbb{R}^{n^j}$ be a jth-order tensor. Following [53], the tensor norm of T is*

$$\|T\|_{[j]} := \max_{\|v_1\| = \ldots = \|v_j\| = 1} \left|T[v_1, \ldots, v_j]\right|$$

where $\|\cdot\|$ is the Euclidean norm and $v_1, \ldots, v_j \in \mathbb{R}^n$ are vectors. Note that $T[v_1, \ldots, v_j]$ stands for applying the jth-order tensor T to the vectors $v_1, \ldots, v_j$.

Assumption 6.3.2. *We assume that the tensor norms $\left\|\nabla_x^j f(x_k)\right\|_{[j]}$ are uniformly bounded at all iterates x_k, such that*

$$\left\|\nabla_x^j f(x_k)\right\|_{[j]} := \max_{\|v_1\| = \ldots = \|v_j\| = 1} \left|\nabla_x^j f(x_k)[v_1, \ldots, v_j]\right| \leq \Lambda_j,$$

for all $k \geq 0$ and $j = 1, \ldots, p$.

Note that Assumption 6.3.1 is a standard assumption in the analysis of algorithms that utilize high-order derivative information [23; 45; 46; 53]. The proof and analysis only require the pth derivative of f to be globally Lipschitz continuous, and do not require f or its lower-order derivatives to have similar smoothness properties. Lastly, this assumption can be relaxed by assuming a weaker but more abstract property that the pth derivative of f is Lipschitz continuous on the path of generated by all the iterates (both successful and unsuccessful); more details can be found in Section 2.4 and Section A.1.1. We also note that the existence of a lower bound of f implies (through our analysis or standard ones) that *approximate* critical points of f exist [71].
Note that Assumption 6.3.2 is only needed for iterates x_k produced by the algorithm.

Definition 6.3.2. *Let $q(s) : \mathbb{R}^n \to \mathbb{R}$ be a polynomial. Following [8], we define $\|q(s)\|_{\infty^*}$ as the largest absolute value of the coefficients of $q(s)$ in the standard monomial basis.*

Note that Assumption 6.3.2 is only needed for iterates x_k produced by the algorithm. Assumption 6.3.2 is satisfied if all derivatives of f up to order $p+1$ are globally Lipschitz continuous (this may be imposed as an a priori smoothness assumption). Indeed, if the i-th derivative is globally Lipschitz, $\|\nabla^i f(x) - \nabla^i f(y)\| \leq (i-1)! L_i \|x - y\|$, then the $(i+1)$-st derivative is globally bounded, $\|\nabla^{i+1} f(x)\| \leq L_i$ for all x. This guarantees boundedness of all required higher-order derivatives. However, this is a stronger and more restrictive assumption, and thus we prefer to keep Assumption 6.3.2 in its current form, which is a weaker and more technical assumption.

Lemma 6.3.1. ***(Related norms)*** *Under Assumption 6.3.2, the following statements hold.*

1. *The coefficients of $\frac{1}{j!}\nabla_x^j f(x_k)[s]^j$ in the standard monomial basis at the iterates x_k are bounded, such that $\left\|\frac{1}{j!}\nabla_x^j f(x_k)[s]^j\right\|_{\infty^*} \leq \Lambda_j$.*
2. *The leftmost eigenvalue of the second-order derivatives of f at the iterates x_k is bounded, such that $\max_k \left|\lambda_{\min}[\nabla_x^2 f(x_k)]\right| \leq \Lambda_2$.*

Proof. The proof of Lemma 6.3.1 can be found in Section A.4.1 □

The subsequent Lemma and assumption can be used to prove that σ_k generated by the SoS-convex paradigm remains bounded regardless of the number of iterations.

Lemma 6.3.2. *([1, Lemma 2]) Let $q : \mathbb{R}^n \to \mathbb{R}$ be a multivariate polynomial expressed in the standard monomial basis and $\|\cdot\|_{\infty^*}$ be defined as in Definition 6.3.2. There exists a constant[4] $R > 0$, such that for any polynomial $q(s)$ of degree at most p with $\|q\|_{\infty^*} \leq R$, the polynomial $\|s\|^2 + \|s\|^{p'} + q(s)$ is SoS-convex, where $p' := p+1$ if p is odd and $p' := p+2$ if p is even.*

Remark 6.3.1. *[1, Lemma 2] gives that for an even integer p', $\|s\|^2 + \|s\|^{p'}$ lies within the interior of the cone of SoS-convex polynomials in n variables and of degree at most p'. This result implies Lemma 6.3.2. Importantly, R in Lemma 6.3.2 is a constant that is independent of the iteration k, ϵ_g, and δ.*

To demonstrate that $\sigma_k = \max\{\bar{\sigma}_k, \sigma_k^r\}$ is uniformly bounded above regardless of the iteration k, we outline our proof in two steps, with some numerical illustration on the size of σ_k are presented in Section 6.5.

1. In Section 6.3.1, we first show that there exists a uniform upper bound for $\bar{\sigma}_k$ such that $m_c(x_k, s)$ is SoS-convex. Note that in [1], only the strongly convex case and the locally nonconvex case are discussed. We extend this result to include the locally nearly strongly convex case, so that all possible cases are now addressed. We also derive an explicit expression for the upper bound on $\bar{\sigma}_k$ (Corollary 6.3.1).
2. Then, in Section 6.3.2, we show that there exists a uniform upper bound on σ_k^r to ensure that the next iteration gives a sufficient decrease in f (Theorem 6.3.2 for odd p and Theorem 6.3.2 for even p).

[4] R is a bound for the largest absolute value of the coefficients of $q(s)$ and is independent of p.

6.3.1 A Uniform Upper Bound on $\bar{\sigma}_k$

In this subsection, we establish a uniform upper bound on $\bar{\sigma}_k$.

Theorem 6.3.1. *([1, Theorem 3][5]) Let $C_{k,p}(s) := \sum_{j=3}^{p} \frac{1}{j!}\nabla_x^j f(x_k)[s]^j$ and let $M_k := \|C_{k,p}(s)\|_{\infty^*}$ where $\|\cdot\|_{\infty^*}$ is defined in Definition 6.3.2. Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ and $\bar{\sigma}_k$ is computed using (6.4) in Algorithm 6.2.1. Then,*

$$\frac{\bar{\sigma}_k}{p'} \leq \frac{\delta}{2}\min\left\{1, \frac{2M_k}{\delta R}\right\}^{p'-2} \tag{6.5}$$

where R is defined in Lemma 6.3.2 and is a constant independent of the iteration k, ϵ_g, and δ.

Corollary 6.3.1. ***(Bound for $\bar{\sigma}_k$)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ and $\bar{\sigma}_k$ is computed using (6.4) in Algorithm 6.2.1. Under Assumption 6.3.2, in all convexity cases and for all $k \geq 0$,*

$$C_R = \frac{p'}{2}\left(\sum_{j=3}^{p}\frac{2\Lambda_j}{R}\right)^{p'-2}. \tag{6.6}$$

Note that C_R is independent of the iteration count k, the tolerance ϵ_g, and the perturbation δ.

Proof. Using Assumption 6.3.2 and Lemma 6.3.1, we deduce that

$$M_k = \|C_{k,p}(s)\|_{\infty^*} \leq \sum_{j=3}^{p}\left\|\frac{1}{j!}\nabla_x^j f(x_k)[s]^j\right\|_{\infty^*} \leq \sum_{j=3}^{p}\Lambda_j. \tag{6.7}$$

In the locally strongly convex case, we have $\bar{H}_k := H_k \succeq \delta I_n$. Similarly, in the locally nonconvex case, we have $\bar{H}_k := H_k - \lambda_{\min}[H_k]I_n + \delta I_n \succeq \delta I_n$, and in the locally nearly strongly convex case, we have $\bar{H}_k := H_k + \delta I_n \succeq \delta I_n$. Therefore, we can apply Theorem 6.3.1 in all three cases. Substituting (6.7) into (6.5), we obtain that

$$\frac{\bar{\sigma}_k}{p'} \leq \frac{\delta}{2}\min\left\{1, \frac{(2/\delta)M_k}{R}\right\}^{p'-2} \leq \max\left\{\frac{\delta}{2}, \quad \left(\sum_{j=3}^{p}\frac{\Lambda_j}{R}\right)^{p'-2}\left(\frac{\delta}{2}\right)^{3-p'}\right\}.$$

Setting C_R as stated in the lemma, we deduce the required bound. □

6.3.2 A Uniform Upper Bound on σ_k^r

We need to use two technical lemmas. The first Lemma is introduced in Lemma 1.2.1. For notation simplicity in this chapter, we use L to denote L_p. The second Lemma is given as follows.

Lemma 6.3.3. *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$. The mechanism of Algorithm 6.2.1 guarantees that, for all $k \geq 0$,*

$$\textit{1: L ocally strongly convex: } f(x_k) - T_p(x_k, s_k) \geq \frac{\sigma_k}{p'}\|s_k\|^{p'}, \tag{6.8}$$

$$\textit{2: Locally nonconvex: } f(x_k) - T_p(x_k, s_k) \geq \big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\|^2 + \frac{\sigma_k}{p'}\|s_k\|^{p'}, \tag{6.9}$$

$$\textit{3: Locally nearly strongly convex: } f(x_k) - T_p(x_k, s_k) \geq \delta\|s_k\|^2 + \frac{\sigma_k}{p'}\|s_k\|^{p'}. \tag{6.10}$$

Proof. Since $s_k = \operatorname{argmin}_{s\in\mathbb{R}^n} m_c(x_k, s)$, we have $m_c(x_k, 0) - m_c(x_k, s_k) \geq 0$. Thus,

$$f(x_k) - T_p(x_k, s_k) = \underbrace{\big[f(x_k) - m_c(x_k, s_k)\big]}_{\geq 0} + \big[m_c(x_k, s_k) - T_p(x_k, s_k)\big] \geq m_c(x_k, s_k) - T_p(x_k, s_k).$$

- In locally strongly convex cases, according (6.1), $m_c(x_k, s_k) - T_p(x_k, s_k) = \frac{\sigma_k}{p'}\|s_k\|^{p'}$.

[5] In [1], slightly different notations are employed, where the cubic and higher terms are defined as $C(s) := \frac{2}{\delta}\sum_{j=3}^{p}\frac{1}{j!}\nabla_x^j f(x_k)[s]^j$ and $M = (2/\delta)M_k$.

- In locally nonconvex cases, according (6.2), $m_c(x_k, s_k) - T_p(x_k, s_k) = \big(-\lambda_{\min}[H_k] + \delta \big)\|s_k\|^2 + \frac{\sigma_k}{p'}\|s_k\|^{p'}$.
- In locally nearly strongly convex cases, according to (6.3), $m_c(x_k, s_k) - T_p(x_k, s_k) = \delta\|s_k\|^2 + \frac{\sigma_k}{p'}\|s_k\|^{p'}$.

□

Lemma 6.3.4. ***(Upper bound for σ_k^r)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$. Under Assumption 6.3.1, if*

$$\sigma_k^r \geq \max\big\{\sigma_0, \quad A_0\|s_k\|^{-\iota}\big\}, \tag{6.11}$$

where $A_0 = \frac{Lp'}{(p+1)(1-\eta)}$, $\iota = 0$ for odd p, and $\iota = 1$ for even p, then, $\rho_k \geq \eta$ and (Ratio Test) for iteration k is successful. Note that[6] if p is odd, (6.11) becomes $\sigma_k \geq \max\big\{\sigma_0, \frac{\gamma_1 L}{(1-\eta)}\big\}$; if p is even, (6.11) becomes $\sigma_k \geq \max\big\{\sigma_0, \frac{\gamma_1 L(p+2)}{(p+1)(1-\eta)\|s_k\|}\big\}$.

Proof. The proof follows a similar setup as [53, Lemma 3.2], [52, Lemma 3.4] and [23, Lemma 2.2]. From (6.8)–(6.10), we deduce that in all three cases $f(x_k) - T_p(x_k, s_k) \geq \frac{\sigma_k}{p'}\|s_k\|^{p'}$. Also, using (1.5), we may then deduce that

$$|\rho_k - 1| \leq \frac{|f(x_k + s_k) - T_p(x_k, s_k)|}{|f(x_k) - T_p(x_k, s_k)|} \leq \frac{Lp'\|s_k\|^{p+1}}{(p+1)\sigma_k\|s_k\|^{p'}} \leq \frac{Lp'}{(p+1)\sigma_k}\|s_k\|^{p+1-p'}.$$

If $\sigma_k \geq \frac{Lp'}{(p+1)(1-\eta)}\|s_k\|^{p+1-p'}$, we deduce that $|\rho_k - 1| \leq 1 - \eta$ and thus $\rho_k \geq \eta$. Then, iteration k is successful. □

Theorem 6.3.2. ***(Uniform upper bound for σ_k when p is odd.)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ and Assumptions 6.3.1–6.3.2 hold. If $p \geq 3$ is odd and $p' = p + 1$, for all $k \geq 0$,*

$$\sigma_k \leq \max\big\{\sigma_0, \quad \gamma_1 A_0, \quad p'\frac{\delta}{2}, \quad C_R\delta^{3-p'}\big\}. \tag{6.12}$$

where[a] $A_0 := \frac{Lp'}{(p+1)(1-\eta)}$. $\sigma_0, \gamma_1, L, p, \eta, C_R, A_0$ are independent of the tolerance ϵ_g, and $\delta = {\epsilon_g}^a$ with $a \in [0, \frac{1}{2}]$. Moreover, for any $0 < \epsilon_g < 1$, we have

$$\sigma_k \leq \hat{C}_R {\epsilon_g}^{a(2-p)} =: \sigma_{\max} \tag{6.13}$$

where $\hat{C}_R = \max\big\{\sigma_0, \gamma_1 A_0, p'\frac{\delta}{2}, C_R\big\}$. Note that $C_R = \frac{p'}{2}(\sum_{j=3}^{p}\frac{2\Lambda_j}{R})^{p'-2}$ as defined in (6.6).

[a] In the general ARp framework [51; 53], we have $p' = p + 1$. Also, we do not need the upper bound in (6.6), and so $\sigma_k \leq \max\big\{\sigma_0, \frac{\gamma_1 L}{(1-\eta)}\big\}$.

Proof. Assume that $p \geq 3$ is odd and $p' = p + 1$. To prove the first result, assume $\sigma_k \geq \frac{\gamma_1 Lp'}{(p+1)(1-\eta)}$, according to Lemma 6.3.4, we deduce that iteration k is successful and $\sigma_{k+1}^r \leq \sigma_k$. As a consequence, the mechanism of the algorithm ensures that

$$\sigma_k^r \leq \max\left\{\sigma_0, \frac{Lp'\gamma_1}{(p+1)(1-\eta)}\right\}$$

for all k. Combining with the bound (6.6) in Corollary 6.3.1, we have $\sigma_k = \max\{\bar{\sigma}_k, \sigma_k^r\}$ which gives (6.12). To prove the second result, using (6.12) and $\hat{C}_R = \max\big\{\sigma_0, \gamma_1 A_0, p'\frac{\delta}{2}, C_R\big\}$, we have

$$\sigma_k \leq \hat{C}_R \max\big\{1, \delta^{3-p'}\big\} = \hat{C}_R \max\big\{1, {\epsilon_g}^{(2-p)a}\big\}. \tag{6.14}$$

By considering $0 < \epsilon_g < 1$, we obtain (6.13). □

[6] (6.11) matches the bound in the universal ARPr framework ([52, Lemma 3.4]) by setting the parameter in [52] as $\beta_p = 1$ and $r = p + 1$ for odd p and $r = p + 2$ for odd p.

Lemma 6.3.5. *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ and Assumptions 6.3.1–6.3.2 hold. If $p \geq 4$ is even and $p' = p+2$, for all $k \geq 0$ and any $0 < \epsilon_g < 1$,*

$$\sigma_k \geq \max\{\sigma_0, B{\epsilon_g}^{-1}\}, \tag{6.15}$$

where $B = 4A_0 \max\left\{1, (\Lambda_2+1), (L+A_0)^{\frac{1}{p}}\right\}$, then (6.11) holds, $\rho_k \geq \eta$ and (Ratio Test) for iteration k is successful.

Proof. We will prove our result by contradiction. We assume (6.11) does not hold on iteration k, such that

$$\sigma_k \|s_k\| < \frac{Lp'}{(p+1)(1-\eta)} = A_0. \tag{6.16}$$

Note that while Algorithm 6.2.1 does not terminate, we have

$$\begin{aligned} \epsilon_g &< \left\|\nabla_x f(x+s_k)\right\| = \left\|\nabla_x f(x+s_k) - \nabla m_c(x_k, s_k)\right\| \\ &\leq \left\|\nabla_x f(x+s_k) - \nabla T_p(x, s_k)\right\| + \left\|\nabla m_c(x_k, s_k) \ - \nabla T_p(x, s_k)\right\| \\ &\underset{(1.6)}{\leq} L\|s_k\|^p + \left\|\nabla m_c(x_k, s_k) \ - \nabla T_p(x, s_k)\right\|. \end{aligned} \tag{6.17}$$

Using (6.8)–(6.10) and (6.16), (6.17) becomes[7]

$$\text{Case 1:} \quad \epsilon_g < L\|s_k\|^p + \sigma_k\|s_k\|\|s_k\|^p \underset{(6.16)}{\leq} (L+A_0)\|s_k\|^p, \tag{6.18}$$

$$\text{Case 2:} \quad \epsilon_g < \big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\| + L\|s_k\|^p + \sigma_k\|s_k\|\|s_k\|^p \leq \big(\Lambda_2+1\big)\|s_k\| + (L+A_0)\|s_k\|^p, \tag{6.19}$$

$$\text{Case 3:} \quad \epsilon_g < \delta\|s_k\| + L\|s_k\|^p + \sigma_k\|s_k\|\|s_k\|^p \leq {\epsilon_g}^a\|s_k\| + (L+A_0)\|s_k\|^p, \tag{6.20}$$

where the last inequality in Case 2 is obtained by Lemma 6.3.1 and $0 < \delta = {\epsilon_g}^a < 1$. From (6.18), we deduce that $\|s_k\| > \left(\frac{\epsilon_g}{L+A_0}\right)^{\frac{1}{p}}$. From (6.19), we deduce that either $\frac{\epsilon_g}{2} < (L+A_0)\|s_k\|^p$ or $\frac{\epsilon_g}{2} < \big(\Lambda_2+1\big)\|s_k\|$. Therefore, $\|s_k\| > \frac{1}{2}\min\left\{\frac{\epsilon_g}{\Lambda_2+1}, \left(\frac{\epsilon_g}{L+A_0}\right)^{\frac{1}{p}}\right\}$. Similarly, we deduce from (6.20) that $\|s_k\| > \frac{1}{2}\min\left\{{\epsilon_g}^{1-a}, \left(\frac{\epsilon_g}{L+A_0}\right)^{\frac{1}{p}}\right\}$. Combining these bounds, we obtain that in all iterations, $\|s_k\|$ has the following lower bound,

$$\|s_k\| > \frac{1}{2}\min\left\{{\epsilon_g}^{1-a}, \quad \frac{\epsilon_g}{\Lambda_2+1}, \quad \left(\frac{\epsilon_g}{L+A_0}\right)^{\frac{1}{p}}\right\} \geq B_1 \min\left\{{\epsilon_g}^{1-a}, \epsilon_g, {\epsilon_g}^{\frac{1}{p}}\right\} = B_1\epsilon_g.$$

where $B_1 = \frac{1}{2}\min\left\{1, (\Lambda_2+1)^{-1}, (L+A_0)^{-\frac{1}{p}}\right\}$ is a constant. The last inequality comes from $\epsilon_g \leq {\epsilon_g}^{1-a}$ and $\epsilon_g \leq {\epsilon_g}^{\frac{1}{p}}$ for $0 < \epsilon_g < 1$ and $a > 0$. On the other hand, by (6.16), we have also $\|s_k\| \leq \frac{A_0}{\sigma_k}$. Therefore,

$$\frac{A_0}{\sigma_k} > B_1\epsilon_g \qquad \Rightarrow \qquad \sigma_k < 2A_0B_1^{-1}{\epsilon_g}^{-1}$$

which contradicts (6.15). Thus, (6.11) must hold and Lemma 6.3.4 implies that k is successful. □

Theorem 6.3.3. ***(Upper bound for σ_k when p is even.)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ and Assumptions 6.3.1–6.3.2 hold. If $p \geq 4$ is even and $p' = p+2$, for all $k \geq 0$,*

$$\sigma_k \leq \max\left\{\sigma_0, \quad \gamma_1 B{\epsilon_g}^{-1}, \quad p'\frac{\delta}{2}, \quad C_R\delta^{3-p'}\right\} \tag{6.21}$$

where $\gamma_1, L, p, \eta, C_R, B$ are independent of the tolerance ϵ_g, and $\delta = {\epsilon_g}^a$ with $a \in [0, \frac{1}{2}]$. More-

[7] In the universal ARPr framework [52], the framework uses the Taylor model with no perturbation term, i.e., $\delta = 0$. Therefore, (6.18)–(6.20) become $\epsilon_g < (L+A_0)\|s_k\|^p$ and we have $\sigma_k \geq \mathcal{O}({\epsilon_g}^{-\frac{1}{p}})$ [52, Lemma 3.4].

over, for any $0 < \epsilon_g < 1$, *we have*

$$\sigma_k \leq \tilde{C}_R \max\left\{{\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)}\right\} = \tilde{C}_R {\epsilon_g}^{-\max\{1,a(1-p)\}} =: \sigma_{\max} \tag{6.22}$$

where $\tilde{C}_R = \max\{\sigma_0, \gamma_1 B, p'\frac{\delta}{2}, C_R\}$. *Note that if* $a \in [0, \frac{1}{p-1}]$, *then* $\sigma_k \leq \tilde{C}_R {\epsilon_g}^{-1}$. *If* $a \in [\frac{1}{p-1}, \frac{1}{2}]$, *then* $\sigma_k \leq \tilde{C}_R {\epsilon_g}^{a(1-p)}$.

Proof. Assume that $p \geq 4$ is even and $p' = p + 2$. To prove the first result, assume $\sigma_k \geq \gamma_1 B {\epsilon_g}^{-1}$, according to Lemma 6.3.5, we deduce that iteration k is successful and $\sigma_{k+1}^r \leq \sigma_k$. As a consequence, the mechanism of the algorithm ensures that $\sigma_k^r \leq \gamma_1 B {\epsilon_g}^{-1}$ for all k. Combining with the bound (6.6) in Corollary 6.3.1, we have $\sigma_k = \max\{\bar{\sigma}_k, \sigma_k^r\}$ which gives (6.21). To prove the second result, we start from (6.21), which gives us $\sigma_k \leq \max\left\{\sigma_0, \gamma_1 B {\epsilon_g}^{-1}, p'\frac{\delta}{2}, C_R {\epsilon_g}^{(3-p')a}\right\}$. Let $\tilde{C}_R = \max\{\sigma_0, \gamma_1 B, p'\frac{\delta}{2}, C_R\}$. Then, we have

$$\sigma_k \leq \tilde{C}_R \max\left\{1, {\epsilon_g}^{-1}, \delta^{3-p'}\right\} = \tilde{C}_R \max\left\{1, {\epsilon_g}^{-1}, {\epsilon_g}^{(1-p)a}\right\}. \tag{6.23}$$

By considering $0 < \epsilon_g < 1$, we obtain (6.22). □

6.4 Convergence and Complexity Analysis

In this section, we prove that Algorithm 6.2.1 generates a monotonically decreasing sequence of function values, $\{f(x_k)\}_{k\geq 0}$, and that $\lim_{k\to\infty} \inf \|\nabla f(x_k)\| = 0$ (i.e., at least one limit points of x_k is a stationary point of $f(x)$). We show that, in strongly convex iterations, Algorithm 6.2.1 achieves a function value decrease that is at least of order $\mathcal{O}({\epsilon_g}^{\frac{p+1}{p}})$ for odd $p > 5$, and $\mathcal{O}({\epsilon_g}^{\frac{p+3}{p+1}})$ for even $p > 5$. Note that we have improved function value decrease for $p = 3, 5$ at $\mathcal{O}({\epsilon_g}^{\frac{2}{p}})$ and for $p = 4$ at $\mathcal{O}({\epsilon_g}^{\frac{4}{5}})$. In locally nonconvex cases, the objective reduction per iteration is at least $\mathcal{O}({\epsilon_g}^2)$. When f is globally strongly convex, we establish an enhanced evaluation complexity bound for the ϵ_g-approximate minimum value of convex functions as $\mathcal{O}({\epsilon_g}^{-\frac{1}{p}})$ for odd p, matching the result in [1; 137].

Algorithm 6.2.1 retains the p-th order local rate of convergence established in [1] and [67] for odd p. This follows from the fact that, when the iterates are sufficiently close to a non-degenerate minimizer, the iterations become strictly locally convex (i.e., $\nabla_x^2 f(x_k) \succ \delta I_n$). In this regime, the SoS-convex Taylor model coincides with the ARp subproblem for odd degrees. Although we do not explicitly provide a proof of the p-th order local rate here, the proof and analysis can be found in [1, Thm 5] where a local convergence rate of order p is established. The global complexity bound in the case where $\nabla^2 f(x_k) \succ \delta$ for all k also matches the global complexity bound established in [1]. Note that the overall bound of $O(\epsilon_g^{-2})$ arises due to the locally nonconvex iterations. To make the subproblem tractable in such cases, we introduce the perturbation term $-\lambda_{\min}[H_k]I_n + \delta I_n$, which ensures local convexity (and global model convexity when the regularization is sufficiently large). However, this modification causes the model m_c to deviate from the original Taylor expansion at the Hessian level, which worsens the complexity bound. Despite this shortcoming, the added value of our higher-order Newton algorithm lies in the fact that it provides a globally convergent algorithm with polynomial computational cost per iteration, even for nonconvex functions, which we expect may be improved later by various techniques, with substantial algorithm modifications (that depart from the SoS-convex framework). For instance, in locally nonconvex cases, instead of adding $-\lambda_{\min}[H_k]I_n + \delta I_n$ to the Hessian, one could use the negative curvature step strategy outlined in Algorithm 4.1 in [89]. Alternatively, the QQR algorithm Algorithm 2.3.1 could be used for local nonconvex iterations, which yields a function value reduction of $O(\epsilon)$ per iteration. Even with these improvements, it may not be possible (at least not to our knowledge) to obtain the optimal bound for ARp methods (that depends on p), which remains a topic for future work.

Our convergence and complexity analysis use standard techniques similar to those in [45; 51; 53]. Firstly, we establish lower bounds on model decrease and step length at each iteration. Note that ensuring a lower bound on the step is crucial in guaranteeing function value reduction, incorporating

techniques from [23, Lemma 2.3] and [53, Lemma 3.3]. Then, we construct the lower bound for function value reduction using the relationship between model decrease characterized by (Ratio Test). While standard techniques offer valuable insights, our innovation lies in their application to the SoS-convex Taylor model across various convexity scenarios. To the best of our knowledge, we have not encountered this type of global complexity results for a tractable cubic or higher-order sub-problem in nonconvex smooth optimization.

The section is structured as follows. In Section 6.4.1, we present the function value reductions in successful iterations of Algorithm 6.2.1 for three different convexity cases. We analyze the impact of choosing δ as a function of ϵ_g on the bound for function decrease. Additionally, improved bounds for function value reduction for $3 \leq p \leq 5$ are provided in Sections 6.4.1.1 and 6.4.1.2, respectively. Subsequently, in Section 6.4.2, we provide the overall complexity bound for Algorithm 6.2.1 across three convexity scenarios counting both successful and unsuccessful iterations. Finally, Section 6.4.2.1 outlines the improved overall complexity bound for a strongly convex objective function f.

6.4.1 Bounding Objective Decrease in Successful Iterations

Theorem 6.4.1. ***(Lower bound on the step size)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Let $\delta = {\epsilon_g}^a$ for any $0 < \epsilon_g < 1$ and $a \in [0, \frac{1}{2}]$. Under Assumptions 6.3.1–6.3.2, for all $k \geq 0$ and any $0 < \epsilon_g < 1$,*

$$\textit{Case 1:} \qquad \|s_k\| > \frac{1}{2} \min\left\{ \left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}, \left(\frac{\epsilon_g}{\sigma_k}\right)^{\frac{1}{p'-1}} \right\}, \tag{6.24}$$

$$\textit{Case 2:} \qquad \|s_k\| > \hat{\kappa}_3 \epsilon_g, \tag{6.25}$$

$$\textit{Case 3:} \qquad \|s_k\| > \hat{\kappa}_4 {\epsilon_g}^{1-a}, \tag{6.26}$$

where $\hat{\kappa}_3$ and $\hat{\kappa}_4$ are iteration independent and ϵ_g independent constants.

Proof. Using the same deduction as in (6.17), we arrive at $\epsilon_g < L\|s_k\|^p + \left\|\nabla m_c(x_k, s_k) - \nabla T_p(x, s_k)\right\|$. Using (6.8)–(6.10),(6.13), (6.22) and (6.16), we have

$$\text{Case 1:} \qquad \epsilon_g < L\|s_k\|^p + \sigma_k \|s_k\|^{p'-1}, \tag{6.27}$$

$$\text{Case 2:} \qquad \epsilon_g < \big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\| + L\|s_k\|^p + \sigma_k \|s_k\|^{p'-1}, \tag{6.28}$$

$$\text{Case 3:} \qquad \epsilon_g < {\epsilon_g}^a \|s_k\| + L\|s_k\|^p + \sigma_k \|s_k\|^{p'-1}. \tag{6.29}$$

Case 1 (locally strongly convex): Using (6.27), we deduce that $\|s_k\| > \frac{1}{2} \min\left\{ \left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}, \left(\frac{\epsilon_g}{\sigma_k}\right)^{\frac{1}{p'-1}} \right\}$.

Case 2 (locally nonconvex): Using (6.28) and $\sigma_k \leq \sigma_{\max}$, we deduce $\|s_k\| > \frac{1}{3} \min\left\{ \frac{\epsilon_g}{-\lambda_{\min}[H_k]+\delta}, \left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}, \left(\frac{\epsilon_g}{\sigma_{\max}}\right)^{\frac{1}{p'-1}} \right\}$. Let $\hat{\kappa}_3 := \frac{1}{3} \min\left\{ (\Lambda_2 + 1)^{-1}, L^{\frac{1}{p}}, \hat{C}_R^{-\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}} \right\}$.

- If p is odd, $p' = p + 1$, by (6.13), $\sigma_{\max} := \hat{C}_R {\epsilon_g}^{a(2-p)}$, then

$$\|s_k\| > \underbrace{\frac{1}{3} \min\left\{ (-\lambda_{\min}[H_k] + \delta)^{-1}, L^{\frac{1}{p}}, \hat{C}_R^{-\frac{1}{p}} \right\}}_{\geq \hat{\kappa}_3} \min\left\{ \epsilon_g, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{a(p-2)+1}{p}} \right\} \geq \hat{\kappa}_3 \epsilon_g$$

  where the last inequality comes from $1 \geq \frac{a(p-2)+1}{p} \geq \frac{1}{p}$ for all $p \geq 3$ and $a \in [0, \frac{1}{2}]$ and $\min\left\{ \epsilon_g, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{a(p-2)+1}{p}} \right\} = \epsilon_g$ for $0 < \epsilon_g < 1$.

- If p is even, $p' = p + 2$, by (6.22), $\sigma_{\max} := \tilde{C}_R \max\left\{ {\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)} \right\}$, then

$$\|s_k\| > \underbrace{\frac{1}{3} \min\left\{ (-\lambda_{\min}[H_k] + \delta)^{-1}, L^{\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}} \right\}}_{\geq \hat{\kappa}_3} \min\left\{ \epsilon_g, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{\max\{a(p-1),1\}+1}{p+1}} \right\} \geq \hat{\kappa}_3 \epsilon_g$$

  where the last inequality comes from $1 \geq \frac{\max\{a(p-1),1\}+1}{p+1}$ and $1 \geq \frac{1}{p}$ for all $p \geq 4$ and $a \in [0, \frac{1}{2}]$ and $\min\left\{ \epsilon_g, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{\max\{a(p-1),1\}+1}{p+1}} \right\} = \epsilon_g$ for $0 < \epsilon_g < 1$.

Case 3 (locally nearly strongly convex): Using (6.29) and $\sigma_k \in \sigma_{\max}$, we deduce $\|s_k\| > \frac{1}{3}\min\left\{{\epsilon_g}^{1-a}, \left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}, \left(\frac{\epsilon_g}{\sigma_{\max}}\right)^{\frac{1}{p'-1}}\right\}$. Let $\hat{\kappa}_4 := \frac{1}{3}\min\left\{1, L^{\frac{1}{p}}, \hat{C}_R^{-\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}\right\}$.

- If p is odd, $\sigma_{\max} = \hat{C}_R {\epsilon_g}^{a(2-p)}$, then

$$\|s_k\| > \underbrace{\frac{1}{3}\min\left\{1, L^{\frac{1}{p}}, \hat{C}_R^{-\frac{1}{p}}\right\}}_{\geq \hat{\kappa}_4} \min\left\{{\epsilon_g}^{1-a}, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{a(p-2)+1}{p}}\right\} \geq \hat{\kappa}_4 {\epsilon_g}^{1-a} \tag{6.30}$$

  where the last inequality comes from $1-a \geq \frac{a(p-2)+1}{p}$ and $1-a \geq \frac{1}{p}$ for all $p \geq 3$ and $\min\left\{{\epsilon_g}^{1-a}, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{a(p-2)+1}{p}}\right\} = {\epsilon_g}^{1-a}$ for $0 < \epsilon_g < 1$.

- If p is even, $\sigma_{\max} := \tilde{C}_R \max\left\{{\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)}\right\}$, then

$$\|s_k\| > \underbrace{\frac{1}{3}\min\left\{1, L^{\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}\right\}}_{\geq \hat{\kappa}_4} \min\left\{{\epsilon_g}^{1-a}, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{\max\{a(p-1),1\}+1}{p+1}}\right\} \geq \hat{\kappa}_4 {\epsilon_g}^{1-a} \tag{6.31}$$

  where the last inequality comes from $1-a \geq \frac{\max\{a(p-1),1\}+1}{p+1}$ and $1-a \geq \frac{1}{p}$ for all $p \geq 4$ and $a \in [0, \frac{1}{2}]$ and $\min\left\{{\epsilon_g}^{1-a}, {\epsilon_g}^{\frac{1}{p}}, {\epsilon_g}^{\frac{\max\{a(p-1),1\}+1}{p+1}}\right\} = {\epsilon_g}^{1-a}$ for $0 < \epsilon_g < 1$.

□

Remark 6.4.1. ***(The choice of bound for a)*** *For odd $p \geq 3$, $1-a \geq \frac{a(p-2)+1}{p}$ is only true for $0 \leq a \leq \frac{1}{2}$. For even $p \geq 4$, $1-a \geq \frac{\max\{a(p-1),1\}+1}{p+1}$ is only true for $0 \leq a \leq \frac{1}{2}$. This rationale justifies the selection of $a \in [0, \frac{1}{2}]$. Further discussion regarding $a \geq a_b$ is provided in Section A.4.2.*

Theorem 6.4.2. ***(Function value decrease in successful iterations)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Under Assumptions 6.3.1–6.3.2, the mechanism of Algorithm 6.2.1 guarantees that, in any successful iteration $k \geq 0$ and for any $0 < \epsilon_g < 1$,*

$$\textit{Case 1 (odd } p\textit{):} \quad \lambda_{\min}[H_k] \geq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) > \kappa_1 \eta {\epsilon_g}^{\frac{p+1+a(p-2)}{p}}, \tag{6.32}$$

$$\textit{Case 1 (even } p\textit{):} \quad \lambda_{\min}[H_k] \geq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) > \kappa_2 \eta {\epsilon_g}^{\frac{p+2+\max\{1,a(p-1)\}}{p+1}}, \tag{6.33}$$

$$\textit{Case 2:} \quad \lambda_{\min}[H_k] < 0, \quad f(x_k) - f(x_k + s_k) > \frac{\eta \hat{\kappa}_3^2}{2(\Lambda_2 + 1)} {\epsilon_g}^2, \tag{6.34}$$

$$\textit{Case 3:} \quad 0 \leq \lambda_{\min}[H_k] \leq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) > \frac{\eta \hat{\kappa}_3^2}{2} {\epsilon_g}^{2-a}, \tag{6.35}$$

where $\kappa_1, \eta, \hat{\kappa}_4^2, \hat{\kappa}_4^2$ and Λ_2 are iteration-independent and ϵ_g-independent constants.

Proof. Denote $E_k := f(x_k) - f(x_k + s_k)$.
Case 1 (locally strongly convex): Using (6.1), we have

$$\eta^{-1} E_k \geq f(x_k) - T_p(x_k, s_k) \underset{(6.8)}{\geq} \frac{\sigma_k}{p'} \|s_k\|^{p'}.$$

According to Theorem 6.4.1, we have $\|s_k\| > \frac{1}{2}\min\left\{\left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}, \left(\frac{\epsilon_g}{\sigma_k}\right)^{\frac{1}{p'-1}}\right\}$. If $\sigma_k \leq {\epsilon_g}^{1-\frac{p'-1}{p}} L^{\frac{p'-1}{p}}$ (i.e., for odd p, this is $\sigma_k \leq L$), then we have $\|s_k\| > \frac{1}{2}\left(\frac{\epsilon_g}{L}\right)^{\frac{1}{p}}$. Consequently,

$$\eta^{-1} E_k > \frac{\sigma_k}{2^{p'} p'} \left(\frac{\epsilon_g}{L}\right)^{\frac{p'}{p}} \geq \frac{\sigma_{\min}}{2^{p'} p'} \left(\frac{\epsilon_g}{L}\right)^{\frac{p'}{p}} \tag{6.36}$$

where the last inequality uses the lower bound $\sigma_k \geq \sigma_{\min}$ as defined in Algorithm 6.2.1. Otherwise, if $\sigma_k \geq {\epsilon_g}^{1-\frac{p'-1}{p}} L^{\frac{p'-1}{p}}$ (i.e., for odd p, this is $\sigma_k \geq L$), we have $\|s_k\| > \frac{1}{2}\big(\frac{\epsilon_g}{\sigma_k}\big)^{\frac{1}{p'-1}}$. Consequently,

$$\eta^{-1}E_k > \frac{{\sigma_k}^{-\frac{1}{p'-1}}}{2^{p'}p'}{\epsilon_g}^{\frac{p'}{p'-1}} \geq \frac{1}{2^{p'}p'}{\sigma_{\max}}^{-\frac{1}{p'-1}}{\epsilon_g}^{\frac{p'}{p'-1}} \tag{6.37}$$

- If p is odd, $p' = p+1$, $\sigma_{\max} = \hat{C}_R{\epsilon_g}^{a(2-p)}$, then

$$\eta^{-1}E_k > \frac{1}{2^{p'}p'}\hat{C}_R^{-\frac{1}{p}}{\epsilon_g}^{\frac{p+1+a(p-2)}{p}}. \tag{6.38}$$

  Combining (6.36) and (6.38), we arrive that $\eta^{-1}E_k \geq \frac{1}{2^{p'}p'}\min\big\{\sigma_{\min}L^{-\frac{p+1}{p}}{\epsilon_g}^{\frac{p+1}{p}}, \hat{C}_R^{-\frac{1}{p}}{\epsilon_g}^{\frac{p+1+a(p-2)}{p}}\big\}$. Setting $\kappa_1 = \frac{1}{2^{p'}p'}\min\big\{\sigma_{\min}L^{-\frac{p+1}{p}}, \hat{C}_R^{-\frac{1}{p}}\big\}$ gives in (6.32).

- If p is even, $p' = p+2$, $\sigma_{\max} := \tilde{C}_R\max\big\{{\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)}\big\}$, then

$$\eta^{-1}E_k > \frac{1}{2^{p'}p'}\tilde{C}_R^{-\frac{1}{p+1}}{\epsilon_g}^{\frac{p+2+\max\{1,a(p-1)\}}{p+1}}. \tag{6.39}$$

  combining (6.36) and (6.39), we arrive that

$$\eta^{-1}E_k \geq \frac{1}{2^{p'}p'}\min\big\{\sigma_{\min}L^{-\frac{p+2}{p}}{\epsilon_g}^{\frac{p+2}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}{\epsilon_g}^{\frac{p+2+\max\{1,a(p-1)\}}{p+1}}\big\}.$$

  Setting $\kappa_2 = \frac{1}{2^{p'}p'}\min\big\{\sigma_{\min}L^{-\frac{p+2}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}\big\}$ gives in (6.32)[8].

Case 2 (locally nonconvex): Using (6.2), we have

$$\eta^{-1}E_k \geq f(x_k) - T_p(x_k, s_k) \underset{(6.9)}{\geq} \frac{1}{2}\big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\|^2 \tag{6.40}$$

$$\underset{(6.25)}{>} \frac{\hat{\kappa}_3^2}{2}\big(-\lambda_{\min}[H_k] + \delta\big)^{-1}{\epsilon_g}^2 \geq \frac{\hat{\kappa}_3^2}{2}(\Lambda_2 + 1)^{-1}{\epsilon_g}^2 \tag{6.41}$$

where the last inequality is obtained by Lemma 6.3.1 and $0 < \delta = {\epsilon_g}^a < 1$.

Case 3 (locally nearly strongly convex): Using (6.3), we have

$$\eta^{-1}E_k \geq f(x_k) - T_p(x_k, s_k) \underset{(6.10)}{\geq} \frac{\delta}{2}\|s_k\|^2 + \frac{\sigma_k}{p'}\|s_k\|^{p'} \underset{(6.26)}{>} \frac{\hat{\kappa}_4^2}{2}{\epsilon_g}^a{\epsilon_g}^{2-2a} = \frac{\hat{\kappa}_4^2}{2}{\epsilon_g}^{2-a}. \tag{6.42}$$

□

Based on Theorem 6.4.2, we analyze how the function value reduction changes with p and δ. In Table 6.1, we give illustrations of the lower bound in (6.32)–(6.35) for different convexity cases. We summarize the highlights as follows.

- **For locally strongly convex iterations,** as a decreases from $\frac{1}{2}$ to 0 (i.e., the lower bound of $\lambda_{\min}[H_k]$ increases from ${\epsilon_g}^{\frac{1}{2}}$ to $\mathcal{O}(1)$), if p is odd, the function value reduction increases from $\mathcal{O}({\epsilon_g}^{\frac{3}{2}})$ to $\mathcal{O}({\epsilon_g}^{\frac{p+1}{p}})$. If p is even, the function value reduction increases from $\mathcal{O}({\epsilon_g}^{\frac{3}{2}})$ to $\mathcal{O}({\epsilon_g}^{\frac{p+3}{p+1}})$.

[8] As indicated in (6.37), the dominating term for the function value reduction is $f(x_k) - f(x_k + s_k) > \frac{\eta}{2^{p'}p'}{\sigma_{\max}}^{-\frac{1}{p'-1}}{\epsilon_g}^{\frac{p'}{p'-1}}$. In the general AR$p$ framework [51; 53], we have $p' = p+1$ and $\sigma_{\max}$ is a constant independent of ϵ_g. Therefore, $f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{p+1}{p}})$ matching there result in [53, Thm 3.7]. In the universal ARPr framework [52], by setting $p' = p+2$, we have $\sigma_{\max} = \mathcal{O}({\epsilon_g}^{-\frac{1}{p}})$ [52, Lemma 3.5]. Therefore, $f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{p+2}{p+1}}{\epsilon_g}^{\frac{1}{p(p+1)}}) = \mathcal{O}({\epsilon_g}^{\frac{p+1}{p}})$ matching there result in [52, Thm 3.7].

However, the region for Case 1 shrinks from $\lambda_{\min}[H_k] \geq {\epsilon_g}^{\frac{1}{2}}$ to $\lambda_{\min}[H_k] \geq c > 0$ where $c > 0$ is a constant with magnitude $\mathcal{O}(1)$. For odd p, when $a = 0$, the bound for the function value reduction in a locally strongly convex iteration is $\mathcal{O}({\epsilon_g}^{\frac{p+1}{p}})$, matching the optimal complexity bound for a pth order method [41; 23; 44; 51; 53]. For even p, when $a = 0$, the bound for the function value reduction in a locally strongly convex iteration is $\mathcal{O}({\epsilon_g}^{\frac{p+3}{p+1}})$. This arises from the choice to regularize our model with a $p' = p + 2$ order of regularization instead of the $p' = p + 1$ order proposed by [41; 23; 44; 51; 53]. The decision to use a $p' = p + 2$ order of regularization when p is even is motivated by the properties of the SoS-convex Taylor model. As indicated by [1], even power in the regularization term is necessary to ensure that the model m_p can be SoS-convex.

- **For locally nearly strongly convex iterations,** conversely, as a decreases from $\frac{1}{2}$ to 0 (i.e., the lower bound of $\lambda_{\min}[H_k]$ increases from ${\epsilon_g}^{\frac{1}{2}}$ to $\mathcal{O}(1)$), the function value reduction for Case 3 worsens from $\mathcal{O}({\epsilon_g}^{\frac{3}{2}})$ to $\mathcal{O}({\epsilon_g}^2)$. The region for Case 3 expands from $0 \leq \lambda_{\min}[H_k] \leq {\epsilon_g}^{\frac{1}{2}}$ to $0 \leq \lambda_{\min}[H_k] \leq c$ where $c > 0$ is a constant with magnitude $\mathcal{O}(1)$.
- **For locally nonconvex iterations,** regardless of how we choose $a \in [0, \frac{1}{2}]$, the function value reduction remains the same at $\mathcal{O}({\epsilon_g}^2)$. This is expected, as we perturbed the second-order term by $-\lambda_{\min}[H_k]I_n + \delta I_n$. In locally nonconvex cases, the model m_c starts to differ from f (or T_p) from the second-order term of the Taylor expansion. Therefore, due to the quadratic regularisation term added in the model in nonconvex iterations, this bound matches the accuracy order for high-order trust-region methods [44].
- In a locally strongly convex iteration with $a = 0$, the magnitude of function value reduction increases for odd p and even p, respectively. A higher-order method provides a larger function reduction on these iterations. However, note that for locally nonconvex iterations, the magnitude of the function value reduction remains unaffected by p due to the term $\|s\|^2$.

Table 6.1: An illustration of the bound in (6.32)–(6.35) for different $a \in [0, \frac{1}{2}]$ in three convexity cases.

Odd $p \geq 3$	$\lambda_{\min}[H_k] \geq {\epsilon_g}^a$	$0 < \lambda_{\min}[H_k] \leq {\epsilon_g}^a$	$\lambda_{\min}[H_k] < 0$
$a = 0$	${\epsilon_g}^{\frac{p+1}{p}}$	${\epsilon_g}^2$	${\epsilon_g}^2$
$a = \frac{1}{3}$	${\epsilon_g}^{\frac{4p+1}{3p}}$	${\epsilon_g}^{\frac{5}{3}}$	${\epsilon_g}^2$
$a = \frac{1}{2}$	${\epsilon_g}^{\frac{3}{2}}$	${\epsilon_g}^{\frac{3}{2}}$	${\epsilon_g}^2$

Even $p \geq 4$	$\lambda_{\min}[H_k] \geq {\epsilon_g}^a$	$0 < \lambda_{\min}[H_k] \leq {\epsilon_g}^a$	$\lambda_{\min}[H_k] < 0$
$a = 0$	${\epsilon_g}^{\frac{p+3}{p+1}}$	${\epsilon_g}^2$	${\epsilon_g}^2$
$a = \frac{1}{3}$	${\epsilon_g}^{\frac{4(p+1)+1}{3(p+1)}}$	${\epsilon_g}^{\frac{5}{3}}$	${\epsilon_g}^2$
$a = \frac{1}{2}$	${\epsilon_g}^{\frac{3}{2}}$	${\epsilon_g}^{\frac{3}{2}}$	${\epsilon_g}^2$

In the light of Theorem 6.4.2, we propose practical choices for the parameter a in Algorithm 6.2.1.

Remark 6.4.2. *Let $0 < \epsilon_g < 1$. If $a = 0$ and $\delta = \mathcal{O}(1) = c$ where $c > 0$ is constant, then*

$$\begin{aligned}
&\textit{locally strongly convex Iterations (odd } p\textit{):} && \lambda_{\min}[H_k] \geq c, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{p+1}{p}}),\\
&\textit{locally strongly convex Iterationss (even } p\textit{):} && \lambda_{\min}[H_k] \geq c, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{p+3}{p+1}}),\\
&\textit{Locally Nearly Convex Iterations:} && 0 \leq \lambda_{\min}[H_k] \leq c, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^2),\\
&\textit{Locally Nonconvex Iterations:} && \lambda_{\min}[H_k] \leq 0, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^2).
\end{aligned}$$

Let $a = \frac{1}{2}$ and $\delta = {\epsilon_g}^{\frac{1}{2}}$, then the order of complexity bound in (6.32), (6.33) and (6.35) equates,

$$\begin{aligned}
&\textit{locally strongly convex Iterations:} && \lambda_{\min}[H_k] \geq {\epsilon_g}^{\frac{1}{2}}, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{3}{2}}),\\
&\textit{Locally Nearly Convex Iterations:} && 0 \leq \lambda_{\min}[H_k] \leq {\epsilon_g}^{\frac{1}{2}}, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{3}{2}}),\\
&\textit{Locally Nonconvex Iterations:} && \lambda_{\min}[H_k] \leq 0, && f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{2}).
\end{aligned}$$

6.4.1.1 Improved Function Value Decrease for $3 \leq p \leq 5$ in Locally Convex Iterations

In this subsection, we demonstrate that for locally convex iterations with $3 \leq p \leq 5$, we can achieve an enhanced function value reduction per iteration. Since m_c is SoS-convex, according to Definition 6.2.1 and Definition 5.2.2, we can deduce that m_c is convex [1; 156]. The idea of the proof is to use the convexity of m_c (i.e., $\nabla^2 m_c(x_k, s) \succeq 0$ for all $s \in \mathbb{R}^n$) to derive a better bound for function value reduction in the locally convex iterations (i.e., Case 1). Note that the proof of the improved function value reduction bound in strongly convex iterations for $3 \leq p \leq 5$ is inspired by [50], although the analyses in [50] are limited to the case $p = 2$.

Theorem 6.4.3. ***(Function value decrease in successful iteration for** $3 \leq p \leq 5$**)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $3 \leq p \leq 5$. Let $0 < \epsilon_g < 1$ and $\delta = {\epsilon_g}^a$ with $a \in [0, \frac{1}{2}]$. Under Assumptions 6.3.1–6.3.2, the mechanism of Algorithm 6.2.1 guarantees that, in successful iteration, for all $k \geq 0$,*

$$\textit{Case 1 } (p = 3, 5): \quad \lambda_{\min}[H_k] \geq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) \geq \eta \hat{c} {\epsilon_g}^{a + \frac{2}{p}(1 + a(p-2))}, \tag{6.43}$$

$$\textit{Case 1 } (p = 4): \quad \lambda_{\min}[H_k] \geq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) \geq \eta \hat{c}_1 {\epsilon_g}^{a + \frac{2}{p+1}(1 + \max\{1, a(p-1)\})}, \tag{6.44}$$

$$\textit{Case 2:} \quad \lambda_{\min}[H_k] < 0, \quad f(x_k) - f(x_k + s_k) \geq \frac{\eta \hat{\kappa}_3^2}{2(\Lambda_2 + 1)} {\epsilon_g}^2,$$

$$\textit{Case 3:} \quad 0 \leq \lambda_{\min}[H_k] \leq {\epsilon_g}^a, \quad f(x_k) - f(x_k + s_k) \geq \frac{\eta \hat{\kappa}_4^2}{2} {\epsilon_g}^{2-a},$$

as $\epsilon_g \to 0$, where $\hat{c}, \hat{c}_1, \eta$, and Λ_2 are iteration-independent and ϵ_g-independent constants.

Proof. **(Proof for $p = 3$)** The function value reductions for Cases 2 and 3 also follow similarly as in Theorem 6.4.2. In this proof, we establish the function value reductions for Case 1 when $p = 3$. While the proof for $p = 4, 5$ follows a similar methodology as the proof for $p = 3$, it is more intricate. Hence, we defer the proof for $p = 4, 5$ to Section 6.4.1.2.

Firstly, we denote $f_k = f(x_k) \in \mathbb{R}$, $g_k = \nabla_x f(x_k) \in \mathbb{R}^n$, $H_k = \nabla_x^2 f(x_k) \in \mathbb{R}^{n \times n}$ and $\hat{\mathcal{T}}_k = \nabla_x^3 f(x_k) \in \mathbb{R}^{n^3}$. Then, m_c and its derivatives have the following expression.

$$\begin{aligned}
m_c(x_k, s) : &= f_k + g_k^T s + \frac{1}{2} H_k[s]^2 + \frac{1}{6} \hat{\mathcal{T}}_k[s]^4 + \frac{\sigma_k}{4} \|s\|^4,\\
\nabla m_c(x_k, s) : &= g_k + H_k[s] + \frac{1}{2} \hat{\mathcal{T}}_k[s]^2 + \sigma_k \|s\|^2 s,\\
\nabla^2 m_c(x_k, s) : &= H_k + \hat{\mathcal{T}}_k[s] + \sigma_k(\|s\|^2 I_n + 2ss^T).
\end{aligned}$$

The function value reduction in each successful iteration is

$$\eta^{-1} E_k = \eta^{-1} \Big[f(x_k) - f(x_k + s_k) \Big] \geq f(x_k) - T_p(x_k, s_k) = -g_k^T s_k - \frac{1}{2} H_k[s_k]^2 - \frac{1}{6} \hat{\mathcal{T}}_k[s_k]^3. \tag{6.45}$$

Sine $s_k = \operatorname{argmin}_{s \in \mathbb{R}^n} m_c(x_k, s)$, by first-order optimality condition $\nabla m_c(x_k, s) = 0$, we have $g_k = -H_k[s_k] - \frac{1}{2} \hat{\mathcal{T}}_k[s_k]^2 - \sigma_k \|s_k\|^2 s_k$. Note that an approximate first-order minimum of $m_c(x_k, s)$ is also allowed as discussed in Remark 6.2.6. Substituting the 1st order optimality condition into (6.45), we have

$$\eta^{-1} E_k \geq \frac{1}{2} H_k[s_k]^2 + \frac{1}{3} \hat{\mathcal{T}}_k[s_k]^3 + \sigma_k \|s_k\|^4. \tag{6.46}$$

Also, since $m_c(x_k, s)$ is SoS-convex with respect to s, we deduce that $m_c(x_k, s)$ is convex, that is

$$\left[H_k + \hat{\mathcal{T}}_k[s] + \sigma_k(\|s\|^2 I_n + 2ss^T)\right][u^2] \succeq 0, \qquad \forall s, u \in \mathbb{R}^n. \tag{6.47}$$

By setting $s = \frac{3}{4}s_k$ and $u = \frac{2}{3}s_k$ in (6.47), we have $\frac{4}{9}H_k[s_k]^2 + \frac{1}{3}\hat{\mathcal{T}}_k[s_k]^3 + \frac{3\sigma_k}{4}\|s_k\|^4 \geq 0$. Substituting this expression into (6.46), and using $H_k \succeq \delta I_n$ for Case 1, we have

$$\eta^{-1}E_k \geq \frac{1}{18}H_k[s_k]^2 + \underbrace{\frac{\sigma_k}{4}\|s_k\|^4}_{\geq 0} \geq \frac{\delta}{18}\|s_k\|^2 \underset{(6.24)}{\geq} \frac{{\epsilon_g}^a}{72}\min\left\{\left(\frac{\epsilon_g}{L}\right)^{\frac{2}{p}}, \left(\frac{\epsilon_g}{\sigma_{\max}}\right)^{\frac{2}{p}}\right\}$$

where $\sigma_{\max} = \hat{C}_R {\epsilon_g}^{a(2-p)}$. Set $\hat{c} = \frac{1}{72}\min\{L^{-\frac{2}{p}}, \hat{C}_R^{-\frac{2}{p}}\}$, then $\eta^{-1}E_k \geq \hat{c}\min\{{\epsilon_g}^{a+\frac{2}{p}}, {\epsilon_g}^{a+\frac{2}{p}(1+a(p-2))}\} = \hat{c}{\epsilon_g}^{a+\frac{2}{p}(1+a(p-2))}$ which gives the desired bound. □

We compare the improved rate of decrease in function value in Theorem 6.4.3 to the rate given in Theorem 6.4.2. In the locally convex case, the improved rate gives a large function value reduction for all choices of $a \in [0, \frac{1}{2}]$. In the light of Theorem 6.4.3, we propose practical choices for the parameter a for $3 \leq p \leq 5$ in Algorithm 6.2.1.

Remark 6.4.3. *Let $0 < \epsilon_g < 1$. Let $a = 0$ and $\delta = \mathcal{O}(1) = c$ where $c > 0$ is constant, then*

locally strongly convex Iterations ($p = 3, 5$):	$\lambda_{\min}[H_k] \geq c$,	$f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{2}{p}})$,
locally strongly convex Iterations ($p = 4$):	$\lambda_{\min}[H_k] \geq c$,	$f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^{\frac{4}{5}})$,
Locally Nearly Convex Iterations:	$0 \leq \lambda_{\min}[H_k] \leq c$,	$f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^2)$,
Locally Nonconvex Iterations:	$\lambda_{\min}[H_k] \leq 0$,	$f(x_k) - f(x_k + s_k) > \mathcal{O}({\epsilon_g}^2)$.

6.4.1.2 Proof for Improved Function Value Decrease for $p = 4, 5$ in Locally Convex Iterations

Proof. **(Proof for $p = 4, 5$)** We start by giving an expression for m_c and its derivatives for $p = 4, 5$ in locally convex iterations (i.e., $\lambda_{\min}[H_k] \geq \delta$). Let $\hat{\mathcal{T}}_k = \nabla_x^3 f(x_k) \in \mathbb{R}^{n^3}$, $\mathcal{M}_k := \nabla_x^4 f(x_k) \in \mathbb{R}^{n^4}$ and $\mathcal{N}_k := \nabla_x^5 f(x_k) \in \mathbb{R}^{n^5}$. Then, m_c and its derivatives have the following expression. Note that in the case of $p = 4$, we have $\mathcal{N}_k = 0$.

$$m_c(x_k, s) : = f_k + g_k^T s + \frac{1}{2}H_k[s]^2 + \frac{1}{3!}\hat{\mathcal{T}}_k[s]^3 + \frac{1}{4!}\mathcal{M}_k[s]^4 + \frac{1}{5!}\mathcal{N}_k[s]^5 + \frac{\sigma_k}{6}\|s\|^6, \tag{6.48}$$

$$\nabla m_c(x_k, s) : = g_k + H_k[s] + \frac{1}{2}\hat{\mathcal{T}}_k[s]^2 + \frac{1}{6}\mathcal{M}_k[s]^3 + \frac{1}{24}\mathcal{N}_k[s]^4 + \sigma_k\|s\|^4 s, \tag{6.49}$$

$$\nabla^2 m_c(x_k, s) : = H_k + \hat{\mathcal{T}}_k[s] + \frac{1}{2}\mathcal{M}_k[s]^2 + \frac{1}{6}\mathcal{N}_k[s]^3 + \sigma_k\|s\|^2\Big(\|s\|^2 I_n + 4ss^T\Big). \tag{6.50}$$

Let $E_k := f(x_k) - f(x_k + s_k)$. In each successful iteration, E_k satisfies

$$\begin{aligned}\eta^{-1}E_k &\geq f(x_k) - T_p(x_k, s_k) = -g_k^T s_k - \frac{1}{2}H_k[s_k]^2 - \frac{1}{3!}\hat{\mathcal{T}}_k[s_k]^3 - \frac{1}{4!}\mathcal{M}_k[s_k]^4 - \frac{1}{5!}\mathcal{N}_k[s_k]^5, \\ &= \frac{1}{2}H_k[s_k]^2 + \frac{1}{3}\hat{\mathcal{T}}_k[s_k]^3 + \frac{1}{8}\mathcal{M}_k[s_k]^4 + \frac{1}{30}\mathcal{N}_k[s_k]^5 + \sigma_k\|s_k\|^6 \end{aligned} \tag{6.51}$$

where the second line uses (6.49) and $\nabla m_c(x_k, s_k)^T s_k = 0$.

(**Scenario I**: assume $\mathcal{N}_k[s_k]^5 \geq 0$) Since $m_c(x_k, s)$ is SoS-convex, we deduce that $m_c(x_k, s)$ is convex. Using $\nabla^2 m_c(x_k, s)[u]^2 \geq 0$ with $s = \frac{3}{4}s_k$ and $u = \frac{2}{3}s_k$, we have $\frac{4}{9}H_k[s_k]^2 + \frac{1}{3}\hat{\mathcal{T}}_k[s_k]^3 + \frac{1}{8}\mathcal{M}_k[s_k]^4 + \frac{1}{32}\mathcal{N}_k[s_k]^5 + \frac{45}{64}\sigma_k\|s_k\|^6 \geq 0$. Substituting this expression into (6.51), we have

$$\eta^{-1}E_k \geq \left(\frac{1}{2} - \frac{4}{9}\right)\underbrace{H_k}_{\succeq \delta I_n}[s_k]^2 + \left(\frac{1}{30} - \frac{1}{32}\right)\underbrace{\mathcal{N}_k[s_k]^5}_{\geq 0} + \left(1 - \frac{45}{64}\right)\sigma_k\underbrace{\|s_k\|^6}_{\geq 0} \geq \frac{\delta}{18}\|s_k\|^2. \tag{6.52}$$

Note that the second inequality follows using the assumption of Scenario I ($N_k[s_k]^5 \geq 0$) and $H_k \succeq \delta I_n$.

(**Scenario II**: assume $\mathcal{N}_k[s_k]^5 \leq 0$) Using the convexity of m_c, we have $0 \leq \nabla^2 m_c(x_k, s_k)[s_k]^2 = H_k[s_k]^2 + \hat{\mathcal{T}}_k[s_k]^3 + \frac{1}{2}\mathcal{M}_k[s_k]^4 + \frac{1}{6}\mathcal{N}_k[s_k]^5 + 5\sigma_k\|s_k\|^6$. For any $\hat{\tau} > 0$, this expression can be rearranged to give

$$\frac{\hat{\tau}}{5}H_k[s_k]^2 \geq \frac{\hat{\tau}}{5}\Big(- \hat{\mathcal{T}}_k[s_k]^3 - \frac{1}{2}\mathcal{M}_k[s_k]^4 - \frac{1}{6}\mathcal{N}_k[s_k]^5 - 5\sigma_k\|s_k\|^6\Big). \tag{6.53}$$

We can rewrite the second-order term in (6.51) as $\big(\frac{1}{2} - \frac{\hat{\tau}}{5}\big)H_k[s_k]^2 + \frac{\hat{\tau}}{5}H_k[s_k]^2$ and substitute (6.53) which gives

$$E_k \geq \big(\frac{1}{2} - \frac{\hat{\tau}}{5}\big)H_k[s_k]^2 + \big(\frac{1}{3} - \frac{\hat{\tau}}{5}\big)\hat{\mathcal{T}}_k[s_k]^3 + \big(\frac{1}{8} - \frac{\hat{\tau}}{10}\big)\mathcal{M}_k[s_k]^4 + \big(\frac{1}{30} - \frac{\hat{\tau}}{30}\big)\mathcal{N}_k[s_k]^5 + \sigma_k\big(1 - \hat{\tau}\big)\|s_k\|^6 \tag{6.54}$$

for all $\hat{\tau} > 0$. Let $a, b \in \mathbb{R}$ and $s = as_k$ and $u = bs_k$. Using convexity of m_c, we have

$$\nabla^2 m_c(x_k, s)[u]^2 = a^2 H_k[s_k]^2 + a^2 b\hat{\mathcal{T}}_k[s_k]^3 + \frac{a^2b^2}{2}\mathcal{M}_k[s_k]^4 + \frac{a^2b^3}{6}\mathcal{N}_k[s_k]^5 + 5a^2b^4\sigma_k\|s_k\|^6 \geq 0. \tag{6.55}$$

We can rewrite the second-order term in (6.54) as $\big(\frac{1}{2} - \frac{\hat{\tau}}{5} - a^2\big)H_k[s_k]^2 + a^2 H_k[s_k]^2$ and substitute (6.55) which gives

$$\begin{aligned}\eta^{-1}E_k \geq \big(\frac{1}{2} - \frac{\hat{\tau}}{5} - a^2\big)H_k[s_k]^2 + \big(\frac{1}{3} - \frac{\hat{\tau}}{5} - a^2 b\big)\hat{\mathcal{T}}_k[s_k]^3 + \big(\frac{1}{8} - \frac{\hat{\tau}}{10} - \frac{a^2b^2}{2}\big)\mathcal{M}_k[s_k]^4 + \\ + \big(\frac{1}{30} - \frac{\hat{\tau}}{30} - \frac{a^2b^3}{6}\big)\mathcal{N}_k[s_k]^5 + \sigma_k(1 - \hat{\tau} - 5a^2b^4)\|s_k\|^6\end{aligned} \tag{6.56}$$

for all $\hat{\tau} > 0$, $a, b \in \mathbb{R}$. Set $\hat{\tau} = \frac{4}{5}$, $a = \frac{26}{45}$ and $b = \frac{27}{52}$, we have $\frac{1}{3} - \frac{\hat{\tau}}{5} - a^2 b = 0$ and $\frac{1}{8} - \frac{\hat{\tau}}{10} - \frac{a^2b^2}{2} = 0$ and (6.56) becomes

$$\eta^{-1}E_k \geq \frac{1}{162}\underbrace{H_k}_{\succeq \delta I_n}[s_k]^2 - \frac{7}{6240}\underbrace{\mathcal{N}_k[s_k]^5}_{\leq 0} + \frac{851}{10816}\sigma_k\underbrace{\|s_k\|^6}_{\geq 0} \geq \frac{\delta}{162}\|s_k\|^2 \tag{6.57}$$

$$\underset{(6.24)}{\geq} \frac{{\epsilon_g}^a}{648}\min\left\{\left(\frac{\epsilon_g}{L}\right)^{\frac{2}{p}}, \left(\frac{\epsilon_g}{\sigma_{\max}}\right)^{\frac{2}{p'-1}}\right\}. \tag{6.58}$$

For $p = 5$, the rest of the analysis follows similarly as in the case of $p = 3$. For $p = 4$, we have $\sigma_{\max} = \tilde{C}_R \max\{{\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)}\}$. We set $\hat{c}_1 = \frac{1}{648}\min\{L^{-\frac{2}{p}}, \tilde{C}_R^{-\frac{2}{p+1}}\}$ and

$$\eta^{-1}E_k \geq \hat{c}_1 \min\left\{{\epsilon_g}^{a+\frac{2}{p}}, {\epsilon_g}^{a+\frac{2}{p+1}(1+\max\{1,a(p-1)\})}\right\} = \hat{c}_1{\epsilon_g}^{a+\frac{2}{p+1}(1+\max\{1,a(p-1)\})}.$$

□

Remark 6.4.4. ***(Generalization to Higher Order)*** *Note that the structure of the proof relies heavily on the specific polynomial form and degree, and thus may be difficult to generalize to higher degrees. In the proof, we carefully select the search directions and step sizes so that, after balancing the polynomial terms, we achieve an overall improvement in the function decrease per iteration. Adjusting a, b, and τ to ensure $E_k \geq \hat{c}\delta\|s_k\|^2$ becomes increasingly intricate as the order p increases.*

6.4.2 Overall Complexity Bound

We have discussed the complexity bounds for locally convex and nonconvex iterations. In this section, we provide the overall complexity bound for Algorithm 6.2.1. We define the sets of *successful and unsuccessful iterations* for Algorithm 6.2.1 as $\mathcal{S}_m$ and $\mathcal{U}_m := [0 : m]\backslash\mathcal{S}_m$. Notice that for $k \in \mathcal{S}_m$, $\rho_k \geq \eta$ and $x_{k+1} = x_k + s_k$, while for $k \in \mathcal{U}_m$, $\rho_k \leq \eta$ and $x_{k+1} = x_k$. Lemma 6.4.1 describes the total number of iterations required to compute an ϵ_g-approximate first-order minimizer using Algorithm 6.2.1, as a function of successful iterations.

Lemma 6.4.1. ***(Successful and unsuccessful adaptive-regularization iterations)*** *Suppose that Algorithm 6.2.1 is used and that $\sigma_k \leq \sigma_{\max}$ for some $\sigma_{\max} > 0$. Then,*

$$m \leq |\mathcal{S}_m|\left(1 + \frac{|\log \gamma_2|}{\log \gamma_1}\right) + \frac{1}{\log \gamma_1} \log\left(\frac{\sigma_{\max}}{\sigma_0}\right). \tag{6.59}$$

Proof. The update of regularization parameters in Algorithm 6.2.1 gives that, for each k,

$$0 \leq \gamma_2 \sigma_k \leq \max\{\bar{\sigma}_{k+1}, \sigma^r_{k+1}\} = \sigma_{k+1}, \quad k \in \mathcal{S}_m,$$

and

$$0 \leq \gamma_1 \sigma_j \leq \max\{\bar{\sigma}_{k+1}, \sigma^r_{k+1}\} = \sigma_{k+1}, \quad k \in \mathcal{U}_m$$

where $k \in [0:m]$. We inductively deduce that $0 \leq \gamma_2^{|\mathcal{S}_m|}\gamma_1^{|\mathcal{U}_m|}\sigma_0 \leq \sigma_k$. Therefore, using our assumption that $\sigma_k \leq \sigma_{\max}$, we deduce that

$$|\mathcal{S}_m| \log \gamma_2 + |\mathcal{U}_m| \log \gamma_1 \leq \log\left(\frac{\sigma_{\max}}{\sigma_0}\right).$$

The desired result follows from the inequality $m = |\mathcal{U}_m| + |\mathcal{S}_m|$. □

Remark 6.4.5 (Choice of γ_1 and γ_2)**.** *The constants $\gamma_1 > 1$ and $\gamma_2 \in (0,1)$ control the update of the regularization parameter after unsuccessful and successful iterations, respectively. From the iteration bound in (6.59), a larger value of γ_1 and a value of γ_2 closer to one reduce the worst-case upper bound on the number of unsuccessful iterations. However, choosing γ_1 too large may lead to overly aggressive increases of the regularization parameter, resulting in unnecessarily conservative trial steps. In practice, we therefore use moderate values to provide a simple and stable balance between theoretical complexity control and practical performance.*

Lemma 6.4.1 employs a similar technique to that in [44, Lemma 2.3.1, Lemma 2.4.1]. Note that Lemma 6.4.1 is independent of the form of the model m_c and depends on the mechanism defined by the update of regularization parameters in Algorithm 6.2.1.

Lemma 6.4.2. ***(Bound on the number of successful iterations)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Suppose that Assumptions 6.3.1–6.3.2 hold. Let f_{low} be a lower bound on $f(x)$ for $x \in \mathbb{R}^n$. Then, there exists a positive constant κ_s such that Algorithm 6.2.1 requires at most*

$$|\mathcal{S}_m| := \kappa_s \frac{f(x_0) - f_{low}}{{\epsilon_g}^2} \tag{6.60}$$

successful iterations before an iterate s_{ϵ_g} is computed for which $\|g_k\| \leq \epsilon_g$ for any $0 < \epsilon_g < 1$.

Proof. According Theorem 6.4.2, for all $k \in [0:(m-1)]$,

$$f(x_k) - f(x_{k+1}) \geq \min\left\{\kappa_1 \eta {\epsilon_g}^{\frac{p+1+a(p-2)}{p}}, \quad \kappa_2 \eta {\epsilon_g}^{\frac{p+2+\max\{1,a(p-1)\}}{p+1}}, \quad \frac{\eta \hat{\kappa_3}^2}{2(\Lambda_2+1)}{\epsilon_g}^2, \quad \frac{\eta \hat{\kappa_4}^2}{2}{\epsilon_g}^{2-a}\right\} \tag{6.61}$$

Given that $p \geq 3$, $a \in [0, \frac{1}{2}]$, and $0 < \epsilon_g < 1$, by setting

$$\hat{\kappa}_s := \min\left\{\kappa_1 \eta, \kappa_2 \eta, \frac{\eta \hat{\kappa_3}^2}{2(\Lambda_2+1)}, \frac{\eta \hat{\kappa_4}^2}{2}\right\}, \qquad \kappa_s := \hat{\kappa}_s^{-1}, \tag{6.62}$$

we have $f(x_k) - f(x_{k+1}) \geq \hat{\kappa}_s {\epsilon_g}^2$. Suppose that $\|g_k\| > \epsilon_g$ for all $k \in [0:(m-1)]$. Then,

$$f(x_0) - f_{\text{low}} \geq f(x_0) - f(x_k) = \sum_{k \in \mathcal{S}_m}\Big[f(x_k) - f(x_{k+1})\Big] > |\mathcal{S}_m| \hat{\kappa}_s {\epsilon_g}^2,$$

we arrive at the bound (6.60). □

Given Lemma 6.4.2, we are now ready to state the worst-case evaluation bound for Algorithm 6.2.1.

Theorem 6.4.4. ***(Complexity bound for Algorithm 6.2.1)*** *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Suppose that Assumptions 6.3.1–6.3.2 hold. Then, there exists positive constants $\bar{\kappa}_s$, κ_c, κ_d such that Algorithm 6.2.1 requires at most*

$$\frac{f(x_0) - f_{low}}{\bar{\kappa}_s {\epsilon_g}^2} + \kappa_c + \kappa_d \log({\epsilon_g}^{-1}) = \mathcal{O}({\epsilon_g}^{-2})$$

function evaluations (i.e., $f(x_k)$) and at most

$$\frac{f(x_0) - f_{low}}{\bar{\kappa}_s {\epsilon_g}^2} + 1 = \mathcal{O}({\epsilon_g}^{-2})$$

derivatives evaluations (i.e., $\nabla_x^j f(x_k)$ for $j \geq 1$) to compute an iterate s_{ϵ_g} such that $\|g_k\| \leq \epsilon_g$ for any $0 < \epsilon_g < 1$.

Proof. The number of successful iterations needed to find an ϵ_g-approximate first-order minimizer is bounded above by (6.60). The total number of iterations (including unsuccessful ones) can also be bounded above (Lemma 6.4.1). Using $\sigma_{\max} = \hat{C}_R {\epsilon_g}^{a(2-p)}$ for odd p and $\sigma_{\max} = \tilde{C}_R \max\left\{{\epsilon_g}^{-1}, {\epsilon_g}^{a(1-p)}\right\}$ for even p and substituting (6.60) into (6.59) yields

$$m \leq \frac{f(x_0) - f_{\text{low}}}{\kappa_s {\epsilon_g}^2}\left(1 + \frac{|\log \gamma_2|}{\log \gamma_1}\right) + \kappa_c + \kappa_d \log({\epsilon_g}^{-1}).$$

Note that κ_s is defined in (6.62), $\kappa_c := \frac{1}{\log \gamma_1} \log\left(\frac{\hat{C}_R}{\sigma_0}\right)$ and $\kappa_d = \frac{a(p-2)}{\log \gamma_1}$ for odd p. $\kappa_c := \frac{1}{\log \gamma_1} \log\left(\frac{\tilde{C}_R}{\sigma_0}\right)$ and $\kappa_d = \frac{\max\{1, a(p-1)\}}{\log \gamma_1}$ for even p. Since Algorithm 6.2.1 uses at most one evaluation of m_c per iteration and at most one evaluation of its derivatives per successful iteration (plus one evaluation of m_c and its derivatives at the final iteration). By setting $\bar{\kappa}_s^{-1} = \kappa_s^{-1}\left(1 + \frac{|\log \gamma_2|}{\log \gamma_1}\right)$, we deduce the desired conclusion. □

Remark 6.4.6. *The overall complexity bound is primarily influenced by the function value reduction in locally nonconvex cases. In these iterations, quadratic regularization is introduced to the nonconvex m_c for SoS convexification. As a result, m_c starts to deviate from T_p or f from the second-order term. The addition of the quadratic regularization term in nonconvex iterations gives the accuracy order that matches the accuracy order of the high-order trust-region methods [44].*

6.4.2.1 Improved Complexity for Strongly Convex Functions

For strongly convex functions $f(x)$, the evaluation complexity bound for the ϵ_g-approximate minimum value of the convex functions is $\mathcal{O}({\epsilon_g}^{-\frac{1}{p}})$ for odd p, and the evaluation complexity bound of $\mathcal{O}({\epsilon_g}^{-\frac{p+1}{p}})$ for even p. The global convergence rate for odd p matches the bound by Nesterov in [137, Thm 2]. It is worth noting that in [8, Thm 5], Ahmadi and Zhang derived the global convergence for odd orders $p \geq 3$ for the higher-order Newton method on a strongly convex objective function [137, Thm 2]. In this section, we extend this result to include even orders $p \geq 2$. Also, note that neither [8, Thm 5] nor [137, Thm 2] discusses the adaptive regularization mechanism. We provide the global convergence rate of Algorithm 6.2.1 when minimizing a strongly convex objective function in Theorem 6.4.5. To lay the groundwork, we introduce two technical lemmas.

Lemma 6.4.3. *([52, Lemma 3.2]) Let $D_\Lambda := \max_{1 \leq j \leq p}\left\{\left(\frac{pp'}{j!\sigma_{\min}}\Lambda_j\right)^{\frac{1}{p'-j}}, 1\right\} > 1$ be a constant independent of k, ϵ_g and δ. Then, at each iteration of Algorithm 6.2.1, we have*

$$\|s_k\| \leq D_\Lambda. \tag{6.63}$$

Lemma 6.4.4. *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Suppose that Assumption 6.3.1 holds and that f is a strongly convex function such that $\nabla_x f(x) \succ \check{c} I_n \succ 0$ for some constant $\check{c} > 0$. The mechanism of Algorithm 6.2.1 guarantees that in a successful iteration,*

$$f(x_{k+1}) = f(x_k + s_k) \leq \min_{s \in \mathbb{R}^n} \left[f(x_k + s) + \frac{L + D_\Lambda \sigma_{\max}}{p'} \|s\|^{p+1} \right]. \tag{6.64}$$

where D_Λ is defined in Lemma 6.4.3. For odd p, $\sigma_{\max} = \hat{C}_R \max\{1, \check{c}^{2-p}\}$, which is an iteration-independent and ϵ_g-independent constant. For even p, $\sigma_{\max} = \tilde{C}_R {\epsilon_g}^{-1}$ for any $0 < \epsilon_g < 1$.

Proof. This proof is adapted from [137, Lemma 1] and uses similar techniques as those in [137, Lemma 1]. If $\nabla_x f(x) \succ \check{c} I_n \succ 0$, then in (6.12) and (6.21), we can replace δ with $\check{c}$, which becomes an iteration-independent and ϵ_g-independent constant. Consequently, for odd p, (6.14) becomes $\sigma_k \leq \hat{C}_R \max\{1, \check{c}^{2-p}\} = \sigma_{\max}$. For even p, (6.23) becomes $\sigma_k \leq \tilde{C}_R \max\{1, {\epsilon_g}^{-1}, \check{c}^{1-p}\} = \tilde{C}_R {\epsilon_g}^{-1} = \sigma_{\max}$ for $0 < \epsilon_g < 1$.

Assume s_k is a successful iteration, then by (Ratio Test), we have $\eta^{-1}[f(x_k) - f(x_k + s_k)] \geq f(x_k) - T_p(x_k, s_k)$ and $f(x_k + s_k) \leq f(x_k)$. Using these two inequalities, we deduce that

$$\begin{aligned} f(x_{k+1}) = f(x_k + s_k) &\leq (1 - \eta^{-1}) f(x_k) + \eta^{-1} f(x_k + s_k) \leq T_p(x_k, s_k) \\ &\leq T_p(x_k, s_k) + \frac{\sigma_k}{p'} \|s_k\|^{p'}. \end{aligned}$$

By adding and subtracting $\frac{L}{p+1}\|s\|^{p+1}$, adding $\frac{\sigma_k}{p'}\|s\|^{p'}$ and (6.63), we deduce

$$f(x_{k+1}) \leq \min_{s \in \mathbb{R}^n} \left[\underbrace{T_p(x_k, s) - \frac{L}{p+1}\|s\|^{p+1}}_{\text{By (1.5)}, \leq f(x_k + s)} + \frac{L}{p+1}\|s\|^{p+1} + \frac{D_\Lambda \sigma_{\max}}{p'}\|s\|^{p+1} \right].$$

(6.64) follows immediately from the fact that $\frac{1}{p'} < \frac{1}{p+1}$. □

Theorem 6.4.5. *Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ with $p \geq 3$. Suppose that Assumptions 6.3.1–6.3.2 hold. Assume that f is a strongly convex function such that $\nabla_x f(x) \succ \check{c} I_n \succ 0$ for some constant $\check{c} > 0$. Let the sequence $\{x_k\}_{k \geq 0}$ be generated by Algorithm 6.2.1. Then, for $k \geq 0$, we have*

$$f(x_k) - f_* \leq \hat{\nu} \left((p+1)^{\frac{1}{p}} + \frac{k-1}{p+1} \right)^{-p} \leq \hat{\nu} \left(\frac{p+1}{k} \right)^p \tag{6.65}$$

where $f_* = \min_{x \in \mathbb{R}^n} f(x)$ and $\hat{\nu} := (L + D_\Lambda \sigma_{\max}) D_\Lambda^{p+1} > 0$ is a constant independent of k and ϵ_g. For odd p, $\hat{\nu}$ is an iteration-independent and ϵ_g-independent constant. For even p, $\hat{\nu} = (L + D_\Lambda \tilde{C}_R {\epsilon_g}^{-1}) D_\Lambda^{p+1} = \mathcal{O}({\epsilon_g}^{-1})$.

Proof. This proof is adapted from [137, Thm 2] and proof techniques follow closely from [137, Thm 2]. Under the strongly convex assumption, $H_k \succeq \check{c} I_n \succ 0$ is always positive definite, the SoS-convex Taylor model will take the form in (6.1). We first prove (6.65) for the case of $k = 0$ and $x_k = 0$. Using Lemma 6.4.4, we derive

$$f(x_1) \leq \min_{s \in \mathbb{R}^n} \left[f(s) + \frac{(L + D_\Lambda \sigma_{\max})}{p+1} \|s\|^{p+1} \right] \underset{(6.63)}{\leq} f_* + \frac{(L + D_\Lambda \sigma_{\max})}{p+1} D_\Lambda^{p+1}. \tag{6.66}$$

Let $\hat{\nu} := (L + D_\Lambda \sigma_{\max}) D_\Lambda^{p+1} > 0$ with $\sigma_{\max}$ defined as in Lemma 6.4.4. Note that for odd p, $\hat{\nu}$ is an iteration and ϵ_g independent constant and for even p, $\hat{\nu} = (L + D_\Lambda \tilde{C}_R {\epsilon_g}^{-1}) D_\Lambda^{p+1} = \mathcal{O}({\epsilon_g}^{-1})$. From (6.66), we have

$$f(x_1) - f_* \leq \frac{\hat{\nu}}{p+1} < \hat{\nu} \left(\frac{p+1}{k} \right)^p \tag{6.67}$$

which satisfies (6.65) for $k = 1$ and $p \geq 3$. Further for $k \geq 1$, let $x_* = \operatorname{argmin}_{x\in\mathbb{R}^n} f(x)$, $f_* = f(x_*) = \min_{x\in\mathbb{R}^n} f(x)$ and $s_* = x_* - x_k$, using Lemma 6.4.4, we have

$$\begin{aligned} f(x_{k+1}) &\leq \min_{s\in\mathbb{R}^n}\left[f(x_k+s) + \frac{(L+D_\Lambda\sigma_{\max})}{p+1}\|s\|^{p+1}\right] \leq \min_{\alpha\in[0,1]}\left[f\big[x_k+\alpha s_*\big] + \frac{(L+D_\Lambda\sigma_{\max})}{p+1}\|\alpha s_*\|^{p+1}\right] \\ &\underset{(6.63)}{\leq} \min_{\alpha\in[0,1]}\left[f\big[(1-\alpha)x_k + \alpha(x_k+s_*)\big] + \frac{(L+D_\Lambda\sigma_{\max})}{p+1}D_\Lambda^{p+1}\alpha^{p+1}\right] \\ &\leq \min_{\alpha\in[0,1]}\left[f(x_k) - \alpha[f(x_k)-f_*] + \frac{\hat{\nu}}{p+1}\alpha^{p+1}\right]. \end{aligned}$$

The last inequality is obtained using the convexity of f. The minimum of the above objective in $\alpha \geq 0$ is achieved for

$$\alpha_* = \left(\frac{[f(x_k)-f_*]}{\hat{\nu}}\right)^{\frac{1}{p}} \leq \left(\frac{[f(x_1)-f_*]}{\hat{\nu}}\right)^{\frac{1}{p}} \underset{(6.67)}{\leq} \big(\frac{1}{p+1}\big)^{\frac{1}{p}} < 1.$$

Thus, we conclude that $f(x_k+s_k) \leq f(x_k) - \frac{p\alpha_*}{p+1}[f(x_k)-f_*]$. Denoting $\Delta_k = f(x_k) - f_*$, we get the following estimate

$$\Delta_k - \Delta_{k+1} = f(x_k) - f(x_{k+1}) \geq \frac{p\alpha_*}{p+1}\Delta_k = C\Delta_k^{\frac{p+1}{p}}$$

where $C = \frac{p}{p+1}\hat{\nu}^{-1/p}$. Let $\mu_k = C^p\Delta_k$. Following a similar recursive inequality analysis as in [137, Thm 2], we deduce that

$$\mu_k^{-1} \geq \left(\mu_1^{-1/p} + (k-1)p^{-1}\right)^p \qquad \text{and} \qquad \mu_1^{-1/p} = C^{-1}\Delta_1^{-1/p} \underset{(6.67)}{\geq} \frac{1}{p}(p+1)^{\frac{p+1}{p}}.$$

Substituting these two inequalities into $\mu_k = C^p\Delta_k$, we have

$$\begin{aligned} \Delta_k &= C^{-p}\mu_k \leq \left(\frac{p+1}{p}\right)^p\hat{\nu}\left(\mu_1^{-1/p} + (k-1)p^{-1}\right)^{-p} \leq \left(\frac{p+1}{p}\right)^p\hat{\nu}\left(\frac{1}{p}(p+1)^{\frac{p+1}{p}} + (k-1)p^{-1}\right)^{-p} \\ &\leq \hat{\nu}\left((p+1)^{\frac{1}{p}} + \frac{k-1}{p+1}\right)^{-p} \geq \hat{\nu}\left(\frac{p+1}{k}\right)^p. \end{aligned}$$

□

Remark 6.4.7. *The assumptions for Theorem 6.4.5 closely resemble those in [137, Theorem 2]. But Theorem 6.4.5 requires f to be strongly convex, whereas [137] only requires f to be convex.*

Remark 6.4.8. ***(Complexity bound for strongly convex objection functions.)*** *The global convergence rates provided by Theorem 6.4.5 can be extended to establish a complexity bound for function and derivative evaluations when seeking the ϵ_g-approximate minimum value of the convex functions. Similar to the approach in [86, Thm 3.3] and [87, Thm 3.4], we bound (6.65) by the tolerance ϵ_g,*

$$f(x_k) - f_* \leq \hat{\nu}\left(\frac{p+1}{k}\right)^p \leq \epsilon_g$$

where $\hat{\nu}$ is a constant for odd p and $\hat{\nu}$ has order $\mathcal{O}({\epsilon_g}^{-1})$ for even p. Therefore, we obtained the evaluation complexity bound of $\mathcal{O}({\epsilon_g}^{-\frac{1}{p}})$ for odd p, and the evaluation complexity bound of $\mathcal{O}({\epsilon_g}^{-\frac{p+1}{p}})$ for even p. While Theorem 6.4.5 specifies the convergence rate in successful iterations, by utilizing Lemma 6.4.1, we can account for both successful and unsuccessful iterations to determine the total number of function and derivative evaluations required by Algorithm 6.2.1.

Remark 6.4.9. *Note that in all three cases, we have $\bar{H}_k \succeq \delta I_n$. If one only enforces the condition $\bar{H}_k \succ 0$, [1, Lemma 4] uses the quadrature rule to prove that a unique minimizer exists for the subproblem. This is sufficient to establish the pth-order rate of convergence [1, Thm 4]. The parameter δ is needed to quantify the size of the regularization parameter and the decrease per iteration. These quantities together contribute to the overall complexity bound of the algorithm.*

6.5 Numerical Results for Chapter 5 and Chapter 6

In this chapter, we proved the global convergence of the iterative algorithm based on SoS convex Taylor models Algorithm 6.2.1. Using the result from Chapter 5 that sufficiently regularized nonnegative nonconvex quartic polynomials are also SoS, we propose an alternative SoS-based AR3 algorithmic framework. The resulting `AR3 + SoS` algorithm follows a similar adaptive update of σ as in Algorithm 6.2.1: at each iteration σ is increased until the AR3 subproblem becomes an SoS polynomial. Assuming uniqueness of the global minimizer, the exact solution of (Quartically Regularized Cubic Model) can then be recovered via SDP in polynomial time.

The practical variant of this algorithmic framework is given in Algorithm 6.5.1. Compared to the `AR3 + SoS Convex` method—where an additional quadratic regularization term must be introduced to make a nonconvex m_3. `AR3 + SoS`—our approach directly minimizes the original third-order Taylor model plus a quartic regularization term. In `AR3 + SoS Convex`, the subproblem may deviate from the true Taylor model once the quadratic regularization is added, causing the model to lose fidelity to the second- and third-order structure of f. In contrast, the proposed `AR3 + SoS` framework preserves the full third-order Taylor model, providing a more accurate local approximation in nonconvex regions. Our goal in this section is to provide preliminary evidence that sufficiently regularized AR3 subproblems can be reformulated as SoS polynomials and solved globally via SDP without introducing an additional quadratic regularization term. We emphasize that it is used only for numerical illustration; the global convergence of this variant, as well as the uniqueness of the subproblem minimizer, remains an open question.

Algorithm 6.5.1: Adaptive Third Order Regularization Method with SoS Taylor Models (AR3+SoS)

Initialization: Choose $x_0 \in \mathbb{R}^n$, Set $\sigma_0, \sigma_{\min} > 0$, $\gamma_2 > 1 > \gamma_1 > 0$, $\rho_2 > \rho_1 > 0$.
Input: x_0, $f_0 := f(x_0), g := \nabla_x f(x_0), H := \nabla_x^2 f(x_0), T := \nabla_x^3 f(x_0)$, $\sigma_k := \sigma_0$. Tolerance ϵ s.t $0 < \epsilon \ll 1$, max iteration $k_{\max}$, $k := 0$.
Step 1: Test for termination. If $\|\nabla_x f(x_k)\| \leq \epsilon$ or $k \geq k_{\max}$, terminate and output $s_* := s_k$.
Step 2: Step computation. Compute $s_k = \operatorname{argmin}_{s \in \mathbb{R}^n} m_3(s)$ by SoS/SDP solver (such as `SoSTOOLS`) where m_3 defined in (Quartically Regularized Cubic Model) with coefficients, $f_0 = f(x_k)$, $g = \nabla_x f(x_k)$, $H = \nabla_x^2 f(x_k)$, $T = \nabla_x^3 f(x_k)$, $\sigma = \sigma_k$.
if *Moment SDP solved successfully (`status==1`)* **then**
  Extract one global optimal solution, s_k.
else
  Increase regularization: σ_k by $\sigma_k := \gamma_1 \sigma_k$ and repeat Step 2.
Step 3: Acceptance of trial point. Compute $T_{3,x_k}(s_k)$ and define $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{3,x_k}(s_k)}$.
Step 4: Regularization parameter update.
if $\rho_k > \rho_1$ **then**
  Successful Iter. $x_{k+1} := x_k + s_k$;
else if $\rho_k > \rho_2$ **then**
  Very Successful Iter. Decrease regularization: $x_{k+1} := x_k + s_k$; $\sigma_{k+1} := \max\{\gamma_2 \sigma_k,\ \sigma_{\min}\}$;
else if $\rho_k \leq \rho_1$ **then**
  Unsuccessful Iter. $x_{k+1} := x_k$; Increase regularization: $\sigma_{k+1} := \gamma_2 \sigma_k$;
Repeat with $k := k + 1$.

6.5.1 Preliminary Numerical Results on MGH Test Set

We present preliminary numerical results on classical unconstrained minimization problems from the Moré, Garbow, and Hillstrom (MGH) test set [129], following the setup in Section 4.5.2. We compare the performance of the following algorithms:

- `ARC`: Adaptive Regularization with Cubics [46].

- `AR3 + ARC`: Third-order method with `ARC` as the inner solver [53].
- `AR3 + SoS Convex`: AR3 with an SoS-convex subproblem solver Algorithm 6.2.1.
- `AR3 + SoS`: AR3 with an SoS-based Taylor subproblem solver (Algorithm 6.5.1).

Throughout this section, unless otherwise stated, the parameters for all methods were set to $\rho_1 = 0.1$, $\rho_2 = 0.9$, $\eta = 0.1$ and $\gamma_1 = 0.5$, and $\gamma_2 = 2$, $\delta = 0.1$, $\sigma_0 = 100$. The inner solvers used for each method were `mcm` from [45, Algorithm 6.1] for `ARC`, the `ARC` algorithm itself for `AR3 + ARC`, and the `MSDP/MSOL` moment-SDP solver from GloptiPoly 3 [94] with `SEDUMI` for `AR3 + SoS Convex` and `AR3 + SoS`. The outer-loop termination tolerance was set to $\|\nabla f\| < 10^{-3}$, and the maximum iteration count at 3000. If an inner iteration was used, its tolerance was fixed at 10^{-6} and the maximum inner iteration count, at 3000. Table 6.2 provides the numerical details for each problem. In performance tables, `eval.` refers to derivative evaluations (successful iterations only) and `iter.` denotes the total number of function evaluations (successful and unsuccessful). Performance profiles were used to compare the algorithms across the test set, following the framework of [69]. We adopt the notation of [20]: $\Gamma_{\hat{m}}(\tau_m)$ is the proportion of problems for which method $\hat{m}$ attained an ϵ_g-approximate solution within a computational cost no more than τ_m times the cost of the most efficient method, where the cost is measured in iteration or evaluation counts[9]. Formally, if $t_{\hat{m}\hat{j}}(\epsilon_g)$ denotes the cost incurred by method $\hat{m}$ on problem $\hat{j}$ to reach an ϵ_g-approximate solution, we define $t^{\hat{j}}_{\min}(\epsilon_g) := \min_{\hat{m}} t_{\hat{m}\hat{j}}(\epsilon_g)$. where $\hat{m} \in \{\texttt{ARC}, \texttt{AR3+ARC}, \texttt{AR3+SoS Convex}, \texttt{AR3+SoS}\}$. If a method fails to satisfy the accuracy requirement within the iteration limit, we set $t_{\hat{m}\hat{j}}(\epsilon_g) = +\infty$.

Table 6.2: Comparison of AR3+SoS Convex (Algorithm 6.2.1), **AR3+SoS** (Algorithm 6.5.1), ARC, and AR3+ARC across 20 test problems. $*$ indicates that the algorithm failed to make progress after 3000 iterations. The Iter. and Eval. for `AR3+SoS` and `AR3+SoS Convex` exclude the unsolved SoS problems—these columns refer specifically to function evaluations and derivative evaluations. Problems marked with $*$ are excluded from the average count. † indicates that the algorithm timed out: $\|\nabla f\|$ was reduced below 10^{-2} but did not reach the prescribed outer-loop tolerance of 10^{-3}.

Problem	**Dim**	**AR3 + SoS C**		**AR3 + SoS**		**ARC**		**AR3 + ARC**	
	n	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	2	2656	2305	2656	2305	2317	2317	2354	2311
2	2	96	84	96	84	70	70	58	57
3	2	29	12	27	10	26	13	31	12
4	2	*	*	*	*	1372	1372	32	31
5	2	67	59	67	59	57	57	58	57
6	2	113	100	113	100	104	104	119	99
7	3	815	711	792	691	779	779	717	716
8	3	825	723	825	723	707	707	726	725
9	3	19	11	18	11	24	17	17	12
10	3	*	*	*	*	*	*	*	*
11	3	51	41	56	47	29	28	50	40
12	3	26	26	26	26	24	24	26	25
13	4	231	207	231	207	209	209	208	206
14	4	93	84	90	81	53	40	119	63
15	4	230	200	22	22	24	22	23	22
16	4	121	121	121	121	91	91	92	85
17	5	335	288	335	288	3000†	2988	173	148
18	6	16	16	17	17	13	13	17	16
19	11	167	155	152	140	96	83	831	800
20†	6	3000	2605	3000	2605	3000	3000	3000	3000
Average		**494**	**431**	**480**	**419**	**631**	**628**	**455**	**443**

All test problems have dimensions ranging from $n = 2$ to $n = 11$, and in this range, the moment-SDP solver can cope with all subproblem dimensions. Across the test set, all methods consistently converged to the same minimizer whenever convergence happened, and each computed solution satisfied second-order necessary conditions. As can be seen in Table 2.2, on average[10] `AR3 + SoS` required

[9] Some comments regarding CPU time are given at the end of the section.
[10] These average counts exclude problems marked with $*$ in Table 2.2 for the respective algorithms.

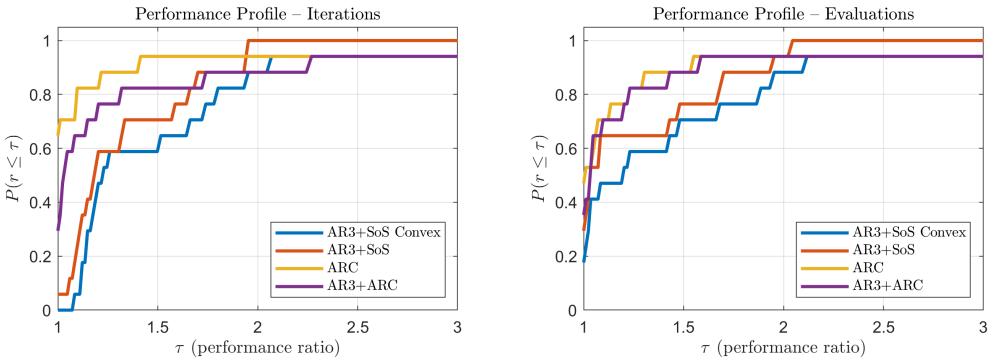


Figure 6.1: Performance profile plots for the three methods (ARC, AR3+ARC and AR3-SoS convex).

480 total iterations and 419 derivative evaluations, which is less than `ARC` (631 and 628) and comparable to `AR3 + ARC` (455 and 443) and `AR3 + SoS Convex` (494 and 431).

Across almost all test problems, `AR3 + SoS` requires either the same or slightly fewer iterations and derivative evaluations than `AR3 + SoS Convex`. This outcome is expected: `AR3 + SoS` minimizes the regularized third-order Taylor model and therefore captures the local structure more accurately, whereas `AR3 + SoS Convex` adds a quadratic regularization term in nonconvex regions to enforce SoS-convexity, causing the model to deviate from the true Taylor approximation. Nevertheless, several MGH problems are either convex or only mildly nonconvex, and therefore the performance of the two SoS-based methods is similar on these instances. For many problems (e.g., Problems 6, 8, 9, 11, 13), the iterates remain entirely within a locally convex region. For others (e.g., Problems 5, 7, 12, 18), only the initial iterate is locally nonconvex, after which all subsequent iterates become locally convex. In these problems, the performance of `AR3 + SoS` and `AR3 + SoS Convex` is almost identical. However, a particularly interesting case is Problem 15, where the Hessian is nearly singular at every iteration. In this setting, `AR3 + SoS Convex` requires roughly 200 iterations and reaches a maximum regularization value of $\sigma \approx 9000$. Also, in this example, the magnitude of the regularization parameter in the iterates of `AR3 + SoS Convex` is highly sensitive to δ. This behaviour is theoretically supported: as indicated in (5.3.3.1), the bound required on σ to ensure that the subproblem in the `AR3 + SoS Convex` method becomes SoS-convex explicitly depends on δ. In contrast, `AR3 + SoS` converges in only 20 iterations with a much smaller maximum regularization of $\sigma \approx 90$.

Problem 4 (Brown's badly scaled problem) has the minimizer at $[10^4, 1]^T$, and the problem is extremely ill-scaled. For `ARC` and `AR3 + ARC`, the algorithm typically proceeds with $\sigma < 50$. For `AR3 + SoS` and `AR3 + SoS Convex`, as given in (5.22), the SoS threshold for σ depends on $\|g\|$, $\|H\|$, $\|T\|$, which in this problem contains entries of magnitude 10^4. Consequently, the SoS-enforcing σ lies between 10^4 and 10^6. The algorithms remain numerically stable under this range of σ and continue to make progress, although with smaller step sizes toward $[10^4, 1]^T$. A simple change-of-variables scaling (see Section 3.3.2) would substantially reduce the required regularization. On the other hand, `AR3 + SoS Convex` performs well on Problem 19, requiring only 155 evaluations compared with 800 for `AR3 + ARC`. In this case, the key difference is that the SoS-convex model identifies a better subproblem minimizer that yields larger objective descent than the one obtained by the ARC inner solver.

Problem 10 and Problem 20 did not converge within the 3000 iteration limit for any of the methods. Problem 10 is known to be severely ill-conditioned, consistent with previously reported numerical difficulties [20]. In Problem 20, all methods, not only the SoS-based ones, generate shrinking steps. When the 3000-iteration limit is reached, all solvers satisfy $\|\nabla f(x_k)\| \approx 10^{-2}$ and are clearly converging but have not yet reached high accuracy. Increasing the maximum iteration limit to 6000 allows `AR3 + ARC` and `AR3 + SoS Convex` to converge fully, but `ARC` still fails to satisfy the gradient tolerance. Thus, the stagnation is intrinsic to the problem geometry rather than a consequence of the SoS requirement or large σ.

Size of σ: A natural question is whether the regularization parameter σ must become extremely large in practice for the shifted AR3 subproblem to be SoS. Our numerical results indicate that this

is generally not the case. Figure 6.2, we see that across the MGH test set, the values of σ used by `AR3 + SoS` remain within a numerically stable range—typically between 10^1 and 10^6. These values are comparable to those produced by `AR3 + SoS Convex` and also broadly consistent with `AR3 + ARC`, except for Problem 4, which we will discuss in later paragraphs.

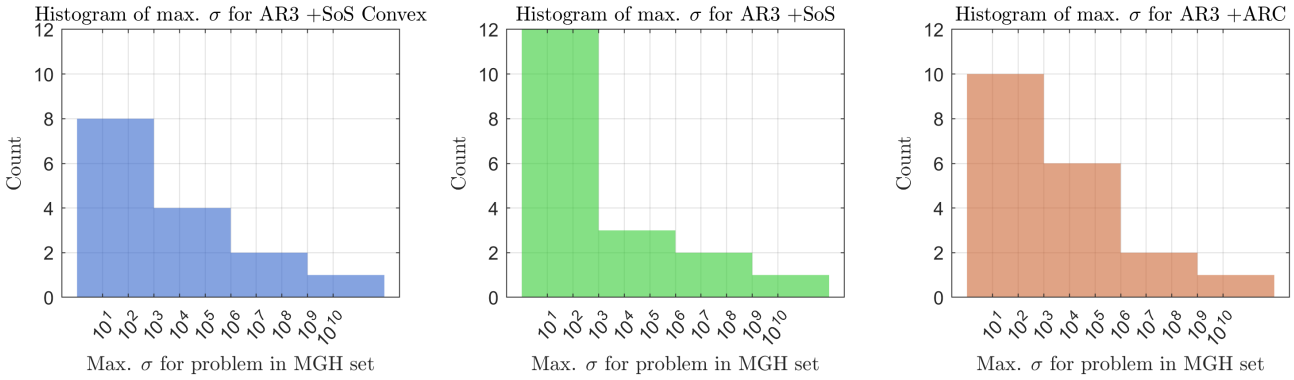


Figure 6.2: Histograms of the maximum regularization parameter σ attained for the three methods: `AR3 + SoS Convex` (left), `AR3 + SoS` (middle), and `AR3 + ARC` (right). The count on the vertical axis represents the number of MGH test problems whose maximum σ falls within each logarithmic scale bin. For most problems, the max. regularization remains within a numerically stable range ($10^1 \sim 10^6$). The absolute magnitude of the regularization parameter should be interpreted together with the scale of the local model coefficients. Normalizing solely by $\|g_k\|$ is not informative near convergence, since $\|g_k\| \to 0$ as the iterates approach stationarity. We therefore also compare the attained values of σ_k with the magnitudes of the Hessian and third-order tensor terms. The few instances requiring unusually large regularization parameters correspond predominantly to badly scaled problems with large values of $\|H_k\|$ and $\|T_k\|$, such as Problem 4.

Sensitivity of the Parameter δ in the AR3 + SoS-Convex Algorithm. We investigated the sensitivity of the parameter δ in the AR3 + SoS-Convex algorithm using Problem 11 (the GRD problem, $n = 3$). This problem features an ill-conditioned and nearly singular Hessian, and consequently many iterations naturally fall into the nearly convex regime where $0 < \lambda_{\min}(H_k) < \delta$. To examine how the choice of δ influences algorithm's behaviour, we varied δ from 10^{-9} to 1 and summarized the results in four diagnostic plots (Figure A.8 in Section A.4.3.2). First, the total iteration and evaluation counts remain remarkably stable across all tested values of δ. The AR3 + SoS-Convex method consistently converges in approximately 50–60 iterations, comparable to the performance of alternative solvers (roughly 20–30 for `ARC`, 40–50 for `AR3 + ARC`, and 50–60 for `AR3 + QQR`). Secondly, the number of iterations entering the locally nearly convex regime increases substantially as δ approaches 1, at which point most iterations satisfy $0 < \lambda_{\min}(H_k) < \delta$. Thirdly, the average magnitude of the regularization parameter σ grows only mildly with δ. Although the theoretical worst-case analysis suggests $\sigma = \mathcal{O}(\delta^{-2})$, our numerical experiments indicate that small choices of δ do not cause σ to become excessively large. Finally, the proportion of successful to unsuccessful steps remains stable across all values of δ. When δ is close to 1, the number of unsuccessful steps increases, as expected, since the SoS-convexified cubic model deviates further from the true local Hessian.

6.5.1.1 Two illustrative examples

Chebyshev-Rosenbrock. This function was introduced by Nesterov [98] and it is defined by

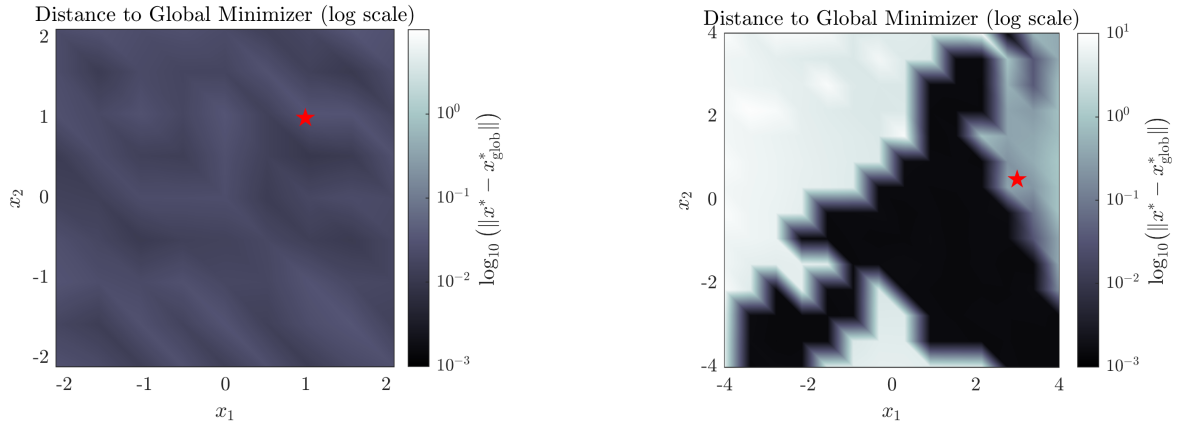
$$f_r(x) = 0.25(x_1 - 1)^2 + \sum_{i=1}^{n-1}(x_{i+1} - 2x_i^2 + 1)^2,$$

a fourth-degree polynomial. For $n = 2$, the Chebyshev–Rosenbrook function becomes $f_r(x) = 0.25(x_1 - 1)^2 + (x_2 - 2x_1^2 + 1)^2$, and is the form of Rosenbrock function[11]. The global minimizer

[11]Note that Problem 1 in the MGH test set corresponds to the Rosenbrock function with different coefficients scaling, which is $(x_1 - 1)^2 + 100(x_2 - x_1^2)^2$.

is $x^*_{\text{glob}} = (1, \ldots, 1)^T$ with $f_r(x^*) = 0$. At any other $x \in \mathbb{R}^n$, the gradient $\nabla f_r(x)$ is nonzero. Although the function has a unique stationary point (i.e., the global minimizer), the function contains many *near-stationary* regions with very small gradients, leading to slow convergence for first- and second-order methods. Although the function has a unique stationary point (i.e., the global minimizer), the function contains many *near-stationary* regions with very small gradients, leading to slow convergence for first- and second-order methods. Classical methods such as Newton's method and BFGS tend to follow a highly oscillatory manifold before reaching the minimizer, often requiring many iterations [91]. Consequently, f_r is a challenging benchmark for nonlinear optimization.

We apply both texttttAR3 + SoS Convex method and `AR3 + SoS` method to this function. The two methods give similar performance behaviour. Figure 6.3a shows a heat map of convergence behaviour over a fine grid of initializations in $[-4, 4]^2$ for `AR3 + SoS` method, where the color denotes the distance between the final iterate and the global minimizer. A result for `AR3 + SoS` method is given in Section A.4.3.3. Although each subproblem is solved globally via the moment-SDP solver, global convergence of the outer iteration is not guaranteed. Nonetheless, Figure 6.3a and Figure A.9b show that, for all initial points in the grid, the `AR3 + SoS` method and `AR3 + SoS Convex` consistently converge to x^*_{glob}, typically within fewer than 30 outer iterations. These preliminary results suggest that globally solving the subproblem can help steer the outer iteration toward the global minimizer.



(a) Basins of attraction for the Chebyshev–Rosenbrock.

(b) Basins of attraction for the Beale function

Figure 6.3: Global convergence behaviours for `AR3 + SoS` across different initializations. Dark regions indicate convergence to the global minimizer, while lighter regions indicate no convergence after reaching the maximum number of iterations.

We also compare `AR3 + SoS Convex` with `ARC` and `AR3 + ARC`. Figure 6.4 summarizes the performance of all methods for dimensions $n = 2, \ldots, 6$, averaged over five random initializations in $[-4, 4]^n$ (full numerical details are reported in Section A.4.3.4). The left panel shows that `AR3 + SoS Convex` achieves a high probability of converging to the global minimizer (i.e., 100% for $n = 2, \ldots, 5$) and remains competitive for $n = 6$. `ARC` and `AR3 + ARC` successfully identify the global minimizer for $n = 2, 3, 4$, but their success rates decrease as n increases to 5 and 6. The middle panel reports the proportion of runs that converge to nonglobal stationary points. `ARC` and `AR3 + ARC` variants increasingly settle at such points as the dimension grows. The right panel displays the average number of outer iterations on a logarithmic scale. The iteration counts of `AR3 + SoS Convex`, `ARC`, and `AR3 + ARC` are broadly comparable. However, `AR3 + SoS Convex` is more expensive in terms of CPU time per iteration due to the moment-SDP solve required to obtain the global optimum of each subproblem. We conjecture that solving each subproblem to global optimality may increase the likelihood that the outer iteration converges to the global minimizer. This is a challenging topic on its own, and further theoretical and numerical investigation is required. As a future direction, we plan to investigate this phenomenon further, using scalable SoS solvers (e.g., sparse SoS, DSoS/SDDSoS, BSoS [4; 128]), which would allow us to test this hypothesis on larger problem classes.

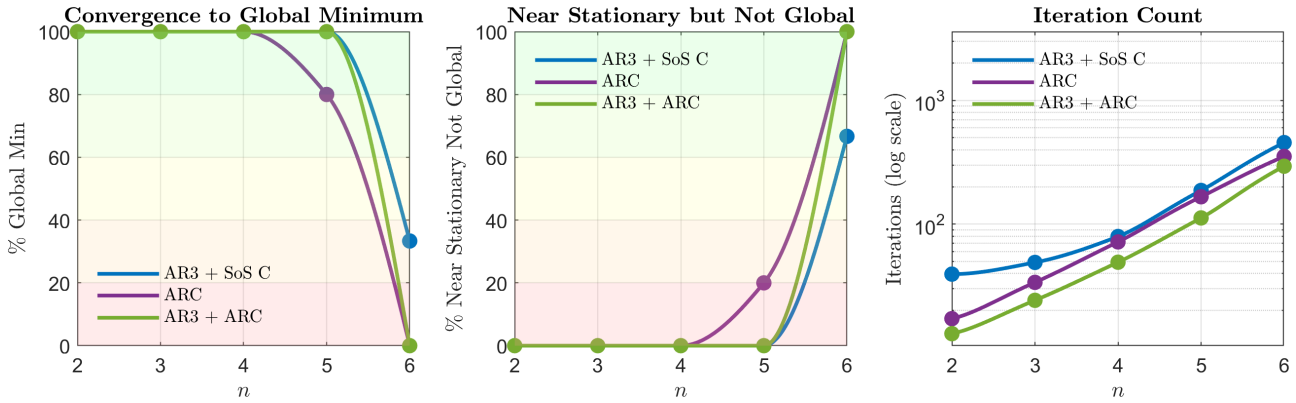
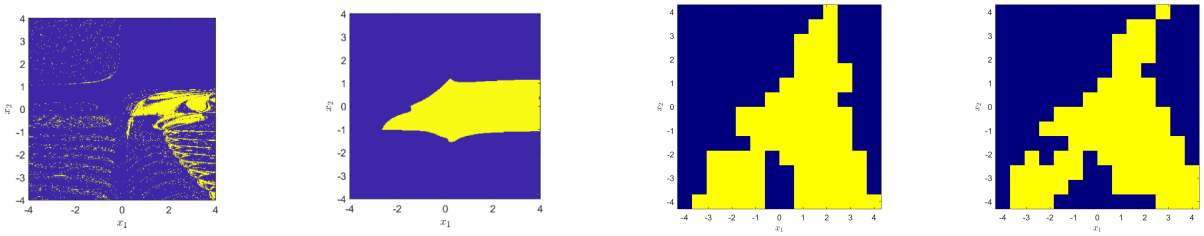


Figure 6.4: Comparison of all methods on the Rosenbrock-like test function in dimensions $n = 2, \ldots, 6$. **Left**: percentage of random initializations that converge to the global minimizer such that $\|x^k - x^*_{\text{glob}}\| \leq \text{Tol}$ where x^k is the outcome of the algorithm and $\text{Tol} = 10^{-2}$. **Middle**: percentage of runs that converge to near stationary regime (i.e, $\|\nabla f_r(x^k)\| < 10^{-6}$ and $\|x^k - x^*_{\text{glob}}\| > \text{Tol}$). **Right**: average number of iterations of the methods.

The Beale function. The Beale function, which is the fifth problem in the MGH test set [129],

$$f_b(x_1, x_2) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2.$$

The Beale function has three real critical points, including a unique global minimizer at $(3, 0.5)^\top$ and two saddle points. It is a standard benchmark for testing robustness to initialization. Although it has a unique global minimizer $\tilde{x}^*_{\text{glob}} = (3, 0.5)^T$ and no other local minima, its landscape is highly nonconvex, with long, narrow valleys and large flat regions that typically stall first- and second-order methods. Following the setup of [1, Sec. 5.2], we evaluate global convergence behaviour over a uniform grid on $[-4, 4]^2$, running each method for at most 350 iterations. In Figure 6.3b, dark regions correspond to initializations that converge (approximately) to the global minimizer. Lighter regions represent runs that fail to reach the minimizer and terminate only because the iteration budget is exhausted; in these cases, the distance to the minimizer remains $\mathcal{O}(1)$. Increasing the iteration cap from 3,000 to 20,000 iterations does not improve these non-convergent cases, as the iterates remain trapped in flat regions—consistent with the behaviour reported in [1, Sec. 5.2]. As shown in Figure 6.5, the region of initializations that converge to the global minimum (yellow area) `AR3+SoS` and `AR3+SoS Convex` are substantially larger than those of the classical Newton method[12], indicating an expansion of convergence regions when moving from second-order to third-order methods.



(a) Newton [1, Fig. 5] (b) 3rd-order Newton [1] (c) AR3+SoS Convex (d) AR3+SoS

Figure 6.5: Global convergence behaviours of different initializations for minimizing the Beale function (a) the classical Newton method, (b) the third-order Newton method, and (c) `AR3 + SoS Convex` with $\delta = 0.01$. Yellow indicates $\|x^k - \tilde{x}^*_{\text{glob}}\|/\|\tilde{x}^*_{\text{glob}}\| \leq 10^{-3}$ where x^k is the outcome after 350 iterations. Blue indicates runs that did not converge within the prescribed iteration budget. In these cases, the iterates typically became trapped in the narrow valleys or large flat regions of the Beale function, without converging to a critical point.

6.5.2 Sensor Network Localization Problem

Quartic optimization problems arise in various engineering applications such as independent component analysis [40], blind channel equalization in digital communication [122], and sensor network

[12]For Newton and the third-order Newton method, full numerical results appear in [1, Sec. 5.2].

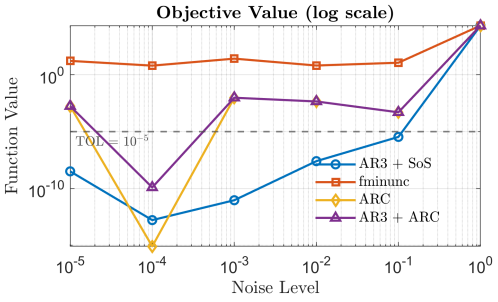
localization [28]. We consider the sensor localization problem, where $\mathcal{A}$ and $\mathcal{S}$ denote the sets of anchor nodes and sensor nodes, with cardinalities $|\mathcal{A}|$ and $|\mathcal{S}|$, respectively. The goal is to estimate the sensor positions $x_i \in \mathbb{R}^3$ for $i \in \mathcal{S}$ by minimizing the quartic objective

$$\min_{x_i \in \mathbb{R}^3,\, i \in \mathcal{S}} f_{\mathcal{S}}(x) := \sum_{i,j \in \mathcal{S}} \left(\|x_i - x_j\|^2 - d_{ij}^2 \right)^2 + \sum_{i \in \mathcal{S},\, j \in \mathcal{A}} \left(\|x_i - s_j\|^2 - d_{ij}^2 \right)^2, \tag{6.68}$$

where d_{ij} are (possibly noisy) distance measurements between nodes, s_j are the known positions of the anchors, and $x = [x_1^T, x_2^T, \ldots, x_{|\mathcal{S}|}^T]^T \in \mathbb{R}^{3|\mathcal{S}|}$. By homogenizing (6.68), one obtains a quartic minimization problem with a single quadratic constraint, fitting the structure of (5.48). As a resuult, (6.68) is known to be NP-hard [120][13]. The relative sizes of the sensor and anchor sets strongly influence the difficulty of the localization problem. When anchors outnumber sensors ($|\mathcal{A}| > |\mathcal{S}|$), the system is overdetermined and relatively easier. In practical settings, however, only a few anchors are available (e.g., GPS-equipped nodes), and the common regime is $|\mathcal{A}| \leq |\mathcal{S}|$. This underdetermined case is significantly more challenging: the resulting optimization landscape is highly nonconvex, often sensitive to measurement noise, and typically contains many spurious local minima. Moreover, due to geometric symmetries, multiple global minimizers may exist. For example, even in the simplest case with two anchors and two sensors, the problem admits several symmetric global solutions, and the rank condition required for moment-SDP solvers (e.g., GloptiPoly/MSDP) fails—preventing extraction of the global minimizer from the relaxation.

The quartic sensor–localization model is well suited for third-order methods. Although the problem is highly nonconvex, its third-order tensor has a structured (and often low-rank) form determined by the anchor geometry. Moreover, the third-order tensor can be applied to directions without explicitly forming the full dense tensor; see Section A.4.3.1 for details. We test four methods on this problem: `AR3 + SoS`, `fminunc`, `ARC`, and `AR3 + ARC`.

We try on the smallest "hard" case, with $|\mathcal{S}| = |\mathcal{A}| = 2$ a various level of noise $10^{-5} - 10^{0}$. In this setting, $f_{\mathcal{S}}$ is a quartic polynomial in $n = 2 \times 3 = 6$ variables. Our experiments show that, from the same initialization, `AR3+SoS` usually reduces the objective value the most giving near zero function value (below $O(10^{-5})$ for noise level $10^{-5} - 10^{-1}$). In contrast, `ARC`, `AR3 + ARC`, and `fminunc` typically converge to a higher objective function value, which correspond to a local minimizers. It is worth noting that, we also tried reformulate the 6–variable quartic directly (6.68) in SoS program and solve it with GloptiPoly/MSDP. However, at $|\mathcal{S}| = |\mathcal{A}| = 2$, the localization problem already have too much global minimizers, so the moment matrix in the Lasserre relaxation fails to satisfy the flat extension (rank) condition and a global minimizer cannot be extracted from the SDP solution. MSDP returning `status ==0` which means SoS relaxation solved correctly, but cannot recover the global minimizer since the rank condition fails. In contrast, the `AR3+SoS` algorithm converges numerically to an approximate global minimizer.



Noise Level	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{0}
`AR3 + SoS`	11/11	11/11	18/17	19/19	20/19	24/22
`fminunc`	8/ 6	12/11	9/ 7	18/10	15/11	13/ 7
`ARC`	17/12	27/19	20/17	19/16	26/18	8/ 7
`AR3 + ARC`	9/ 7	10/ 7	13/ 9	10/ 7	14/10	28/16

Figure 6.6: Comparison of four methods on the sensor–localization problem with 2 anchors and 2 sensors ($n = 6$ variables). Distances d_ij are perturbed by adding Gaussian noise of magnitude 10^{-6} to 10^{0}. All algorithms start from the anchor initialization. **Left:** Function values at x^* for different noise levels. **Right:** Iteration / evaluation counts for each method at the same noise levels.

[13]Indeed, even deciding whether the optimal value of (6.68) is zero is NP-hard.

Scalability and CPU time considerations. Our experiments are limited to relatively small-scale problems because the inner subproblems are solved using moment-SDP relaxations, which are practical only for small polynomial systems ($n < 10$). In comparison with other methods, `AR3 + SoS` is more expensive in terms of CPU time per iteration. Although SoS relaxations offer a principled route to global optimality, the SDP size $N = \binom{n+4}{4}$ limits their applicability. In settings where the polynomial subproblems admit structured sparsity, a sparsity-adapted hierarchy of relaxations can significantly reduce SDP size [39; 111; 128; 177]. Additionally, several scalable alternatives to full SoS relaxations have been proposed, such as (scaled) diagonally dominant SoS (DSoS/SDSoS) and bounded-degree SoS (BSoS) [4; 113; 177; 182; 189]. Replacing the SDP solves in Algorithm 1 with more scalable LP- or SOCP-based approaches or exploiting problem structure to generate smaller SDPs is a promising direction for making the approach applicable to larger-scale problems [4]. The focus of this chapter is on establishing a theoretical framework that enables a higher-order Newton-type algorithm to achieve global convergence guarantees while maintaining polynomial-time per-iteration complexity. In particular, our work—together with the recent results of Ahmadi [1]—provides one of the first guarantees that a third-order regularization scheme can be paired with a subproblem that is globally solvable in polynomial time. The development of scalable variants based on approximate SoS relaxations or structured sparsity is left for future work.

In conclusion, this chapter introduced an algorithmic framework that integrates the SoS Taylor model with adaptive regularization techniques. In each iteration, we minimize a SoS Taylor model which can be solved with polynomial cost per iteration. Also Specifically, for a polynomial with n variables and a degree of $2d$, the SDP's dimension is given by $N = \binom{n+2d}{2d}$, making it challenging to scale this approach to larger problems. This could cause challenges associated with the curse of dimensionality. The function value reduction in each strongly convex iteration is $\mathcal{O}(\epsilon_g^{\frac{p+1}{p}})$ for odd p, and $\mathcal{O}(\epsilon_g^{\frac{p+3}{p+1}})$ for even p. Note that we have function value reduction for $p = 3, 5$ at $\mathcal{O}(\epsilon_g^{\frac{2}{p}})$ and for $p = 4$ at $\mathcal{O}(\epsilon_g^{\frac{4}{5}})$. For a general nonconvex function, the overall worst-case evaluation complexity bound is $\mathcal{O}(\epsilon_g^{-2})$. For strongly convex functions $f(x)$, the evaluation complexity bound for the ϵ_g-approximate minimum value of the convex functions is $\mathcal{O}(\epsilon_g^{-\frac{1}{p}})$ for odd p, and the evaluation complexity bound of $\mathcal{O}(\epsilon_g^{-\frac{p+1}{p}})$ for even p.

Note that these bounds do not address sub-problem costs, nor do they match the optimal complexity of the pth order optimization method in nonconvex iterations in [41; 44]. Yet, given that determining the convexity of polynomials of degree at least four is NP-hard [5], SoS-convexity offers an effective means of verifying and classifying convexity. Moreover, finding the global minimum of a generally nonconvex ARp ($p \geq 3$) sub-problem is also NP-hard [36; 120]. Leveraging the SoS-convex Taylor model enables the minimization subproblem to be reformulated as semidefinite programming (SDP), rendering it tractable to achieve arbitrary precision within polynomial time. This study represents the first investigation into both global convergence and global complexity bounds for an adaptive regularization algorithm with a tractable high-order sub-problem ($p \geq 3$) in nonconvex smooth optimization, to the best of our knowledge, opening the way for further improvements.

Chapter 7

Conclusions and Future Work

This thesis investigates high-order regularization methods for nonconvex smooth optimization, with a particular focus on the structure, complexity, and global solvability of quartically regularized cubic models arising from adaptive regularization algorithms. By combining tools from numerical optimization, polynomial optimization, and algebraic geometry, we gain a deeper understanding of the ARp subproblem, the regularization strength, and the computational tractability of high-order methods. The thesis closes the research gaps by designing a series of iterative solvers for AR3 subproblem, characterizing the global optimality of the problem and investigating the potential NP-hardness of the subproblem.

We begin by studying models and algorithmic frameworks for approximating the AR3 subproblem, ranging from the quartic-quadratic regularization (QQR) model, through the cubic-quartic regularization (CQR) model, to the more general diagonal tensor model (DTM). QQR method is a second-order methods that aim to achieve high-accuracy minimization of nonconvex quartically-regularised cubic polynomials. This method can be viewed as a nonconvex extension of Nesterov's method proposed in [142], which was originally designed for convex m_3 functions. When applied to convex m_3 functions, the QQR method exhibits the same linear convergence profile as Nesterov's method. For locally convex iterations, the QQR method achieves either a linear rate of convergence in gradient norm or an error reduction of at least $\mathcal{O}(\epsilon^{4/3})$ per iteration. In the case of locally nonconvex iterations, the QQR method provides an error reduction of at least $\mathcal{O}(\epsilon)$. The CQR method iteratively minimizes the so-called CQR polynomial, a local quadratic model incorporating cubic and quartic terms. The cubic term has a coefficient β_i that captures scalar information about the local third-order tensor, while the quartic term controls the regularization and algorithmic progress. We have derived necessary and sufficient global optimality conditions that hold at a global minimizer of the CQR polynomial. Our theoretical analysis addressed the global convergence of the CQR method, showing that it is guaranteed to find a first-order critical point for the AR3 subproblem within at most $\mathcal{O}(\epsilon^{-3/2})$ function and derivative evaluations; here, ϵ represents the prescribed first-order optimality tolerance. The complexity bound for the CQR method is at least as good as that of ARC, and in specific cases, such as small third-order tensor term, it is better. The DTM framework minimizes the AR3 subproblem by iteratively solving a sequence of local quadratic models that incorporate both a diagonal cubic term and a quartic regularization term. Our theoretical analysis established the global convergence of the DTM algorithmic variant, proving it guarantees a first-order critical point for the AR3 subproblem within at most $\mathcal{O}(\epsilon^{-3/2})$ function and derivative evaluations, where ϵ represents the prescribed first-order optimality tolerance. For subproblems with diagonal or low-rank third-order tensor information, the DTM model was able to exactly recover the subproblem, and therefore converges in one iteration after the change of basis. From QQR to CQR to DTM, these models provide increasingly flexible approximations of the underlying third-order Taylor expansion, and are well-suited to different numerical regimes. In particular, Preliminary numerical results suggest that the QQR model offers simplicity and robustness, the CQR model is especially effective for rank-one or single-dominant-direction tensor structures, and the DTM approach performs better for diagonal or low-rank tensor subproblems.

We also addressed the research gap in characterizing the global optimality conditions for the AR3 subproblem. The global optimality characterization is derived for the general form of m_3, without

assuming any special structure on the tensor or the Hessian. In particular, for special cases such as zero, low-rank, or diagonal third-order tensor terms, the optimality conditions admit further simplifications. We identified the conditions under which the necessary and sufficient conditions coincide.

A central contribution of this thesis is the study of SoS representations for quartically regularized cubic polynomials. We establish the first results showing that, under sufficiently strong quartic regularization, a broad class of generally nonconvex shifted quartic polynomials becomes globally nonnegative and admits an SoS certificate. These results clarify how the strength of regularization influences convexification, algebraic certifiability, and global optimality, and identify the regimes in which exact SoS certificates are available.

Motivated by these structural results, we propose an algorithmic framework that integrates SoS-convex Taylor models with adaptive regularization techniques. At each iteration, an SoS-convex high-order model is minimized, yielding polynomial per-iteration complexity while retaining global convergence guarantees. To the best of our knowledge, this work provides the first global rate analysis for an adaptive regularization algorithm whose high-order subproblem is both tractable and certifiable in the nonconvex setting. This framework offers a bridge between high-order numerical optimization and polynomial optimization techniques.

In Table 7.1, we summarize all algorithms discussed in this thesis. Broadly speaking, QQR, CQR, and DTM belong to the same category of methods: they iteratively minimize the AR3 subproblem and are relatively scalable in practice. All three methods are convergent, with worst-case complexity guarantees at least as good as those of ARC, and with improvements observed in many cases. In Table 7.1, we also indicate the types of AR3 subproblems for which each iterative algorithm is most suitable[1]. Notably, for QQR, CQR, and DTM, we establish corresponding global optimality conditions for the associated polynomial models. On the other hand, the focus and aims of the SoS and SoS-Convex methods are different. These approaches are theoretical in nature and are concerned with whether the AR3 subproblem or convexified AR3 subproblem can be represented by an SDP of polynomial size. When the regularization parameter is sufficiently large and under mild assumptions, we show that the model can be expressed in SoS (or SoS convex) form.

Table 7.1: Summary of algorithms for solving the AR3 subproblem

Method	**Polynomial work per solve?**	**Suitable size**	**Suitable problem class**	**Regularization parameter**
QQR	No	Scalable	General AR3 subproblems	Adaptive via performance tests.
CQR	No	Scalable	AR3 subproblems with a dominant tensor direction, e.g., a **single large tensor entry or a rank-one tensor term**	Adaptive via performance tests.
DTM	No	Scalable	AR3 subproblems with **diagonal tensor terms or low-rank tensor structure**	Adaptive via performance tests.
SoS	Yes (see footnote for assumptions)	Small-scale problems	**Globally** solves the AR3 subproblem when the regularization is sufficiently large	Adaptive via performance tests; must also be large enough to ensure the model is SoS.
SoS Convex	Yes	Small-scale problems	Convexified AR3 subproblems with polynomial work per iteration to arbitrary accuracy	Similar to SoS; must be sufficiently large. A suitable σ can also be computed through SDP.

[1]For sufficiently large σ, and provided that the associated optimal moment matrix satisfies the required rank condition, the global minimizer s^* can be recovered to arbitrary prescribed accuracy from the moment relaxation.

7.1 Future Research Plan

The key numerical bottleneck in applying tensor methods is the computation and storage of the third-order tensor term. Since this term involves n^3 variables and is typically dense, direct computation becomes infeasible for large-scale problems. To mitigate the computational burden of calculating the exact third-order tensor, we will explore tensor approximation techniques that update the tensor using second-order information. Recently, Welzel and Hauser proposed a Generalized Quasi-Newton (GQN) update for higher-order derivative tensors [183]. The GQN method generates low-rank approximations that converge to the true third-order derivative under certain conditions. Using GQN methods and other advanced tensor approximation techniques within the QQR or CQR solvers to improve the scalability of the AR3 solver is an interesting direction for future research. Other potential techniques include the randomized subspace method, sketching [92; 127; 154], Tucker decomposition, and oblique projection [59], as well as CP decomposition and nonlinear model reduction techniques. Another promising avenue is exploring ways to exploit sparsity in the third-order tensor. We will also explore tensor compression techniques to minimize storage and computational costs for higher-order tensors.

Another avenue that we may explore is the global optimization of the objective function f. We also plan to extend the benefits of the high-order method from local to global optimization using the branch-and-bound algorithm and the Lip(γ) property of the function, as defined in [121, Def 2.2], which generalizes the classical notion of Lipschitz (or Hölder) continuity [173]. Branch-and-bound algorithms, which form the foundation of several general-purpose global optimization solvers, involve recursively partitioning the feasible region and constructing nonconvex quadratic or cubic lower bounds. These algorithms, as described in [43; 76; 96; 107; 147], aim to identify the global minimum through efficient bounding procedures. In ongoing joint work with Lyons and Walker, which is not included in this thesis, we investigate extensions of higher-order methods from local to global optimization using branch-and-bound frameworks together with the Lip(γ) property. The Lip(γ) property, as defined in [121, Def. 2.2], generalizes the classical notions of Lipschitz and Hölder continuity [173]. Branch-and-bound algorithms, which form the foundation of many general-purpose global optimization solvers, aim to identify global minima through rigorous bounding procedures by recursively partitioning the feasible region and constructing nonconvex cubic or higher order lower bounds [43; 76; 96; 107; 147].

During the four years of this PhD, alongside our study of the ARp subproblem, there are follow-up work demonstrated growing interest in this line of research. For instance, recent work by [190] leverages the global sufficient optimality conditions for cubic quartically regularized (CQR) polynomials [56] to derive a tight semidefinite programming (SDP) formulation of size $n + 1$. Another interesting direction is to exploit additional structural properties to develop more scalable variants using alternative relaxation techniques [113; 177; 182; 189]. While this thesis focuses primarily on the ARp method, there exists a broad class of other higher-order methods that are also of significant research interest, including higher-order trust-region methods [44; 105; 60; 104, Sec. 12]. A future direction is to explore the connections between the ARp subproblem and higher-order trust-region methods. In particular, the subproblem techniques developed in this thesis could be adapted and extended to design more effective solvers for higher-order trust-region frameworks. In addition to nonconvex optimization, higher-order methods can also be used for solving nonlinear least-squares problems, which have traditionally been addressed using Gauss–Newton or Newton-type methods. The tensor-Newton framework introduced in [81; 82] provides a higher-order perspective for such problems. The algorithmic developments from the ARp subproblem could be leveraged to solve the subproblem in tensor-Newton methods for nonlinear least-squares optimization.

Through our work on efficient subproblem solvers, global complexity analysis, and extensive numerical testing, this thesis helps bring higher-order methods closer to practical applicability. In particular, this PhD research narrows the research gaps by characterizing both local and global optimality conditions, investigating the potential NP-hardness of the ARp subproblem, and designing efficient algorithmic frameworks for its minimization. Overall, this work aims to bridge the gap between theoretical advances in high-order tensor methods and their real-world applications in nonconvex optimization.

Acknowledgments: This work was supported by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA).

Appendix A

Appendix

A.1 Appendix for Chapter 3

A.1.1 A Uniform Upper Bound for the Iterates

In this section, we prove that the iterates of Algorithm 3.3.1, including both successful and unsuccessful iterations, are uniformly bounded above by a constant r_c independent of the iteration count. The constant r_c is determined by the coefficients in m_3, specifically f_0, g, H, and T. We first require a technical lemma.

Lemma A.1.1. *Let M_c be defined as in (CQR Model) in Algorithm 3.3.1 with $d_i \geq 0$. Denote $\hat{s} := s^{(i)} + s$. If $\|\hat{s}\| = \|s^{(i)} + s\| \geq \|s^{(i)}\|$, then*

$$M_c(s^{(i)}, s) \geq \underbrace{f_0 + g^T[s^{(i)} + s] + \frac{1}{2}H[s^{(i)} + s]^2 + \frac{1}{6}T[s^{(i)} + s]^3}_{=T_3(\hat{s})} - \frac{1}{6}T[s]^3 + \frac{\beta_i}{6}\|s\|^3 + \frac{\sigma}{20}\left\|s^{(i)} + s\right\|^4.$$

Proof. Using the condition $\|s^{(i)} + s\| \geq \|s^{(i)}\|$, we obtain that

$$\|s^{(i)}\|^2 + \|s\|^2 + 2s^T s^{(i)} \geq \|s^{(i)}\|^2 \qquad \Rightarrow \qquad s^T s^{(i)} + \frac{1}{2}\|s\|^2 \geq 0. \tag{A.1}$$

Also, by Taylor expansion at $s = s^{(0)} = 0$,

$$\begin{aligned} f_i &= m_3(s^{(i)}) = f_0 + g^T s^{(i)} + \frac{1}{2}H[s^{(i)}]^2 + \frac{1}{6}T[s^{(i)}]^3 + \frac{\sigma}{4}\|s^{(i)}\|^4, \\ g_i^T s &= \nabla m_3(s^{(i)}) = g^T s + H[s^{(i)}][s] + \frac{1}{2}T[s^{(i)}]^2[s] + \sigma\|s^{(i)}\|^2 s^T s^{(i)}, \\ H_i[s]^2 &= \nabla^2 m_3(s^{(i)}) = H[s]^2 + T[s^{(i)}][s]^2 + \sigma\|s^{(i)}\|^2\|s\|^2 + 2\sigma[s^T s^{(i)}]^2. \end{aligned}$$

Substituting the above expressions into $M_c(s^{(i)}, s) = f_i + g_i^T s + \frac{1}{2}H[s_i]^2 + \frac{\beta}{6}\|s\|^3 + \frac{\sigma_c}{4}\|s\|^4$ and rearranging, we have

$$\begin{aligned} M_c(s^{(i)}, s) = f_0 + g^T(s^{(i)} + s) + \frac{1}{2}\Big(H[s^{(i)}]^2 + 2H[s^{(i)}][s] + H[s]^2\Big) + \\ + \frac{1}{6}\Big(T[s^{(i)}]^3 + 3T[s^{(i)}]^2[s] + 3T[s^{(i)}][s]^2 + \beta_i\|s\|^3\Big) + \\ + \frac{\sigma}{4}\Big(\|s^{(i)}\|^4 + 4\|s^{(i)}\|^2 s^T s^{(i)} + 2\|s^{(i)}\|^2\|s\|^2 + 4[s^T s^{(i)}]^2 + \|s\|^4\Big) + d_i\|s\|^4. \end{aligned}$$

Using $T[s^{(i)} + s]^3 - T[s]^3 = T[s^{(i)}]^3 + 3T[s]^2[s^{(i)}] + 3T[s^{(i)}]^2[s]$ and $\|s^{(i)} + s\|^4 = \|s^{(i)}\|^4 + 4\|s\|^2 s^T s^{(i)} +$

$2\|s\|^2\|s^{(i)}\|^2 + 4[s^T s^{(i)}]^2 + 4\|s^{(i)}\|^2 s^T s^{(i)} + \|s\|^4$, we deduce that

$$M_c(s^{(i)}, s) = f_0 + g^T[s^{(i)} + s] + \frac{1}{2}H[s^{(i)} + s]^2 + \frac{1}{6}\Big(T[s^{(i)} + s]^3 - T[s]^3 + \beta_i\|s\|^3\Big) + \frac{\sigma}{20}\big\|s^{(i)} + s\big\|^4$$
$$+\frac{\sigma}{5}\underbrace{\Bigg(\|s^{(i)}\|^4 + \underbrace{4\|s^{(i)}\|^2 s^T s^{(i)} + 2\|s\|^2\|s^{(i)}\|^2}_{\geq 0 \text{ by (A.1)}} + 4[s^T s^{(i)}]^2 + \|s\|^4 - \|s\|^2 s^T s^{(i)}\Bigg)}_{I_4} + \underbrace{d_i\|s\|^4}_{\geq 0}.$$

Note that

$$I_4 \geq \|s^{(i)}\|^4 + 4[s^T s^{(i)}]^2 + \|s\|^4 - \|s\|^2 s^T s^{(i)} \geq \|s\|^4 - \|s\|^3\|s^{(i)}\| + \|s^{(i)}\|^4.$$

If $\|s\| \leq \|s^{(i)}\|$, the sum of the last two terms is nonnegative. If $\|s\| \geq \|s^{(i)}\|$, the sum of the first two terms is nonnegative. In both cases, $I_4 \geq 0$, thus, we obtained the desired result. □

Now, we are ready to prove that the iterates generated by Algorithm 3.3.1 are uniformly bounded.

Theorem A.1.1. *Let M_c be defined as in (CQR Model) with $d_i \geq 0$ and $|\beta_i| \leq B$. Suppose that the CQR algorithmic framework (Algorithm 3.3.1) or its variants are used with $s^{(0)} = 0$ and $s_c^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_c(s^{(i)}, s)$. Then, for all $i \geq 0$,*

$$\big\|s^{(i)} + s_c^{(i)}\big\| < r_c := \max\left\{\left(\frac{60\|g\|}{\sigma}\right)^{1/3}, \left(\frac{30\|H\|}{\sigma}\right)^{1/2}, \left(\frac{80B}{\sigma} + \frac{90\Lambda_0}{\sigma}\right)\right\} \tag{A.2}$$

where $\|g\| = \sqrt{g^T g}$, $\|H\| = \max_i |\lambda_i(H)|$ is the maximum absolute value of eigenvalues of H and Λ_0 given in Definition 2.3.2.

Proof. We prove the desired result by induction. For $i = 0$, $s^{(0)} = 0$, $M_c(s^{(0)}, 0) = f_0$, $s_c^{(0)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_c(s^{(0)}, s)$. Clearly, $0 \geq M_c(s^{(0)}, s_c^{(0)}) - f_0$. Therefore,

$$0 \geq g^T s_c^{(0)} + \frac{1}{2}H[s_c^{(0)}]^2 + \frac{\beta_0}{6}\|s_c^{(0)}\|^3 + \frac{\sigma}{4}\big\|s_c^{(0)}\big\|^4 \geq -\|g\|\|s_c^{(0)}\| - \frac{\|H\|}{2}\|s_c^{(0)}\|^2 - \frac{B}{6}\|s_c^{(0)}\|^3 + \frac{\sigma}{4}\big\|s_c^{(0)}\big\|^4$$

where the second inequality uses the Cauchy-Schwarz inequality, norm properties, and $\beta_0 > -B$. We further deduce that

$$0 \geq \left(-\|g\| + \frac{\sigma}{12}\big\|s_c^{(0)}\big\|^3\right)\|s_c^{(0)}\| + \left(-\frac{\|H\|}{2} + \frac{\sigma}{12}\big\|s_c^{(0)}\big\|^2\right)\|s_c^{(0)}\|^2 + \left(-\frac{B}{6} + \frac{\sigma}{12}\big\|s_c^{(0)}\big\|\right)\|s_c^{(0)}\|^3.$$

The inequality above cannot hold unless at least one of the terms on the right-hand side is negative, which is equivalent to

$$\big\|s^{(0)} + s_c^{(0)}\big\| \leq \max\left\{\left(\frac{12\|g\|}{\sigma}\right)^{1/3}, \left(\frac{6\|H\|}{\sigma}\right)^{1/2}, \frac{2B}{\sigma}\right\} < r_c.$$

Thus, (A.2) is true for $i = 0$.

For the inductive hypothesis[1], assume (A.2) is true at the $(i-1)$th iteration $\big\|s^{(i-1)} + s_c^{(i-1)}\big\| \leq r_c$.

At the ith iteration, either we are in the good case, $\big\|s^{(i)} + s_c^{(i)}\big\| \leq \big\|s^{(i)}\big\|$ which leads to $\big\|s^{(i)} + s_c^{(i)}\big\| \leq \big\|s^{(i)}\big\| \leq r_c$ directly. Or, we are in the hard case, $\big\|s^{(i)} + s_c^{(i)}\big\| \geq \big\|s^{(i)}\big\|$. In this case, since $s_c^{(i)} = \operatorname{argmin}_{s\in\mathbb{R}^n} M_c(s^{(i)}, s)$, we have $0 \geq M_c(s^{(i)}, s_c^{(i)}) - M_c(s^{(i)}, 0) = M_c(s^{(i)}, s_c^{(i)}) - f_i \geq \ldots \geq M_c(s^{(i)}, s_c^{(i)}) - f_0$. Denoting $\hat{s} := s^{(i)} + s_c^{(i)}$, using Lemma A.1.1, we have

$$\begin{aligned} 0 \;&\geq\; M_c(s^{(i)}, s_c^{(i)}) - f_0 \geq g^T\hat{s} + \frac{1}{2}H[\hat{s}]^2 + \frac{1}{6}T[\hat{s}]^3 - \frac{1}{6}T[s_c^{(i)}]^3 + \frac{\beta_i}{6}\|s_c^{(i)}\|^3 + \frac{\sigma}{20}\big\|\hat{s}\big\| \\ &>\; -\|g\|\|\hat{s}\| - \frac{\|H\|}{2}\|\hat{s}\|^2 - \frac{\Lambda_0}{6}\|\hat{s}\|^3 - \frac{\Lambda_0}{6}\|s_c^{(i)}\|^3 - \frac{B}{6}\|s_c^{(i)}\|^3 + \frac{\sigma}{20}\big\|\hat{s}\big\|^4. \end{aligned}$$

[1]This is the same as $\big\|s^{(i)}\big\| = \big\|s^{(i-1)} + s_c^{(i-1)}\big\| \leq r_c$ if iteration is successful.

The last inequality uses the Cauchy-Schwarz inequality, norm properties, and $\beta_i > -B$. Since $\|\hat{s}\| \geq \|s^{(i)}\|$, this gives $\|s_c^{(i)}\| = \|\hat{s} - s^{(i)}\| \leq \|\hat{s}\| + \|s^{(i)}\| \leq 2\|\hat{s}\|$. Consequently,

$$\begin{aligned} 0 &> -\|g\|\|\hat{s}\| - \frac{\|H\|}{2}\|\hat{s}\|^2 - \frac{3\Lambda_0}{2}\|\hat{s}\|^3 - \frac{4B}{3}\|\hat{s}\|^3 + \frac{\sigma}{20}\|\hat{s}\|^4 \\ &= \left(-\|g\| + \frac{\sigma}{60}\|\hat{s}\|^3\right)\|\hat{s}\| + \left(-\frac{\|H\|}{2} + \frac{\sigma}{60}\|\hat{s}\|^2\right)\|\hat{s}\|^2 + \left(-\frac{4B}{3} - \frac{3\Lambda_0}{2} + \frac{\sigma}{60}\|\hat{s}\|\right)\|\hat{s}\|^3. \end{aligned}$$

The inequality above cannot hold unless at least one of the terms on the right-hand side is negative, which is equivalent to $\|s^{(i)} + s_c^{(i)}\| = \|\hat{s}\| \leq \max\left\{\left(\frac{60\|g\|}{\sigma}\right)^{1/3}, \left(\frac{30\|H\|}{\sigma}\right)^{1/2}, \left(\frac{80B}{\sigma} + \frac{90\Lambda_0}{\sigma}\right)\right\} = r_c$. □

Note that r_c is an iteration-independent bound and only depends on g, H, T, and B, which are fixed for m_3. The only requirement for Theorem A.1.1 is that $|\beta_i| \leq B$, where B is a positive constant (i.e., the coefficients of the cubic-order term of M_c are uniformly bounded), and $m_3(s^{(i)}) \leq m_3(s^{(0)})$. The proof of Theorem A.1.1 does not require the performance ratio test and is thus valid for both successful and unsuccessful iterations.

A.1.2 Proof of Lemma 3.3.3

Proof. For fixed i, the update of regularization parameters in Algorithm 3.3.2 gives that,

$$0 \leq d_{j+1} \leq d_j, \quad j \in \mathcal{S}_i, \qquad \gamma \max\{1, d_j\} = d_{j+1}, \quad j \in \mathcal{U}_i,$$

where $j \in [0:i]$. Using $d_0 = 0$, we inductively deduce that $\gamma^{|U_i|} = \max\{1, d_0\}\gamma^{|U_i|} \leq d_i$. Therefore, using our assumption that $d_i \leq d_{\max}$, we deduce that

$$|U_i| \log \gamma \leq \log(d_{\max}).$$

The desired result follows from the inequality $i = |U_i| + |S_i|$. □

A.1.3 Analysis on the Additional Condition (3.31)

Algorithm 3.3.2 has additional condition (3.31). Algorithm 3.4.1 does not have the additional requirement on $\sigma_c^{(i)}$, thereby allowing for a more adaptive choice of $\sigma_c^{(i)}$.

Figure A.1: Performance comparison between Algorithm 3.3.2 (blue line) and Algorithm 3.4.1 (black line) with the second choice of β_i and $\alpha = 0.1$: Algorithm 3.3.2 uses 14 total iterations while Algorithm 3.4.1 uses 13 total iterations. The differences in iterations between the two algorithms are with a red asterisk $*$. The setup of the test problems are in Section 3.4.1.1 and parameters in (2.59) are set at $\mathtt{a} = 80$, $\mathtt{b} = 80$, $\mathtt{c} = 80$, $\sigma = 5$, and $n = 100$.

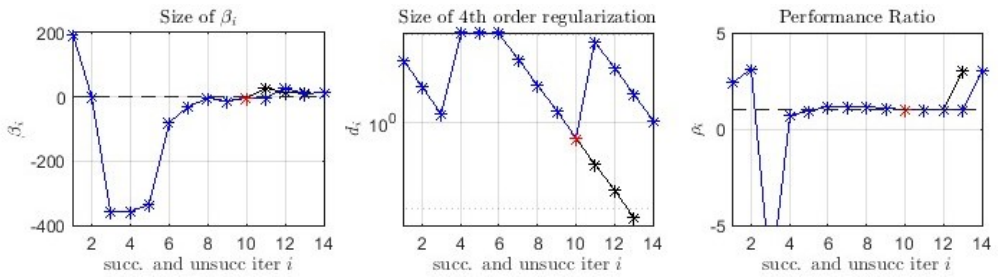
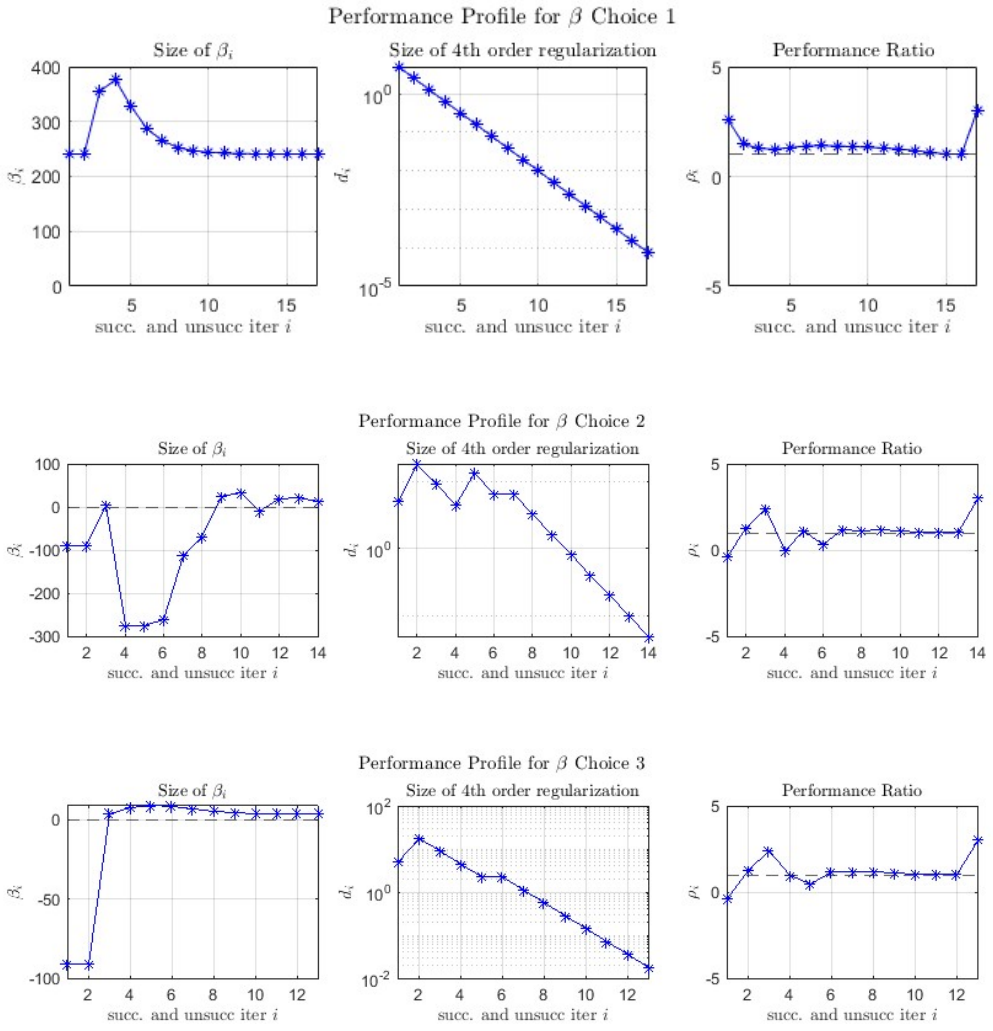


Figure A.2: Performance profile of Algorithm 3.4.1 using the different choices of β_i. The same test example is used as in Figure A.1.



A.1.4 Numerical Results for Section 3.4.1.2

The results presented in each table are obtained over 10 random problems randomized using MATLAB functions `rand()` or `randn()`. In all of the test cases, all algorithms converge to a point that satisfies $\|g_i\| < \epsilon$ with $\epsilon = 10^{-5}$. The *relative minimum values* of a method are obtained by dividing the minimum value achieved by this algorithm by the minimum of all the minimum values obtained by all algorithms.

Table A.1: **Convex m_3 and Convex H:** parameters in (2.59) are `a` $= 80$, and $\sigma = 80$. In the **first part of the table**, m_3 is convex, $H =$ `symm(30*(randn(n)+eye(n)))` and `c` $= 1$; In the **second part of the table**, m_3 is nonconvex but locally convex at $s = \mathbf{0}$, $H =$ `symm(30*(randn(n)+eye(n)))` and `c` $= 80$.

	Relative Min.			Iterations.			Evaluations.		
Method	$n = 5$	50	100	5	50	100	5	50	100
ARC	1	1	1	5	4	3	5	4	3
Nesterov	1	1	1	12.9	14	15	12.9	14	15
QQR-v1	1	1	1	6	7.3	7	6	7	7
QQR-v2	1	1	1	2.9	2	2	4	3	3
CQR-Choice 2	1	1	1	2.9	2	2	2.9	2	2
CQR-Choice 3	1	1	1	2.9	2	2	2.9	2	2
ARC	1	1	1	5	4	4	5	4	4
QQR-v1	1	1	1	6.5	7	7	6.5	7	7
QQR-v2	1	1	1	4	3	3	4	3	3
CQR-Choice 2	1	1	1	4.9	3	3	4.7	3	3
CQR-Choice 3	1	1	1	4.7	3	3	4.7	3	3

Table A.2: **Ill-Conditioned H, Diagonal H or Singular H:** parameters in (2.59) are `a` $= 80$, `c` $= 80$ and $\sigma = 80$. H is a diagonal ill-conditioned matrix with diagonal entries uniformly distributed in $[-10^{-10}, 10^{10}]$.

	Relative Min.			Iterations.			Evaluations.		
Method	$n = 5$	50	100	5	50	100	5	50	100
ARC	1	1	0.99033	5.3	6.5	5.3	5.3	6.2	5.3
QQR-v1	1	1	0.99033	9	9.7	10.2	8.8	8.6	9.2
QQR-v2	1	1	0.99033	4.1	4.8	4.7	4.1	4.8	4.5
CQR-Choice 2	1	1	1	4.9	7.5	6.5	4.4	5.3	4.7
CQR-Choice 3	1	1	0.99033	4.7	5.6	5.7	4.1	4.7	4.5

A.1.5 More Examples of Badly Scaled Subproblems

A.1.6 Alternative Proof for Lemma 4.2.2

Proof. By 4th order Taylor expansion of $m_3(s+v)$, we have

$$m_3(s+v) - m_3(s) = \left[m_3(s) + \nabla m_3(s)^T v + \frac{1}{2}\nabla^2 m_3(s)[v]^2 + \frac{1}{6}\nabla^3 m_3(s)[v]^3 + \frac{1}{24}\nabla^4 m_3(s)[v]^4 \right] - m_3(s).$$

Table A.3: **Changing σ:** From top to bottom, parameters in (2.59) are `a` $= 80$, `b` $= 80$, `c` $= 80$ and $\sigma = 10, 350$, respectively. The experiments show the algorithm's ability to handle a range of σ values. Generally, smaller values of σ for $m_3(s)$ require more function/derivative evaluations for convergence across all algorithms.

	Relative Min.			Iterations.			Evaluations.		
Method	$n = 5$	50	100	5	50	100	5	50	100
ARC	1	1	0.98313	11.8	21.7	26.3	8.1	15.1	20.5
QQR-v1	1	0.9958	1	14.5	19.6	22	12.5	17.3	19.5
QQR-v2	1	0.9958	0.97191	8.4	14	16.5	7.6	12.1	14.4
CQR-Choice 2	1	1	0.98498	8.2	15.8	17.5	7	12	15.5
CQR-Choice 3	1	1	0.97191	8.2	13.1	18.8	7.2	11.8	14.8
ARC	1	1	0.99642	12.5	18.4	19.2	6.5	8.5	12.2
QQR-v1	1	1	0.99825	9.9	12.6	16.5	8.5	11.8	15.3
QQR-v2	1	1	1	4.9	7.6	10	4.9	7.4	9.7
CQR-Choice 2	1	1	0.99642	4.9	10.7	11.2	4.9	7.1	9.6
CQR-Choice 3	1	1	0.99642	4.8	7	16	4.8	7	9.8

Table A.4: **Large Tensor Term and Small Regularization:** parameters in (2.59) are `a` $= 80$, `b` $= 80$, $\sigma = 5$, and `c` $= 300$ respectively. Minimizing $m_3(s)$ becomes progressively harder with a larger tensor term.

	Relative Min.			Iterations.			Evaluations.		
Method	$n = 5$	50	100	5	50	100	5	50	100
ARC	1	1	0.9845	16.7	24.3	35.1	9.1	15.8	27.1
QQR-v1	0.99204	0.94926	1	18.3	22.1	25.6	15.2	18.6	22.4
QQR-v2	1	1	0.96128	10.6	18.2	22.1	8.4	12.6	18
CQR-Choice 2	1	1	0.98301	10.8	15.4	22	8.3	13.1	19.1
CQR-Choice 3	1	1	0.98301	11.2	20.9	19.6	9	13.2	17.3

Table A.5: **Directional and Diagonal Tensors:** In the **first table**, g and T are negative and directional and H is near zero where `g = -80*rand(n,1)` and `T = -80*symm(rand(n,n,n))`. The parameters in (2.59) are `b` $= 0.1$ and $\sigma = 80$. In the **second table**, we consider positive and directional tensors, where parameters in (2.59) are `a` $= 80$, `b` $= 80$, $\sigma = 80$ and T is a diagonal tensor with entries that are uniformly distributed in the range of $[0, 40]$.

	Relative Min.			Iterations.			Evaluations.		
Method	$n = 5$	50	100	5	50	100	5	50	100
ARC	1	1	1	13.2	17.4	15.4	7.1	8.9	7.7
QQR-v1	1	1	1	11.7	19.3	21.1	10.5	15.3	16.6
QQR-v2	1	1	1	6.2	8.7	12.1	6.1	7.3	8.2
CQR-Choice 2	1	1	1	10.4	8.7	8	5.7	7.7	8
CQR-Choice 3	1	1	1	6.7	8.4	11.2	6.1	7.4	8
ARC	1	1	1	11.1	12.5	16.4	6.2	6.5	8
QQR-v1	1	1	1	10.1	10.5	10.4	8.7	8.9	8.4
QQR-v2	1	1	1	4.1	4.4	4.1	4.1	4.4	4.1
CQR-Choice 2	1	1	1	8.3	5.4	5.2	4.9	5.4	5.2
CQR-Choice 3	1	1	1	4.9	5.4	5.2	4.9	5.4	5.2

Figure A.3: **Examples with varying magnitudes of small entries:** In these tests, the setup is the same as in Figure 3.4 with $n = 99$. The left two plots depict scenarios where the other entries follow a standard normal distribution $N(0, 1)$. In the right two plots, all other entries, except the first one, are set to zero.

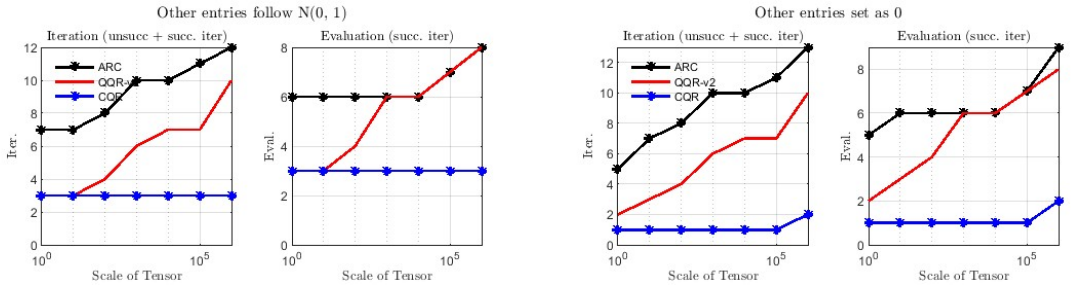
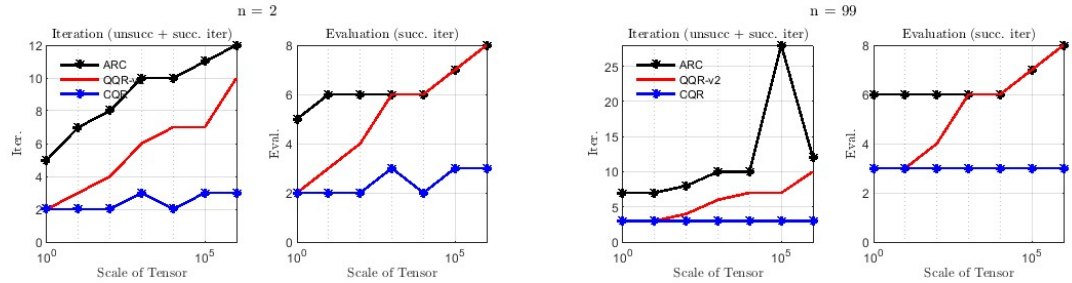


Figure A.4: **Examples with different β Choices:** These scenarios use the same setup as in Figure 3.4, but β_i is chosen according to Choice 3. Similar performance advantages are observed.



Let $\mathcal{B}(s)$ and $\mathcal{G}(s)$ be defined as in (4.1). Using $\nabla m_3(s) = g+\mathcal{B}(s)s$, $\nabla^2 m_3(s) = \mathcal{G}(s)+2\sigma(Ws_*)(Ws_*)^T$, $\frac{1}{6}\nabla^3 m_3(s)[v]^3 = \frac{1}{6}T[v]^3 + \|v\|_W^2 s^T W v$ and $\frac{1}{24}\nabla^4 m_3(s)[v]^4 = \frac{\sigma}{4}\|v\|_W^4$, we deduce that

$$\begin{aligned} m_3(s+v) - m_3(s) &= \left[g+\mathcal{B}(s)s\right]^T v + \frac{1}{2}\left[\mathcal{G}(s) + 2\sigma s W s^T\right][v]^2 + \frac{1}{6}T[v]^3 + \|v\|_W^2 s^T W v + \frac{\sigma}{4}\|v\|_W^4. \\ &= \left[g+\mathcal{B}(s)s\right]^T v + \frac{1}{2}\mathcal{G}(s)[v]^2 + \frac{1}{6}T[v]^3 + \frac{\sigma}{4}\left[\|s\|_W^2 - \|s+v\|_W^2\right]^2 \end{aligned}$$

which is equivalent to (4.7). □

A.2 Appendix for Chapter 4

A.2.1 Global Optimality Condition for (SQR)

In this section, we investigate the (global) optimality conditions of (SQR). If s_* is a local minimum, the first-order and second-order local optimality conditions of (SQR) yield

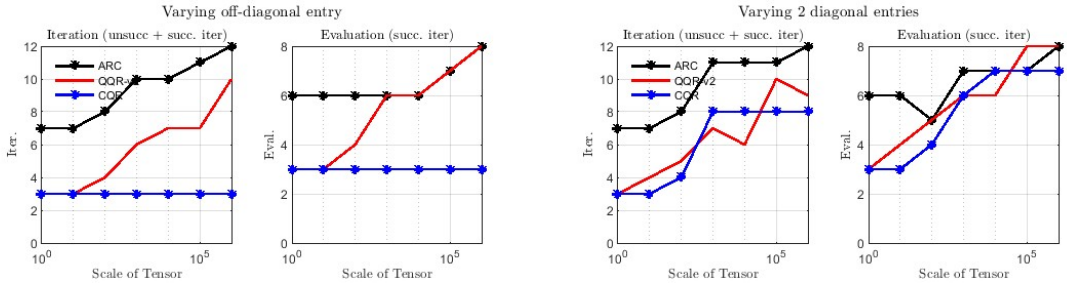
$$\nabla m_{\mathbf{SQR}}(s_*) = 0 \quad \Rightarrow \quad \hat{\mathcal{B}}(s_*)s_* := \left(H + \frac{1}{2}\mathcal{T}[s_*] + \Sigma[s_*]^2\right)s_* = -g_i, \tag{A.3}$$

$$\nabla^2 m_{\mathbf{SQR}}(s_*) \succeq 0 \quad \Rightarrow \quad H + \mathcal{T}[s_*] + 3\Sigma[s_*]^2 \succeq 0 \tag{A.4}$$

where $\Sigma[s]^2 := \operatorname{diag}\left\{\sigma_j s_j^2\right\}_{1\le j\le n} \in \mathbb{R}^{n\times n}$. Before proving the global optimality conditions for $m_{\mathbf{SQR}}$, we need the following technical lemma.

Lemma A.2.1. *Let $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_n]^T \in \mathbb{R}^n$, $t = [t_1, \ldots, t_n]^T \in \mathbb{R}^n$ and $\hat{\mathcal{B}}(s) := H + \frac{1}{2}\mathcal{T}[s] + \Sigma[s]^2$. For any vector $s, v \in \mathbb{R}^n$, we have (4.25).*

Figure A.5: **Examples with multiple large entries:** These tests maintain the same setup as in Figure 3.4, with $n = 99$. In the left two plots, apart from the first entry diagonal of T, the size of certain off-diagonal entries, such as $T[2, 3, 4]$ and its permutations, also varies from -0.1 to -10^5. In the right two plots, aside from the first diagonal entry of T, the size of two diagonal entries of T is also varied from -0.1 to -10^5.



Proof. We assume wlog $f_0 = 0$. For all vectors $v \in \mathbb{R}^n$, we have

$$
\begin{aligned}
m_{\mathbf{SQR}}(s+v) &= g^T(s+v) + \frac{1}{2}H[s+v]^2 + \frac{1}{6}\mathcal{T}[s+v]^3 + \frac{1}{4}\Sigma[s+v]^4, \\
&= \underbrace{g^T(s+v) + \frac{1}{2}\hat{\mathcal{B}}(s)[s+v]^2 - \frac{1}{12}\mathcal{T}[s]^3 - \frac{1}{4}\Sigma[s]^4 - \frac{1}{12}\mathcal{T}[s][v]^2 +}_{\hat{\mathcal{F}}_1} \\
&\underbrace{-\frac{1}{2}\left[\frac{1}{2}\mathcal{T}[s] + \Sigma[s]^2\right][s+v]^2 + \frac{1}{6}\mathcal{T}[s+v]^3 + \frac{1}{4}\Sigma[s+v]^4 + \frac{1}{12}\mathcal{T}[s]^3 + \frac{1}{4}\Sigma[s]^4 + \frac{1}{12}\mathcal{T}[s][v]^2}_{\hat{\mathcal{F}}_2}.
\end{aligned}
$$

Note that

$$
\begin{aligned}
\hat{\mathcal{F}}_1 &= g^T s + \frac{1}{2}\left[\hat{\mathcal{B}}(s) - \frac{1}{6}\mathcal{T}[s]^3 - \frac{1}{2}\Sigma[s]^2\right][s]^2 + \left[g + \hat{\mathcal{B}}(s)s\right]^T v + \frac{1}{2}\hat{\mathcal{B}}(s)[v]^2 - \frac{1}{12}\mathcal{T}[s][v]^2, \\
&= m_{\mathbf{SQR}}(s) + \left[g + \hat{\mathcal{B}}(s)s\right]^T v + \frac{1}{2}\hat{\mathcal{B}}(s)[v]^2 - \frac{1}{12}\mathcal{T}[s][v]^2. \qquad \text{(A.5)}
\end{aligned}
$$

Rearranging $\hat{\mathcal{F}}_2$ gives

$$
\hat{\mathcal{F}}_2 \;=\; \frac{1}{3}\mathcal{T}[s][v]^2 + \frac{1}{6}\mathcal{T}[v]^3 + \frac{1}{4}\Big[- 2\Sigma[s]^2[s+v]^2 + \Sigma[s]^4 + \Sigma[v+s]^4\Big]. \qquad \text{(A.6)}
$$

We add $\operatorname{diag}\{\frac{t_j^2}{36\sigma_j}\}[v]^2$ to (A.6) and give the sum of square expression,

$$
\begin{aligned}
\hat{\mathcal{F}}_2 + \operatorname{diag}\{\frac{t_j^2}{36\sigma_j}\}[v]^2 &= \sum_{j=1}^{n}\left[\frac{1}{3}s_j v^2 + \frac{1}{6}v_j^3 + \frac{\sigma_j}{4}\big[-2(s_j+v)^2 s_j^2 + s_j^4 + (v+s_j)^4\big] + \frac{t_j^2 v_j^2}{36\sigma_j}\right] \\
&= \frac{1}{4}\sum_{j=1}^{n}\left[\sigma_j v_j^2 (v_j + 2s_j + \frac{t_j}{3\sigma_j})^2\right] = \frac{1}{4}\left\|\boldsymbol{\sigma}^{\frac{1}{2}} \cdot v \cdot (v + 2s + \frac{t \cdot \boldsymbol{\sigma}^{-1}}{3})\right\|^2.
\end{aligned}
$$

Combining (A.5) and (A.6), we have,

$$
\begin{aligned}
m_{\mathbf{SQR}}(s+v) &= m_{\mathbf{SQR}}(s) + \left[g + \hat{\mathcal{B}}(s)s\right]^T v + \frac{1}{2}\hat{\mathcal{B}}(s)[v]^2 - \frac{1}{12}\mathcal{T}[s][v]^2 \qquad \text{(A.7)} \\
&\quad - \operatorname{diag}\{\frac{t_j^2}{36\sigma_j}\}[v]^2 + \frac{1}{4}\left\|\boldsymbol{\sigma}^{\frac{1}{2}} \cdot v \cdot (v + 2s + \frac{t \cdot \boldsymbol{\sigma}^{-1}}{3})\right\|^2. \qquad \text{(A.8)}
\end{aligned}
$$

Rearrange gives (4.25). □

Theorem A.2.1. *(Global Sufficient Condition) Assume (4.26) is satisfied. Then, s_* is a global minimum of $m_{\boldsymbol{SQR}}(s)$ over $\mathbb{R}^n$.*

Proof. Substituting (4.26) into (4.25) gives, for all $v \in \mathbb{R}^n$, $m_{\mathbf{SQR}}(s_* + v) - m_{\mathbf{SQR}}(s_*) \geq 0$. □

Note that the sufficient global optimality condition (4.26) implies the second-order local optimality condition (A.4). To see this, for all $v \in \mathbb{R}^n$, denote $s_* = [s_{*1}, \ldots, s_{*n}]^T$, then

$$\left[H + \mathcal{T}[s_*] + 3\Sigma[s_*]^2\right][v]^2 \underset{(4.26)}{\geq} \operatorname{diag}\left[2\sigma_j s_{*j}^2 + \frac{2}{3}t_j s_{*j} + \frac{t_j^2}{18\sigma_j}\right][v]^2 = 2\operatorname{diag}\left[\sigma_j\left(s_{*j} + \frac{t_j}{6\sigma_j}\right)^2\right][v]^2 \geq 0$$

Theorem A.2.2. *(Global Necessary Condition) Assume s_* is a global minimum of $m_{\boldsymbol{SQR}}(s)$ over $\mathbb{R}^n$. Then, (4.27) is satisfied.*

Proof. Since s_* is the global minimum we have $m_{\mathbf{SQR}}(s_* + v) - m_{\mathbf{SQR}}(s_*) \geq 0$ for all $v \in \mathbb{R}^n$. s_* is also the local minimum, therefore $\hat{\mathcal{B}}(s_*)s_* = -g$. If $2s_{*j} + \frac{t_j}{3\sigma_j} \neq 0$, let $\tilde{v} = \tilde{k}e_j := -(2s_{*j} + \frac{t_j}{3\sigma_j})e_j$. Then, $\tilde{v} \cdot (\tilde{v} + 2s_* + \frac{t \cdot \boldsymbol{\sigma}^{-1}}{3}) = 0$. Substituting $\hat{\mathcal{B}}(s_*)s_* = -g$ and $\tilde{v} = -\tilde{k}e_j$ into (4.25) in Lemma A.2.1, we have

$$0 \leq m_{\mathbf{SQR}}(s_* + \tilde{v}) - m_{\mathbf{SQR}}(s_*) = \frac{1}{2}\hat{\mathcal{G}}(s_*)[e_j]^2\tilde{k}^2, \qquad \forall j = 1, \ldots, n \tag{A.9}$$

where[2] $\hat{\mathcal{G}}$ is defined in (4.24). If $2s_{*j} + \frac{t_j}{3\sigma_j} = 0$, set $\tilde{v} = e_j$, the same results follows.

Now, take any unit vector $v = [v_1, \ldots, v_n]^T = \sum_{j=1}^n v_j e_j$, then

$$\begin{aligned}\hat{\mathcal{G}}(s_*)[v]^2 = & \sum_{j=1}^n v_j^2 \underbrace{\hat{\mathcal{G}}(s_*)[e_j]^2}_{\geq 0 \text{ by (A.9)}} + 2 \underbrace{\sum_{\iota,j=1,\iota\neq j}^n (v_\iota v_j) H[e_\iota][e_j]}_{:=\mathbf{H}} + \\ & + \sum_{\iota,j=1,\iota\neq j}^n 2(v_\iota v_j) \underbrace{\left[\frac{1}{3}\mathcal{T}[s_*] + \sigma_j\Sigma[s_*]^2 - \operatorname{diag}\left\{\frac{t_j^2}{18\sigma_j}\right\}\right][e_\iota][e_j]}_{=0}.\end{aligned} \tag{A.10}$$

Using $H[e_\iota][e_j] = |H_{\iota,j}|$, we deduce that

$$\mathbf{H} \geq -\max_{\|v\|=1} |v_\iota v_j| \sum_{\iota,j=1,\iota\neq j}^n |H_{\iota,j}| \geq -\|H_0\|_1 \tag{A.11}$$

where $\|H_0\|_1 = \sum_{\iota,j=1,\iota\neq j}^n |H_{\iota,j}|$. Substituting (A.11) in (A.10) gives the result. □

If the second order term H is a diagonal matrix, then $\|H_0\|_1 = \sum_{\iota,j=1,\iota\neq j}^n |H_{\iota,j}| = 0$. Consequently, the global necessary condition (4.27) and the global sufficient condition (4.26) coincide. Furthermore, if H is a diagonal Hessian, minimizing (SQR) simplifies to the minimization of n univariate CQR polynomials, as illustrated in Section 3.3.2.

A.2.2 Optimality of Subproblems in the AR3 algorithm

Assumption A.2.1. *(Lipschitz Property) Assume that the pth derivative ($p \geq 1$) of f at x is globally Lipschitz; that is, there exists a constant $L > 0$ such that, for all $x, y \in \mathbb{R}^n$, $\|\nabla_x^p f(x) - \nabla_x^p f(y)\| \leq (p-1)!L\|x - y\|_W$.*

Remark A.2.1. *There are slight differences in the definition of the Lipschitz constant across the literature. For instance, the Lipschitz constant in [137] differs from that in [53] by a constant factor of the order of p. The Lipschitz properties in [53; 137] are defined with respect to the Euclidean norm, specifically, $\|\nabla_x^p f(x) - \nabla_x^p f(y)\| \leq (p-1)!\tilde{L}\|x - y\|$. In contrast, Assumption A.2.1 uses the W-norm. The two definitions are related by the inequalities $\lambda_{\min}(W)\tilde{L}\|x-y\| \leq L\|x-y\|_W \leq \lambda_{\max}(W)\tilde{L}\|x-y\|$.*

[2] Note that $\hat{\mathcal{G}}(s_*)[e_j]^2 \geq 0$ for all $1 \leq j \leq n$ does not necessarily mean that $\hat{\mathcal{G}}(s_*) \succeq 0$, a counter example is $[0, 1; 1, 0]$.

Lemma A.2.2. *[53, Lemma 2.1 and Appendix A.1] Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ for $p \geq 2$ which means that f is p-times continuously differentiable, bounded below and the pth derivative of f is globally Lipschitz continuous, $p \geq 1$. $T_p(x,s)$ is the Taylor approximation of $f(x+s)$ about x. Then, under Assumption A.2.1, for all $x, s \in \mathbb{R}^n$,*

$$f(x+s) \leq T_p(x,s) + \frac{L}{p}\|s\|_W^{p+1}, \qquad \|\nabla_x f(x+s) - \nabla_s T_p(x,s)\| \leq L\|s\|_W^p. \tag{A.12}$$

Lemma A.2.3. *(Adapted from [53, Lemma 3.1]) Let $f \in \mathcal{C}^{p,1}(\mathbb{R}^n)$ for $p = 3$, meaning that f is p-times continuously differentiable, bounded below, and the pth derivative of f is globally Lipschitz continuous (Assumption A.2.1). If $\|\nabla_x f(x+s_*)\| > \epsilon_g$ for a positive tolerance ϵ_g, then, a first order critical point s_* satisfies $\|s_*\|_W \geq \left[\frac{\epsilon_g}{L+\sigma}\right]^{1/3}$.*

Proof. Since s_* is a first-order critical point, therefore $\nabla m_3(x_k, s_*) = 0$. We deduce that

$$\begin{aligned} \epsilon &< \|\nabla_x f(x+s_*)\| = \|\nabla_x f(x+s_*) - \nabla m_3(x_k, s_*)\| \\ &\leq \|\nabla_x f(x+s_*) - \nabla T_3(x, s_*)\| + \|\nabla m_3(x_k, s_*) \ - \nabla T_3(x, s_*)\| \\ &\leq L\|s_*\|_W^3 + \|\nabla m_3(x_k, s_*) \ - \nabla T_3(x, s_*)\| = (L+\sigma)\|s_*\|_W^3. \end{aligned}$$

where ∇ denote derivative with respect to s. Note that the last inequality uses Lemma 2.1 and Appendix A.1 [53] with slight modification to W-norm. □

Theorem A.2.3. *Assume that Assumption A.2.1 holds and that $\|g\| \geq \epsilon_g$ for some positive tolerance ϵ_g. Let $\lambda := \min\{0, -\lambda_{\min}[H]\}\lambda_{\min}(W)^{-1}$, and if σ satisfies*

$$\sigma \geq \max\left\{L, 4(3\lambda)^3\epsilon_g^{-2}, \sqrt{2}(7\Lambda_W)^{3/2}\epsilon_g^{-1/2}\right\}, \tag{A.13}$$

then (4.8) and (4.10) are equivalent. Moreover, if s_* and σ satisfy $\mathcal{B}(s_*)s_* := -g$, (4.8) and (A.13), then s_* is a global minimum.

Proof. If σ satisfies (A.13), then

$$\begin{aligned} \sigma(L+\sigma)^{-2/3} \geq \sigma(2\sigma)^{-2/3} = 2^{-2/3}\sigma^{1/3} \geq 3\lambda\epsilon_g^{-2/3}, \\ \sigma(L+\sigma)^{-1/3} \geq \sigma(2\sigma)^{-1/3} = 2^{-1/3}\sigma^{2/3} \geq 7\Lambda_W\epsilon_g^{-1/3}. \end{aligned}$$

We deduce that

$$\sigma \geq 3\max\left\{\lambda(L+\sigma)^{2/3}\epsilon_g^{-2/3}, \frac{7}{3}\Lambda_W(L+\sigma)^{1/3}\epsilon_g^{-1/3}\right\} \geq 3\max\left\{-\lambda\|s_*\|_W^{-2}, \frac{7}{3}\Lambda_W\|s_*\|_W^{-1}\right\}$$

where the last inequality uses $\|s_*\|_W^{-1} \leq (L+\sigma)^{1/3}\epsilon_g^{-1/3}$ from Lemma A.2.3. Lastly, using Lemma 4.2.3, we obtain the desired result. □

Remark A.2.2. *If $\lambda_{\min}[H] < 0$, then $\lambda := -\lambda_{\min}[H]\lambda_{\min}(W)^{-1} > 0$. For sufficiently small $\epsilon_g > 0$, the dominant term in (A.13) is $\mathcal{O}(\epsilon_g^2)$, leading to $\sigma \geq 4(3\lambda)^3\epsilon_g^{-2}$.*

A.2.3 Convergence and Complexity and DTM

Theorem A.2.4. ***(An upper bound on step size, adapted from Theorem 3.3.1)*** *Suppose that Assumption 4.4.1 holds and Algorithm 4.4.2 is employed. Then, the norm of the iterates (both successful and unsuccessful) is uniformly bounded above independently of i, such that $\|s^{(i)}\|_W < r_c$ for all $i \geq 0$, where r_c is a constant that depends only on the coefficients of m_3.*

Corollary A.2.1. ***(Upper bounding the tensor term, adapted from Section 2.4.2*** *Let $L_H := \Lambda_W + 6\sigma r_c$, where Λ_W defined in Definition 2.3.2. Then, for every iteration, i, we have $\|T_i[s]^2\| < L_H\|s\|_W^2$ and $\|T_i[s]^3\| < L_H\|s\|_W^3$ for all $s \in \mathbb{R}^n$. Note that L_H is an iteration-independent constant that depends only on the coefficients in m_3.*

Lemma A.2.4. ***(A lower bound on the step, adapted from Lemma 3.3.1*** *Suppose that Assumption 4.4.1 holds and Algorithm 4.4.2 is used. Assume that* $\|\nabla m_3(s^{(i)} + s_d^{(i)})\| > \epsilon$ *and* $\|g_i\| > \epsilon$*, then if* [3] $d_i \geq 0$*,*

$$\|s_d^{(i)}\|_W > \min\left\{(B + L_H)^{-1/2}\epsilon^{1/2}, \frac{1}{2}d_i{}^{-1/3}\epsilon^{1/3}\right\}. \tag{A.14}$$

Proof. The proof is the same as the proof for Lemma 3.3.1, with a slight modification: we replace $\left|\beta_i\|s_d\|^2\right| \leq B\|s_d\|^2$ by $\|\mathcal{T}_i[s_d]^2\| \leq B\|s_d\|_W^2$ and change the regularization norm from $\|\cdot\|$ to $\|\cdot\|_W$. □

Theorem A.2.5. ***(A local upper bound for*** m_3***, adapted from Theorem 3.3.2*** *Suppose that Assumption 4.4.1 holds and Algorithm 4.4.2 is used. Then, if*

$$d_i > \frac{B + L_H}{6}\|s_d^{(i)}\|_W^{-1} > 0, \tag{A.15}$$

we have $m_3(s^{(i)} + s_d^{(i)}) = M(s^{(i)}, s_d^{(i)}) \leq M_d(s^{(i)}, s_d^{(i)})$*. with the equality only happens at* $s_d^{(i)} = 0$*.*

Proof. For notational simplicity, in this proof, we use s_d to represent $s_d^{(i)}$. By definition of M_d and M, we have

$$M_d(s^{(i)}, s_d) - M(s^{(i)}, s_d) = \frac{1}{6}\left[[\mathcal{T}_i - T_i][s_d]^3\right] + d_i\|s_d\|_W^4. \tag{A.16}$$

If $s_d = 0$, clearly $M(s^{(i)}, 0) = M_d(s^{(i)}, 0)$. Assume that $\|s_d\|_W > 0$, we choose $d_i > \frac{1}{6}(B+L_H)\|s_d\|_W^{-1} > 0$. Using $\mathcal{T}[s_d]^3 > -B\|s_d\|_W^3$ from Assumption 4.4.1 and Remark 4.4.2, and $T[s_d]^3 \geq -L_H\|s_d\|_W^3$ from Corollary A.2.1, we deduce from (A.16) that $M_d(s^{(i)}, s_d) - M(s^{(i)}, s_d) > \frac{1}{6}(-B - L_H)\|s_d\|_W^3 + d_i\|s_d\|_W^4 \underset{(A.15)}{>} 0$. □

Remark A.2.3. *If* d_i *satisfies (A.15), then* $\sigma_d^{(i)}$ *satisfies* $\sigma_d^{(i)} \geq \frac{2}{3}(-\beta_i+\alpha)\|s_d^{(i)}\|_W^{-1}$ *in Algorithm 4.4.2. To see this, since* $B > -\beta_i$ *, we have* $d_i > \frac{1}{6}(B+L_H)\|s_d^{(i)}\|_W^{-1} > \frac{1}{6}(-\beta_i+\alpha)\|s_d^{(i)}\|_W^{-1}$ *and, consequently,* $\sigma_d^{(i)} = \sigma + 4d_i \geq \frac{2}{3}(-\beta_i + \alpha)\|s_d^{(i)}\|_W^{-1}$*.*

Lemma A.2.5. ***(An upper bound for*** d_i***, adapted from Lemma 3.3.4*** *Suppose that Assumption 4.4.1 holds and Algorithm 4.4.2 is used. Then, for all* $i \geq 0$

$$d_i \leq d_{\max} := \gamma(B + L_H)^{3/2}\epsilon^{-1/2}, \tag{A.17}$$

where γ *is a fixed constant from Algorithm 4.4.2 .*

A.2.3.1 Proof of Theorem 4.4.1

For notational simplicity, in this proof, we use s_d to represent $s_d^{(i)}$ and σ_d to represent $\sigma_d^{(i)}$. Since i is a successful iteration, (Ratio Test) gives

$$\mathcal{D} := \eta^{-1}\left[m_3(s^{(i)}) - m_3(s^{(i+1)})\right] \geq M_d(s^{(i)}, 0) - M_d(s^{(i)}, s_d) = -g_i^T s_d - \frac{1}{2}H_i[s_d]^2 - \frac{1}{6}\mathcal{T}_i[s_d]^3 - \frac{\sigma_d}{4}\|s_d\|_W^4.$$

s_d is the global minimum of M_d and satisfies $-g_i = \mathcal{B}_i(s_d)s_d$. We have

$$M_d(s^{(i)}, 0) - M_d(s^{(i)}, s_d) = \frac{1}{2}H_i[s_d]^2 + \frac{1}{3}\mathcal{T}_i[s_d]^3 + \frac{3\sigma_d}{4}\|s_d\|_W^4. \tag{A.18}$$

Using Lemma 4.2.2 with $-g_i = \mathcal{B}_i(s_d)s_d$, we have

$$M_d(s^{(i)}, s_d + v) - M_d(s^{(i)}, s_d) = \frac{1}{2}\left[H_i + \mathcal{T}_i[s] + \sigma_d W\|s\|_W^2\right][v]^2 + \frac{1}{6}T[v]^3 + \frac{\sigma}{4}\left[\|s+v\|_W^2 - \|s\|_W^2\right]^2.$$

Let $v = -2s_d$,

$$\frac{1}{2}H_i[s_d]^2 + \frac{1}{6}\mathcal{T}_i[s_d]^3 + \frac{\sigma_d}{2}\|s_d\|_W \geq 0. \tag{A.19}$$

[3] If $d_i = 0$, $\|s_d^{(i)}\|_W > (B + L_H)^{-1/2}\epsilon^{1/2}$.

- Assume that $\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 \geq \alpha > 0$, substituting (A.19) into (A.18), we have $\mathcal{D} \geq \frac{1}{6}\mathcal{T}_i[s_d]^3 + \frac{\sigma_d}{4}\|s_d\|_W^4 \geq \frac{\alpha}{3}\|s_d\|_W^3$.
- Assume that $\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 \leq \alpha$ and $\sigma_d \geq \frac{2}{3}\bigg(-\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 + \alpha\bigg)\|s_d\|_W^{-1} > 0$,

$$\mathcal{D} \geq \frac{1}{6}\mathcal{T}_i[s_d]^3 + \frac{1}{6}\bigg(-\mathcal{T}_i\bigg[\frac{s_d}{\|s_d\|_W}\bigg]^3 + \alpha\bigg)\|s_d\|_W^3 \geq \frac{\alpha}{6}\|s_d\|_W^3.$$

- Assume that $\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 \in [-B, -4\alpha]$ and $0 \leq \sigma_d \leq \frac{1}{6}\bigg(-\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 + \alpha\bigg)\|s_d\|_W^{-1}$, according to the second-order local optimality condition, $0 \leq \nabla^2 M_d(s^{(i)}, s_d)[s_d]^2 = H_i[s_d]^2 + \mathcal{T}_i[s_d]^3 + 3\sigma_d\|s_d\|_W^4$. Substituting the local optimality condition into (A.18), we have

$$\begin{aligned}\mathcal{D} \geq -\frac{1}{6}\mathcal{T}_i[s_d]^3 - \frac{3\sigma_d}{4}\|s_d\|_W^4 &\geq -\frac{1}{6}\mathcal{T}_i[s_d]^3 - \frac{1}{8}\bigg(-\mathcal{T}_i\bigg[\frac{s_d}{\|s_d\|_W}\bigg]^3 + \alpha\bigg)\|s_d\|_W^3 \\ &\geq -\frac{1}{24}\mathcal{T}_i[s_d]^3 - \frac{\alpha}{8}\|s_d\|_W^3 \geq \frac{\alpha}{24}\|s_d\|_W^3\end{aligned}$$

  where the last inequality uses $\mathcal{T}_i\big[\frac{s_d}{\|s_d\|_W}\big]^3 \leq -4\alpha$.

In all cases, we have $m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\|s_d\|_W^3$. These three cases correspond to the requirements of β_i and σ_d in Algorithm 4.4.2. Using the lower bound on $\|s_d\|_W$ in Lemma A.2.4 and the bound on d_i in Lemma A.2.5,

$$m_3(s^{(i)}) - m_3(s^{(i+1)}) \geq \frac{\alpha\eta}{24}\max\bigg\{(B+L_H)^{-3/2}\epsilon^{3/2}, \frac{1}{8}{d_{\max}}^{-1}\epsilon\bigg\} \geq \frac{\alpha\eta}{24}\min\bigg\{1, \frac{1}{8\gamma}\bigg\}(B+L_H)^{-3/2}\epsilon^{3/2}.$$

Using $\gamma > 1$, gives the desired result in (4.49).

Remark A.2.4. *Note that the $\sigma_d^{(i)}$ condition in Algorithm 4.4.2 is meant to ensure that the decrease in values of m_3 is at least $\mathcal{O}(\epsilon^{3/2})$ as indicated in (4.49). If we only have the condition $\rho_i \geq \eta$, Algorithm 4.4.1 still results in a decrease in the value of m_3 at each successful iteration. We can verify the decrease in value of m_3 using a standard Cauchy analysis argument as illustrated in Remark 3.3.4.*

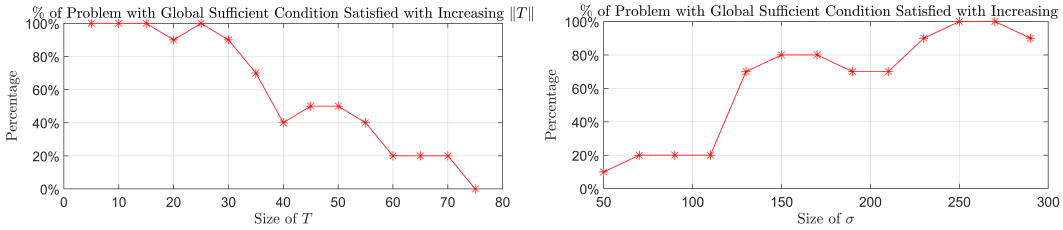
A.2.3.2 Proof of Theorem 4.4.2

Proof. The complexity proof follows the same approach as in Lemma 3.3.3 and Lemma 3.3.4. We begin by bounding the number of successful iterations, denoted as $|\mathcal{S}_i| := \kappa_s\frac{m_3(s^{(0)}) - m_{\text{low}}}{\epsilon^{3/2}}$ and then proceed to bound the total number of iterations, given by $i \leq |\mathcal{S}_i| + \frac{\log(d_{\max})}{\log\gamma}$. Substituting $|\mathcal{S}_i|$ and applying the bound on $d_{\max}$ from (A.17) yields the desired result with $\kappa_c := 1 + \frac{3\log(B+L_H)}{2\log\gamma}$. A detailed proof can be found in Theorem 3.3.4. □

A.2.4 Supplement for Numerical Results

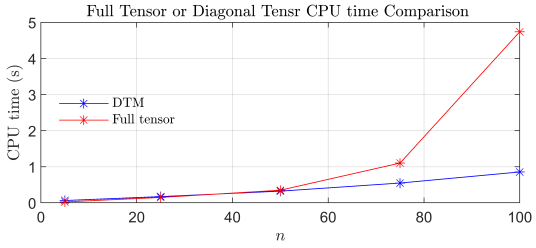
A.2.4.1 Size of Tensor and Global Optimality Characterization

Figure A.6: Global optimality condition and size of $\|T\|$ or σ using the Diagonal Tensor Test Set: **First plot**: `a = 10, b = 20`, $\sigma = 100$, `c` $\in [5, 80]$. **Second plot**: `a = 10, b = 20, c= 50`, $\sigma \in [50, 300]$.



A.2.4.2 Computational Time Comparison for Applying Algorithm 4.4.1 Directly to AR3 Subproblem

Figure A.7: Computational time comparison: DISPL Algorithm 4.4.1 with $\mathcal{T}_i = T_i$, where T_i is the full third-order derivative, versus with $\mathcal{T}_i$ chosen as the diagonal tensor using the Diagonal Rule in Algorithm 4.5.1. The test functions and experimental setup are identical to those in Table 4.1.



A.3 Appendix for Chapter 5

A.3.1 Supplement for the Main Proof

A.3.1.1 Alternative formulations for $m_3(s) - m_3(s^*)$

In this section, we present alternative formulations of m_3 expanded at its global minimizer s^*. We highlight the connection between existing results in polynomial global optimization [93] and the global optimality conditions for m_3 developed in Chapter 4.

Lemma A.3.1 (Adapted from Lemma 8 in [93]). *Assume* $\nabla m_3(s^*) = 0$,

$$m_3(s^* + v) - m_3(s^*) = v^T \Bigg[\underbrace{\int_0^1 \int_0^t \nabla^2 m_3(s^* + rv)\, dr dt}_{:= \mathcal{I}(s^*, v)} \Bigg] v. \tag{A.20}$$

Lemma A.3.2. *A point* $s^* \in \mathbb{R}^n$ *is a global minimizer of* m_3 *if and only if*

$$\nabla m_3(s^*) = 0 \qquad \text{and} \qquad \mathcal{I}(s^*, v) \succeq 0 \quad \text{for all } v \in \mathbb{R}^n.$$

Proof. of Lemma A.3.2 If s^* is the global minimum, then by the first-order condition we have $\nabla m_3(s^*) = 0$. By Lemma A.3.1, we have $m_3(s^* + v) - m_3(s^*) = \mathcal{I}(s^*, v)[v]^2$, since s^* is the global minimum, $\mathcal{I}(s^*, v)[v]^2$ is positive for all $v \in \mathbb{R}^n$. On the other hand, if $\nabla m_3(s^*) = 0$, by Lemma A.3.1, we have $m_3(s^* + v) - m_3(s^*) \geq 0$ for all $v \in \mathbb{R}^n$, then s^* is the global minimum. □

Lemma A.3.1 is equivalent to Lemma 5.3.1. We can see this using the following deductions. Using $\nabla^2 m_3(s) = H + T[s] + \sigma(\|s\|^2 + 2ss^T)$, from (A.20), we have

$$\begin{aligned}
\mathcal{I}(s^*, v) &= \int_0^1 \int_0^t \left[H + T[s^* + rv] + \sigma\left(\|s^* + rv\|^2 I_n + 2[s^* + rv][s^* + rv]^T \right) \right] dr dt \\
&= \int_0^1 \int_0^t \left[\nabla^2 m_3(s^*) + r\left(T[v] + 2\sigma s^{*T} v I_n + 4\sigma s^* v^T \right) + r^2 \sigma(\|v\|^2 I_n + 2vv^T) \right] dr dt \\
&= \frac{1}{2}\nabla^2 m_3(s^*) + \frac{1}{6} T[v] + \frac{\sigma}{6}\left(2s^{*T} v I_n + 4s^* v^T \right) + \frac{\sigma}{12}(\|v\|^2 I_n + 2vv^T). \qquad \text{(A.21)}
\end{aligned}$$

Multiplying $\mathcal{I}(s^*, v)$ with v^T and v ,

$$\mathcal{I}(s^*, v)[v]^2 = \frac{1}{2}\nabla^2 m_3(s^*)[v]^2 + \frac{1}{6} T[v]^3 + \sigma s^{*T} v \|v\|^2 + \frac{\sigma}{4}\|v\|^4$$

which is the same result in (5.5).

A.3.1.2 Comparison of Tensor Norm

For a symmetric third-order tensor $T \in \mathbb{R}^{n \times n \times n}$, define its cubic (spectral) norm as

$$\|T\| := \max_{\|s\|=1} |T[s]^3| = \max_{\|s\|=1} |\langle T,\, s \otimes s \otimes s \rangle|.$$

By the Cauchy–Schwarz inequality for tensor inner products, we have

$$|T[s]^3| = |\langle T,\, s \otimes s \otimes s \rangle| \le \|T\|_F \, \|s \otimes s \otimes s\|_F = \|T\|_F \, \|s\|^3 = \|T\|_F.$$

Hence,

$$\boxed{\|T\| \le \|T\|_F.}$$

Moreover, using the mode-1 unfolding $T_{(1)} \in \mathbb{R}^{n \times n^2}$, one obtains

$$T[s]^3 = s^T T_{(1)}(s \otimes s) \quad \Rightarrow \quad |T[s]^3| \le \|T_{(1)}\|_2,$$

and therefore

$$\|T\| \le \min\{\|T_{(1)}\|_2,\, \|T_{(2)}\|_2,\, \|T_{(3)}\|_2\} \le \|T\|_F.$$

Equality $\|T\| = \|T\|_F$ holds if and only if T is (up to a sign) a rank-one symmetric tensor of the form $T = \alpha\, u \otimes u \otimes u$ with $\|u\| = 1$. A simple lower bound follows from the diagonal entries:

$$\max_{1 \le i \le n} |T_{iii}| \le \|T\|.$$

Thus, $\|T\|$ is always upper bounded by the Frobenius norm, attains equality for rank-one symmetric tensors, and provides a tighter measure of the dominant cubic interaction captured by T.

A.3.1.3 s^* a non-increasing function of σ

Theorem A.3.1. *Suppose that $s^* = s^*(\sigma) \neq 0$ is any second-order local minimizer of the model $m_3(s)$. Then the length of the minimizer $v(\sigma) = \|s^*(\sigma)\|$ is a non-increasing function of σ.*

Proof. Since $s^*(\sigma) \neq 0$ a second-order local minimizer of the model $m_3(s)$, then

$$\nabla m_3\big(s^*(\sigma)\big) = 0, \qquad H s^*(\sigma) + \frac{1}{2} T[s^*(\sigma)]^2 + \sigma \|s^*(\sigma)\|^2 s^*(\sigma) = -g. \qquad \text{(A.22)}$$

Differentiating (A.22) with respect to σ gives

$$H\nabla_\sigma s^*(\sigma) + T[s^*(\sigma)]\nabla_\sigma s^*(\sigma) + \sigma\|s^*(\sigma)\|\nabla_\sigma s^*(\sigma) + 2s^*(\sigma)^T s^*(\sigma)\nabla_\sigma s^*(\sigma) + \|s^*(\sigma)\|^2 s^*(\sigma) = 0 \qquad \text{(A.23)}$$

and therefore

$$\nabla^2 m_3(s^*(\sigma))\nabla_\sigma s^*(\sigma) + \|s^*(\sigma)\|^2 s^*(\sigma) = 0. \tag{A.24}$$

Let $v(\sigma) = \|s^*(\sigma)\|$, then $v'(\sigma) = \|s^*(\sigma)\|^{-1}\nabla_\sigma s^*(\sigma)^T s^*(\sigma)$. Premultiplying (A.24) by $\nabla_\sigma s^T(\sigma)$ and dividing by $\|s^*(\sigma)\|$ gives

$$v'(\sigma) = \|s^*(\sigma)\|^{-1}\nabla_\sigma s^*(\sigma)^T s^*(\sigma) = \frac{-\nabla_\sigma s^*(\sigma)^T \nabla^2 m_3(s^*(\sigma))\nabla_\sigma s^*(\sigma)}{\|s^*(\sigma)\|^2} \leq 0.$$

Since we have seen that $\nabla^2 m_3(s^*(\sigma))$ is positive semidefinite. Thus, $s^*(\sigma)$ is a non-increasing function of σ. □

Remark A.3.1. *Theorem A.3.1 can be extended to pth order for any second-order local minimizer of* $m_p(s)$.

A.3.1.4 Using Cauchy analysis to give upper bound for m_*

Theorem A.3.2. ***(Upper bound for*** m_****)*** *Assume* σ *satisfies*

$$\sigma \geq \max\left\{4{\lambda^*}^3\|g\|^{-2}, \frac{1}{2}\Lambda_T^{3/2}\|g\|^{-1/2}\right\}, \tag{A.25}$$

then $m_3(s^*) \leq -\frac{3}{8}2^{-1/3}\|g\|^{4/3}\sigma^{-1/3}$.

Proof. Let

$$0 \leq k \leq \min\left\{\frac{1}{2\lambda^*}, \sqrt{\frac{3}{2\Lambda_T\|g\|}}\right\}. \tag{A.26}$$

We use a Cauchy analysis to give the upper bound.

$$\begin{aligned}
m_3(s^*) &\leq m(-kg) = -k\|g\|^2 + \frac{k^2}{2}H[g]^2 + \frac{k^3}{6}T[g]^3 + \frac{k^4\sigma}{4}\|g\|^4 \\
&\leq -k\|g\|^2 + \frac{k^2}{2}\|g\|^2\lambda^* + \frac{k^3}{6}\|g\|^3\Lambda_T + \frac{k^4\sigma}{4}\|g\|^4 \\
&\leq -\frac{k}{2}\|g\|^2 + \underbrace{\left[-\frac{k}{4} + \frac{k^2}{2}\lambda^*\right]}_{\leq 0 \text{ by (A.26)}}\|g\|^2 + \underbrace{\left[-\frac{k}{4} + \frac{k^3}{6}\Lambda_T\|g\|\right]}_{\leq 0 \text{ by (A.26)}}\|g\|^2 + \frac{k^4\sigma}{4}\|g\|^4 \\
&\leq -\frac{k}{2}\|g\|^2 + \frac{k^4\sigma}{4}\|g\|^4 \leq -\frac{3}{8}2^{-1/3}\|g\|^{4/3}\sigma^{-1/3}
\end{aligned}$$

where the last inequality happens by taking $k_* = 2^{-1/3}\sigma^{-1/3}\|g\|^{-2/3}$. k_* is well-defined, since using (A.25), we have $k_* \in \left[0, \min\left\{\frac{1}{2\lambda^*}, \sqrt{\frac{3}{2\Lambda_T\|g\|}}\right\}\right]$ which satisfies (A.26). □

A.3.1.5 Global Optimality Conditions for m_3

For a general m_3, global optimality conditions were established in Chapter 4. Let $s^* \in \mathbb{R}^n$ satisfy $\nabla m_3(s^*) = 0$. If s^* is a global minimum, then

$$B_N(s^*) := H + \frac{2}{3}T[s^*] + \sigma\|s^*\|^2 I_n + \frac{\|T\|}{3}\|s^*\|I_n \succeq 0. \qquad \text{(Necessary cond.)}$$

If the following is satisfied,

$$B_S(s^*) := H + \frac{2}{3}T[s^*] + \sigma\|s^*\|^2 I_n - \frac{\|T\|}{3}\|s^*\|I_n - \frac{\|T\|^2}{18\sigma}I_n \succeq 0, \qquad \text{(Sufficient cond.)}$$

then s^* is a global minimum.

Theorem A.3.3. *Assume that m_3 is locally nonconvex at $s = 0$, namely $\nabla m_3(0) = H$ is not positive definite. Let $s^* \in \mathbb{R}^n$ satisfy $\nabla m_3(s^*) = 0$. If σ is chosen such that*

$$\sigma \geq 9 \max\left\{ \frac{27\lambda_*^3}{\|g\|^2}, \quad \frac{\|H\|^3}{\|g\|^2}, \quad \left(\frac{\|T\|^3}{\|g\|}\right)^{1/2} \right\}, \tag{A.27}$$

where $\lambda_* := \min\{-\lambda_{\min}(H), 0\} \geq 0$, $\|H\|$, $\|T\|$ are defined in (5.15)–(5.16). Then, the following statements hold:

1. $\left(\frac{\|g\|}{3\sigma}\right)^{1/3} \leq \|s^*\| \leq 2\left(\frac{\|g\|}{\sigma}\right)^{1/3}$ *and* $\sigma \geq 3 \max\left\{ \lambda_* \|s^*\|^{-2}, \|T\| \, \|s^*\|^{-1} \right\}$,
2. $B_S(s^*) \succeq 0$, *s^* is the global minimizer of m_3, the necessary condition* (Necessary cond.) *and sufficient condition* (Sufficient cond.) *coincide.*

Proof. **To prove result 1:** If σ satisfies (A.27), then σ satisfies the condition in Theorem 5.3.2, and the bound on $\|s^*\|$ in the first result follows directly from Theorem 5.3.2. Moreover, from (A.27), we have

$$\sigma^{1/3} \geq 3^{5/3} \|g\|^{-2/3} \lambda_*, \qquad \sigma^{2/3} \geq 3^{4/3} \|g\|^{-1/3} \|T\|.$$

We deduce that

$$\sigma \geq 3 \max\left\{ 3^{2/3} \sigma^{2/3} \|g\|^{-2/3} \lambda_*, \quad 3^{1/3} \sigma^{1/3} \|g\|^{-1/3} \|T\| \right\} \geq 3 \max\left\{ \lambda_* \|s^*\|^{-2}, \quad \|T\| \|s^*\|^{-1} \right\} \tag{A.28}$$

where the last inequality uses $\|s^*\| \geq \left(\frac{\|g\|}{3\sigma}\right)^{1/3}$ (i.e., $3^{1/3}\sigma^{1/3}\|g\|^{-1/3} \geq \|s^*\|^{-1}$) from Theorem 5.3.2.
To prove result 2: Given (A.28), we have

$$\begin{aligned} \sigma > 6^{-1/2} \|T\| \|s^*\|^{-1} \quad &\Rightarrow \quad \frac{\sigma}{3} \|s^*\|^2 > \frac{\|T\|^2}{18\sigma}, \\ \sigma \geq 3 \|T\| \|s^*\|^{-1} \quad &\Rightarrow \quad \frac{\sigma}{3} \|s^*\|^2 \geq \|T\| \|s^*\|, \\ \sigma \geq 3\lambda_* \|s^*\|^{-2} \quad &\Rightarrow \quad \frac{\sigma}{3} \|s^*\|^2 \geq \lambda_*. \end{aligned}$$

We deduce that

$$B_S(s^*) \succeq \left(\lambda_* - \|T\| \|s^*\| + \sigma \|s^*\| - \frac{\|T\|^2}{18\sigma} \right) I_n \succeq 0.$$

Since $\nabla m_3(s^*) = 0$ and $B_S(s^*) \succeq 0$, it follows that s^* is the global minimizer of m_3. Furthermore, as s^* is a global minimizer, the necessary condition $B_N(s^*) \succeq 0$ also holds. By construction, we already have $B_N(s^*) \succeq B_S(s^*) \succeq 0$, and thus the necessary condition (Necessary cond.) and the sufficient condition (Sufficient cond.) coincide.

□

A.3.2 Supplement for Classes of SoS Quartic Regularized Polynomial

A.3.2.1 Proof for Theorem 5.4.3

Proof. Assume $T = 0$, since s^* is the global minimum, we have $\nabla m_3(s^*) = 0$, according to (5.5), we can write

$$m_3(s^* + v) - m_3(s^*) = \frac{1}{2}\phi_2(v) \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \nabla^2 m_3(s^*) & \sigma s \mathbf{1}^T & 0 \\ 0 & \sigma \mathbf{1} s^T & \frac{\sigma}{2} \mathbf{1}\mathbf{1}^T & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{Q_b} \phi_2(v) \tag{A.29}$$

where ϕ_2 defined as in (5.2), $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^n$ and $\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{n\times n}$ is a matrix with all entries equal to 1s. Let $\omega \in \mathbb{R}^{\frac{1}{2}(n+2)(n+1)}$ be any vector, we write $\omega = [1, v, u, z]^T$ as defined in (5.3). For any ω, or equivalently for any (u, v, z), we have

$$\begin{aligned} 2\omega^T Q_b(v)\omega &= \nabla^2 m_3(s^*)[v]^2 + 2\sigma v^T s^* \sum_{i=1}^n u_i + \frac{\sigma}{2}\sum_{i=1}^n u_i^2 + \sigma \sum_{1\le i<j\le n} u_i u_j \\ &= \nabla^2 m_3(s^*)[v]^2 + \frac{\sigma}{2}\Big(\sum_{i=1}^n u_i + 2v^T s^*\Big)^2 - 2\sigma(v^T s^*)^2 \\ &= \big[\underbrace{\nabla^2 m_3(s^*) - 2\sigma s^* s^{*T}}_{\succeq 0}\big][v]^2 + \frac{\sigma}{2}\Big(\sum_{i=1}^n u_i + 2v^T s^*\Big)^2 \ge 0 \end{aligned} \tag{A.30}$$

where the last inequality uses the second-order global optimality condition for s^* given in Lemma 5.4.3 (i.e., $H + \sigma\|s^*\| I_n \succeq 0$). Thus, we obtained that $Q_b \succeq 0$ and therefore $m_3(s^* + v) - m_3(s^*)$ is SoS. □

Remark A.3.2. *The bound in (A.30) is tight, as we can set $v = -s^*$ and $\sum u_i = \|s^*\|^2$. When $n = 1$, Theorem 5.4.3 becomes the result in Remark 5.4.1.*

A.3.2.2 Nonnegative Bivariate Quartic Polynomials are SoS

Theorem A.3.4 (Hilbert, 1888)**.** *Every nonnegative bivariate quartic polynomial is a sum of squares of quadratic polynomials.*

Outline of proof. Let $p(x, y)$ be a homogeneous quartic with $p(x, y) \ge 0$ for all $(x, y) \in \mathbb{R}^2$. By homogeneity, write $p(x, y) = y^4 f(x/y)$ for a univariate quartic $f(t)$. Nonnegativity of p implies $f(t) \ge 0$ for all $t \in \mathbb{R}$. It is well known that any nonnegative univariate quartic can be expressed as a sum of two squares of quadratic polynomials: $f(t) = r(t)^2 + s(t)^2$. Multiplying by $(x^2 + y^2)^2$ and homogenizing yields

$$p(x, y) = (x^2 + y^2)^2 f(x/y) = q_1(x, y)^2 + q_2(x, y)^2,$$

where q_1, q_2 are quadratic forms. Thus p is SoS of quadratics. □

Theorem A.3.5. *Every nonnegative quartic polynomial $p(x, y)$ in two variables is SoS of quadratic polynomials; i.e., there exist quadratic polynomials $q_k(x, y)$ such that $p(x, y) = \sum_{k=1}^r q_k(x, y)^2$.*

Proof. **Step 1: Translate to a minimizer and normalize.** Since $p \ge 0$ and $\deg p = 4$, its degree-4 homogeneous part p_4 is nonnegative (otherwise p would be negative and unbounded along a ray). Choose a global minimizer (x_0, y_0) of p (if the minimum is not attained, argue with a limiting sequence and the same local analysis below; the identity obtained is polynomial and hence global). Define $u = x - x_0$, $v = y - y_0$ and write $p(x, y) = c + q_2(u, v) + q_3(u, v) + q_4(u, v)$, where $c = p(x_0, y_0) \ge 0$, q_k is homogeneous of degree k, and $\nabla p(x_0, y_0) = 0$ implies $q_1 \equiv 0$.

Step 2: Handle the quadratic part by completing squares. Let $H = \nabla^2 p(x_0, y_0)$ be the Hessian at the minimizer; then $H \succeq 0$. Hence $q_2(u, v) = \ell_1(u, v)^2 + \ell_2(u, v)^2$ for some real linear forms ℓ_1, ℓ_2. Thus $c + q_2(u, v) = \sum_{i=0}^2 \alpha_i \ell_i(u, v)^2$ with $\ell_0 \equiv 1$ (absorbing $c \ge 0$ as α_0).

Step 3: Eliminate the cubic part via quadratic squares. Let l_1, l_2 be linear forms and s_1, s_2 quadratic homogeneous forms to be chosen. Consider $(\ell_1 + s_1)^2 + (\ell_2 + s_2)^2 = \ell_1^2 + \ell_2^2 + 2(\ell_1 s_1 + \ell_2 s_2) + s_1^2 + s_2^2$. The cubic terms arise solely from $2(\ell_1 s_1 + \ell_2 s_2)$, since ℓ_i is degree 1 and s_i is degree 2. Because q_3 is an arbitrary cubic homogeneous polynomial in two variables, we can choose s_1, s_2 (which span the space of homogeneous quadratics) so that $2(\ell_1 s_1 + \ell_2 s_2) = q_3$. This is a surjective linear map from the 4-dimensional space of quadratic pairs $(s_1 = a_1 u^2 + a_2 v^2, s_2 = a_3 u^2 + a_4 v^2)$ onto the 4-dimensional space of homogeneous cubic forms $(q_3 = t_1 v^3 + t_2 v^2 u + t_3 v u^2 + t_4 u^3)$, hence solvable.[4] With these s_1, s_2, we have $c + q_2 + q_3 = \alpha_0 + (\ell_1 + s_1)^2 + (\ell_2 + s_2)^2 - (s_1^2 + s_2^2)$.

[4]The linear map $(s_1, s_2) \longmapsto \ell_1 s_1 + \ell_2 s_2 = u(a_1 u^2 + a_2 v^2) + v(a_3 u^2 + a_4 v^2) = a_1 u^3 + a_3 u^2 v + a_2 u v^2 + a_4 v^3$ is therefore surjective onto the 4-dimensional space of homogeneous cubic binary forms $q_3(u, v) = t_1 v^3 + t_2 u v^2 + t_3 u^2 v + t_4 u^3$.

Step 4: The remaining homogeneous quartic is nonnegative and hence SoS. Collecting terms,

$$p(u,v) = \left[\alpha_0 \cdot 1^2 + (\ell_1 + s_1)^2 + (\ell_2 + s_2)^2\right] + \underbrace{\left(q_4(u,v) - s_1(u,v)^2 - s_2(u,v)^2\right)}_{=:\, R_4(u,v)}.$$

By construction, the bracketed term is SoS of polynomials of degree ≤ 2. We claim the residual R_4 is a nonnegative *homogeneous* quartic in (u,v). Indeed, near the minimizer,

$$p(u,v) - \left[\alpha_0 \cdot 1^2 + (\ell_1 + s_1)^2 + (\ell_2 + s_2)^2\right] = R_4(u,v) \geq 0 \quad \text{to order 4},$$

because all lower-degree terms (degree ≤ 3) have been matched exactly and $p \geq 0$ globally; hence the quartic homogeneous term R_4 must be nonnegative[5]. By Hilbert's 1888 binary form theorem (Theorem A.3.4), any nonnegative homogeneous quartic in two variables is SoS of homogeneous quadratics. Therefore, $R_4(u,v) = q_3(u,v)^2 + q_4(u,v)^2$, for some homogeneous quadratic forms q_3, q_4.

Step 5: Conclusion. Combining Steps 2–4, $p(x,y) = \alpha_0 \cdot 1^2 + (\ell_1 + s_1)^2 + (\ell_2 + s_2)^2 + q_3^2 + q_4^2$, SoS of degree at most 2 in (x,y). □

A.3.3 Supplementary Results for Perturbing $m_{\mathcal{E}}$

Table A.6: Comparison of Global Minima Found by MoM and γ^* Found by SoS

Objective	Global Mins Found by MoM	γ^* Found by SoS
$m_{\mathcal{E}}$	-	$1 - 3.6655 \times 10^{-10}$
$\hat{m}_{\mathcal{E},1}$	$\pm[0.7071, -0.7071, -0.0000]^T$	$1 - 4.0000 \times 10^{-5}$
$\hat{m}_{\mathcal{E},2}$ - run 1	$\pm[0.7809, -0.1830, -0.5979]^T$	$1 - 6.6080 \times 10^{-6}$
$\hat{m}_{\mathcal{E},2}$ - run 2	$\pm[0.2089, 0.5790, -0.7879]^T$	$1 - 2.1461 \times 10^{-5}$
$\hat{m}_{\mathcal{E},2}$ - run 3	$\pm[0.7107, -0.0072, -0.7035]^T$	$1 - 4.3092 \times 10^{-5}$
$\hat{m}_{\mathcal{E},2}$ - run 4	$\pm[0.8096, -0.3135, -0.4962]^T$	$1 - 5.6846 \times 10^{-5}$
$\hat{m}_{\mathcal{E},2}$ - run 5	$\pm[0.7903, -0.2176, -0.5728]^T$	$1 - 2.8288 \times 10^{-5}$

A.4 Appendix for Chapter 6

A.4.1 Proof of Lemma 6.3.1

Proof. **Result 1)** Let $q(s) := \frac{1}{j!}\nabla_x^j f(x_k)[s]^j$ be the symmetric degree j polynomial in n variables,

$$q(s) = \frac{1}{j!}\nabla_x^j f(x_k)[s]^j = \frac{1}{j!}\sum_{i_1,\ldots,i_j=1}^{n} \left\{\nabla_x^j f(x_k)\right\}_{[i_1,\ldots,i_j]} s_{[i_1]} \cdots s_{[i_j]}$$

where $1 \leq i_1, \ldots, i_j \leq n$, $\left\{\nabla_x^j f(x_k)\right\}_{[i_1,\ldots,i_j]}$ denotes the $\{i_1,\ldots,i_j\}$th entry of the jth order tensor $\nabla_x^j f(x_k)$ and $s_{[l]}$ denotes the lth entry of the vector $s \in \mathbb{R}^n$ for $1 \leq l \leq n$. Then the largest absolute value of the coefficients of $q(s)$ in a monomial basis can be expressed as

$$\|q(s)\|_{\infty^*} = \max_{1\leq i_1,\ldots,i_j\leq n} \left\{\frac{1}{(j+1-\varsigma_j)!}\left\{\left|\nabla_x^j f(x_k)\right|\right\}_{[i_1,\ldots,i_j]}\right\} \leq \max_{1\leq i_1,\ldots,i_j\leq n} \left\{\left|\nabla_x^j f(x_k)\right|\right\}_{[i_1,\ldots,i_j]}.$$

where $1 \leq \varsigma_j \leq j$ denotes the number of distinct elements[6] in $\{i_1,\ldots,i_j\}$. We proved that $\|q(s)\|_{\infty^*}$ is smaller than the largest absolute value of the entry of $\nabla_x^j f(x_k)$. Assume without loss of generality that the largest absolute value of the entry of $\nabla_x^j f(x_k)$ is attained at the $\{i_1^*,\ldots,i_j^*\}$th entry, then

$$\|q(s)\|_{\infty^*} \leq \left\{\left|\nabla_x^j f(x_k)\right|\right\}_{[i_1^*,\ldots,i_j^*]} = \left|\nabla_x^j f(x_k)[e_{i_1^*},\ldots,e_{i_j^*}]\right| \leq \left\|\nabla_x^j f(x_k)\right\|_{[j]} \underset{\text{Assumption 6.3.2}}{\leq} \Lambda_j$$

[5]More formally, consider $t \mapsto R_4(tu,tv) = t^4 R_4(u,v)$ as the order-4 Taylor jet of a nonnegative function at 0; the coefficient must be ≥ 0 for all directions (u,v).

[6]Due to the symmetry of the tensor, we collect the coefficients corresponding to the same monomial basis element.

where $e_{i_j^*} \in \mathbb{R}^n$ represents the unit vector with all entries 0 except for the i_j^*th entry, which is 1.
Result 2) For any $k \geq 0$, under Assumption 6.3.2, we have $\left|\lambda_{\min}[\nabla_x^2 f(x_k)]\right| \leq \max_{\|v\|=1} |\nabla_x^2 f(x_k)[v]^2| \leq \max_{\|v_1\|=\|v_2\|=1} |\nabla_x^2 f(x_k)[v_1, v_2]| = \left\|\nabla_x^2 f(x_k)\right\|_{[2]} \leq \Lambda_2$. □

A.4.2 A Discussion for $a > \frac{1}{2}$

For locally convex iterations, the function value reduction in (6.32) and (6.33) deteriorates as the value of a increases. Opting for $a > \frac{1}{2}$ is unfavorable in this scenario.
For the locally nonconvex case, if $\frac{1}{2} < a < \frac{p'-2}{p'-3}$, then the same function value reduction bound in (6.34) remains. If $a > \frac{p'-2}{p'-3} > \frac{1}{2}$, then $\frac{a(p-2)+1}{p} \geq 1 \geq \frac{1}{p}$, $\max\{a(p-1), 1\} = a(p-1)$ and $\frac{a(p-1)+1}{p+1} \geq 1 \geq \frac{1}{p}$, therefore (6.25) becomes

$$\|s_k\| > c_3 \epsilon_g^{\frac{a(p'-3)+1}{p'-1}}$$

where $c_3 = \frac{1}{3}\min\{\hat{C}_R^{-\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}\}$. Consequently, (6.41) becomes

$$\eta^{-1} E_k > \frac{1}{2}\big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\|^2 \geq \delta \epsilon_g^{\frac{2a(p'-3)+2}{p'-1}} = \frac{c_3^2}{2}\epsilon_g^{a+\frac{2a(p'-3)+2}{p'-1}}.$$

Note that $a + \frac{2a(p'-3)+2}{p'-1} > 2 + a > 2$ which is a worsen bound compared to (6.34).
For locally nearly strongly convex cases: For odd $p \geq 3$ and $a > \frac{1}{2}$, then $1 - a \leq \frac{a(p-2)+1}{p}$ and $\frac{1}{p} \leq \frac{a(p-2)+1}{p}$. For even $p \geq 4$ and $a > \frac{1}{2}$, then $\max\{a(p-1), 1\} = a(p-1)$, $1 - a \leq \frac{a(p-1)+1}{p+1}$ and $\frac{1}{p} \leq \frac{a(p-1)+1}{p+1}$. In both cases, the third term in (6.31) and (6.33) are dominant. Consequently, $\|s_k\| > \frac{1}{3}\hat{C}_R^{-\frac{1}{p}} \epsilon_g^{\frac{a(p-2)+1}{p}}$ for odd p and $\|s_k\| > \frac{1}{3}\tilde{C}_R^{-\frac{1}{p+1}} \epsilon_g^{\frac{a(p-1)+1}{p+1}}$ for even p. Therefore (6.26) becomes

$$\|s_k\| > c_3 \epsilon_g^{\frac{a(p'-3)+1}{p'-1}}$$

where $c_3 = \frac{1}{3}\min\{\hat{C}_R^{-\frac{1}{p}}, \tilde{C}_R^{-\frac{1}{p+1}}\}$. Consequently, (6.42) becomes

$$\eta^{-1} E_k > \delta\frac{1}{2}\big(-\lambda_{\min}[H_k] + \delta\big)\|s_k\|^2 \geq \delta \epsilon_g^{\frac{2a(p'-3)+2}{p'-1}} = \frac{c_3^2}{2}\epsilon_g^{a+\frac{2a(p'-3)+2}{p'-1}}.$$

Note that $\frac{2a(p'-3)+2}{p'-1} + a \geq \frac{3}{2}$ for $a > \frac{1}{2}$. The minimum of $\frac{2a(p'-3)+2}{p'-1} + a$ is attained at $a = \frac{1}{2}$ and decreases as a increases.

A.4.3 Supplement for Numerical Results

A.4.3.1 Derivative and Tensor for Sensor–Localization model

Let S be the set of sensor indices, A the set of anchors, and let each sensor/anchor position lie in $\mathbb{R}^3$. For $(i,j) \in S \times S$ and $(i,j) \in S \times A$, define the residuals

$$r_{ij}(x) \;=\; \|x_i - x_j\|^2 - d_{ij}^2, \qquad \tilde{r}_{ij}(x) \;=\; \|x_i - s_j\|^2 - d_{ij}^2,$$

so that the objective can be written as the quartic polynomial

$$f(x) = \sum_{i,j \in S} r_{ij}(x)^2 + \sum_{i \in S,\, j \in A} \tilde{r}_{ij}(x)^2.$$

Each term is a composition of a squared norm and a quadratic residual. Consequently, the derivatives can be obtained by simple applications of the chain rule. For example, the gradient with respect to x_i is

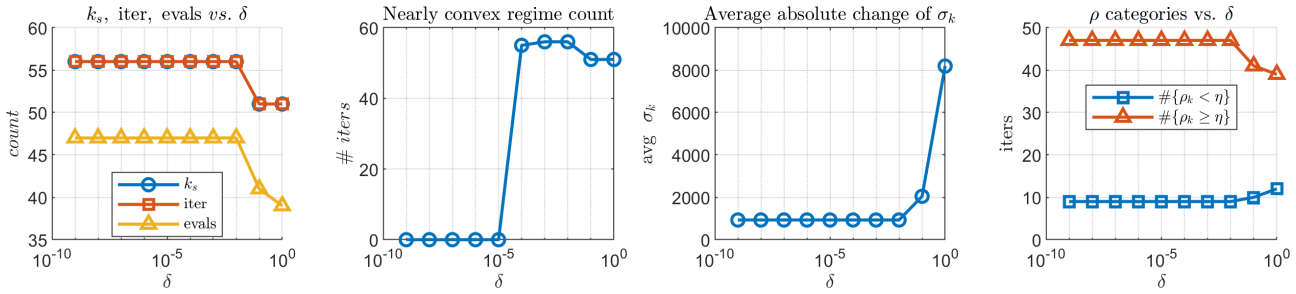
$$\nabla_{x_i} f(x) = 4\sum_{j \in S} r_{ij}(x)\,(x_i - x_j) + 4\sum_{j \in A} \tilde{r}_{ij}(x)\,(x_i - s_j),$$

and the Hessian consists of rank–one and rank–two updates built from the displacement vectors $x_i - x_j$ and $x_i - s_j$. Differentiating once more, the third derivative $\nabla^3 f(x)$ is a structured $3|\mathcal{S}| \times 3|\mathcal{S}| \times 3|\mathcal{S}|$ tensor whose nonzero blocks are linear combinations of tensors of the form $(x_i - x_j) \otimes (x_i - x_j) \otimes (x_i - x_j)$, and $(x_i - s_j) \otimes (x_i - s_j) \otimes (x_i - s_j)$, together with lower–rank symmetric permutations. Thus, although the problem is highly nonconvex, its third–order tensor has a highly structured form (and potentially low rank depending on the location of the anchor). Also, the third–order tensor can be applied to directions without ever forming the full dense tensor explicitly. This makes the model particularly suitable as a testbed for third–order methods.

A.4.3.2 Sensitivity of the Parameter δ in the AR3 + SoS-Convex Algorithm.

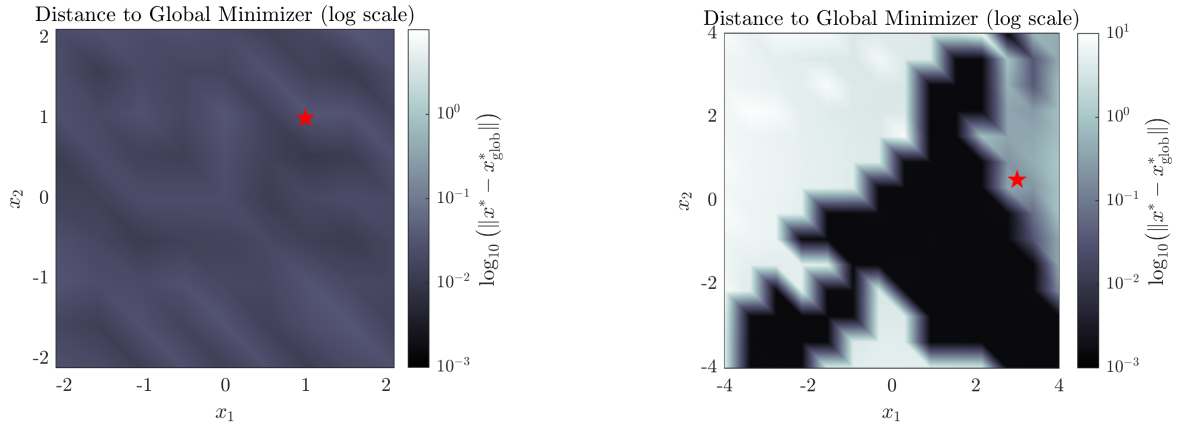
In Figure A.8, the four panels report: (i) the number of model evaluations, total iterations, and accepted steps required for convergence; (ii) the number of iterations entering the nearly convex regime $0 < \lambda_{\min}(H_k) < \delta$; (iii) the average magnitude of the regularization parameter σ_k; and (iv) the number of iterations with agreement ratio below and above the acceptance threshold η. The overall behaviour is stable for $\delta \in [10^{-9}, 1]$, with larger values of δ producing more iterations in the nearly convex regime and a mild increase in average σ_k.

Figure A.8: Sensitivity of the parameter δ in the AR3 + SoS-Convex algorithm on Problem 11 (GRD, $n = 3$).



A.4.3.3 Minimization of the Beale function using AR3 + SoS-Convex Algorithm.

The Beale function has three real critical points, including a unique global minimizer at $(3, 0.5)^\top$ and two saddle points.



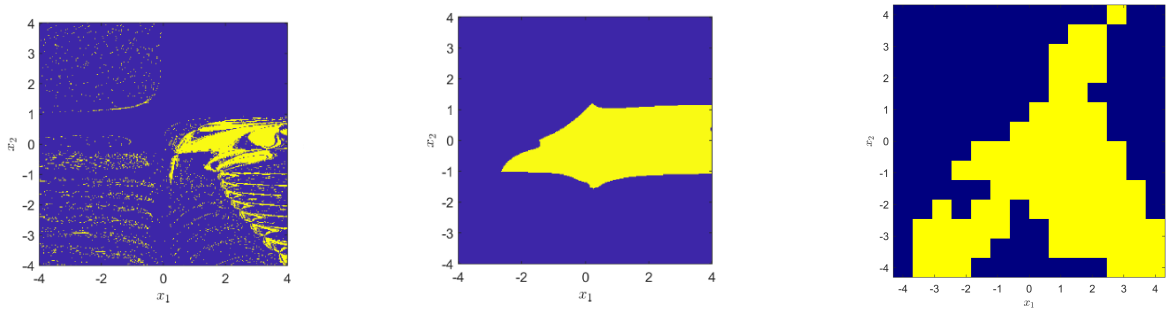
(a) Basins of attraction for the Chebyshev–Rosenbrock.

(b) Basins of attraction for the Beale function

Figure A.9: Global convergence behaviours across different initializations. Dark regions indicate convergence to the global minimizer, while lighter regions indicate no convergence after reaching the maximum number of iterations.

In Figure 6.5, yellow indicates initial points that converge to the global minimizer, i.e., points for which $\|x^k - \tilde{x}^*_{\text{glob}}\| \leq \text{Tol}$ where x^k is the outcome of the algorithm after 350 iterations and $\text{Tol} = 10^{-2}$.

Blue indicates no convergence after budgeted iterations[7]. We observe that the region of initializations that converge to the global minimum (yellow area) is larger for Algorithm 6.5.1 than for the classical Newton method, demonstrating an expansion of the basins of attraction from the second-order Newton method to third-order methods.



(a) Classical Newton [1, Fig. 5] (b) Third-order Newton [1, Fig. 5] (c) AR3 + SoS Convex method

Figure A.10: Global convergence behaviours of different initialization for minimizing the Beale function (a) the classical Newton method, (b) the third-order Newton method, and (c) Algorithm 6.5.1 with $\delta = 0.01$.

A.4.3.4 Minimization of Chebyshev–Rosenbrock function

Results in Table A.7 are averaged over 5 random initializations drawn uniformly from $[-4, 4]^n$ where n is the dimension of the Chebyshev–Rosenbrock function. A run is classified as *global* if the final iterate satisfies $\|x_k - x^*_{\text{glob}}\| < \texttt{Tol} = 10^{-2}$, and as *stationary* if the gradient norm satisfies $\|\nabla f(x_k)\| < 10^{-6}$. Any run that reaches the maximum iteration limit ($\texttt{max_iter} = 3000$) without satisfying either criterion is reported as "NoConv" (no convergence).

Table A.7: Summary of convergence behaviour and computational cost for all methods and dimensions.

n	Method	% Global	% Near Stationary	% No Conv	Avg Iter.	Avg Eval.
2	SoS + AR3	100	0	0	63	39
	ARC	100	0	0	17	17
	AR3 + ARC	100	0	0	17	13
3	SoS + AR3	100	0	0	67	49
	ARC	100	0	0	38	34
	AR3 + ARC	100	0	0	37	24
4	SoS + AR3	100	0	0	92	79
	ARC	100	0	0	110	72
	AR3 + ARC	100	0	0	86	49
5	SoS + AR3	100	0	0	210	187
	ARC	80	20	0	394	166
	AR3 + ARC	100	0	0	206	112
6	SoS + AR3	40	60	0	502	456
	ARC	0	100	0	1072	353
	AR3 + ARC	0	100	0	564	294
7	SoS + AR3	0	100	0	932	846
	ARC	0	100	0	1644	537
	AR3 + ARC	0	100	0	1055	543

[7]For Newton and the third-order Newton method, full numerical results appear in [1, Sec. 5.2]; we reproduce the corresponding plots in Figure 6.5 for completeness.

Bibliography

[1] Amir Ali Ahmadi, Abraar Chaudhry, and Jeffrey Zhang. Higher-order Newton methods with polynomial work per iteration. *arXiv preprint arXiv:2311.06374*, 2023.

[2] Amir Ali Ahmadi, Etienne De Klerk, and Georgina Hall. Polynomial norms. *SIAM Journal on Optimization*, 29(1):399–422, 2019.

[3] Amir Ali Ahmadi, Cemil Dibek, and Georgina Hall. Sums of separable and quadratic polynomials. *Mathematics of Operations Research*, 48(3):1316–1343, 2023.

[4] Amir Ali Ahmadi and Anirudha Majumdar. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

[5] Amir Ali Ahmadi, Alex Olshevsky, Pablo A Parrilo, and John N Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical programming*, 137:453–476, 2013.

[6] Amir Ali Ahmadi and Pablo A Parrilo. Sum of squares and polynomial convexity. In *Proc. IEEE Conf. Decision Control*, 2009.

[7] Amir Ali Ahmadi and Pablo A Parrilo. A complete characterization of the gap between convexity and SoS-convexity. *SIAM Journal on Optimization*, 23(2):811–833, 2013.

[8] Amir Ali Ahmadi and Jeffrey Zhang. On the complexity of finding a local minimizer of a quadratic function over a polytope. *Mathematical Programming*, 195(1-2):783–792, 2022.

[9] Orhan Arıkan, Regina S Burachik, and C Yalçın Kaya. Steklov regularization and trajectory methods for univariate global optimization. *Journal of Global Optimization*, 76(1):91–120, 2020.

[10] J. Arora. *Engineering Optimization*. Wiley, 2012.

[11] Brett W Bader. Tensor-krylov methods for solving large-scale systems of nonlinear equations. *SIAM journal on numerical analysis*, 43(3):1321–1347, 2005.

[12] Brett W Bader and Tamara G Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):635–653, 2006.

[13] Brett W Bader and Tamara G Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, 2008.

[14] Brett W Bader and Robert B Schnabel. Curvilinear linesearch for tensor methods. *SIAM Journal on Scientific Computing*, 25(2):604–622, 2003.

[15] Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2(1), 2009.

[16] Amir Beck. *First-Order Methods in Optimization*. SIAM, 2017.

[17] A. A. Bennett. On the convergence of Newton's method. *Annals of Mathematics*, 17(3):105–109, 1916.

[18] Albert A Bennett. Newton's method in general analysis. *Proceedings of the National Academy of Sciences*, 2(10):592–598, 1916.

[19] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[20] Eduardo G Birgin, José M Martinez, and Marcos Raydan. On the use of third-order models with fourth-order regularization for unconstrained optimization. *Computational Optimization and Applications*, 68(3):599–622, 2017.

[21] EG Birgin, JL Gardenghi, JM Martínez, and SA Santos. Fortran routines for testing unconstrained optimization software with derivatives up to third-order. ., 2018.

[22] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, and Sandra A Santos. On the use of third-order models with fourth-order regularization for unconstrained optimization. *Optimization Letters*, 14(4):815–838, 2020.

[23] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Ph L Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1):359–368, 2017.

[24] Ernesto G Birgin and Jan M Gentil. New and improved results for packing identical unitary radius circles within triangles, rectangles and strips. *Computers & Operations Research*, 37(7):1318–1327, 2010.

[25] Ernesto G Birgin, JM Martınez, and Débora P Ronconi. Optimizing the packing of cylinders into a rectangular container: a nonlinear approach. *European Journal of Operational Research*, 160(1):19–33, 2005.

[26] Ernesto G Birgin and José Mario Martínez. *Practical augmented Lagrangian methods for constrained optimization*. SIAM, 2014.

[27] Ernesto G Birgin and Francisco Nogueira Calmon Sobral. Minimizing the object dimensions in circle and sphere packing problems. *Computers & Operations Research*, 35(7):2357–2375, 2008.

[28] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

[29] Grigoriy Blekherman. Nonnegative polynomials and sums of squares. *Journal of the American Mathematical Society*, 25(3):617–635, 2012.

[30] Ali Bouaricha. Tensor methods for large, sparse unconstrained optimization. *SIAM Journal on Optimization*, 7(3):732–756, 1997.

[31] Ali Bouaricha and Robert B Schnabel. Tensolve: A software package for solving systems of nonlinear equations and nonlinear least squares problems using tensor methods. Technical report, Argonne National Lab., IL (United States). Mathematics and Computer Science Div., 1996.

[32] Charles Broadbent, Tianci Song, and Rui Kuang. Deciphering high-order structures in spatial transcriptomes with graph-guided tucker decomposition. *Bioinformatics*, 40(Supplement_1):i529–i538, 2024.

[33] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

[34] Brian Bullins. Fast minimization of structured convex quartics. *arXiv preprint arXiv:1812.10349*, 2018.

[35] Brian Bullins. Highly smooth minimization of non-smooth problems. In *Conference on Learning Theory*, pages 988–1030. PMLR, 2020.

[36] Regina S Burachik and C Yalçın Kaya. Steklov convexification and a trajectory method for global optimization of multivariate quartic polynomials. *Mathematical Programming*, 189(1):187–216, 2021.

[37] RH Byrd, J Nocedal, and RA Waltz. Knitro: An integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization*. Springer, 2006.

[38] Yubo Cai, Wenqi Zhu, Coralia Cartis, and Gioele Zardini. A globally convergent third-order newton method via unified semidefinite programming subproblems. *arXiv preprint arXiv:2603.09682*, 2026.

[39] Octavia Camps and Mario Sznaier. The interplay between big-data and sparsity in systems identification. In Jean-Paul Laumond, Nicolas Mansard, and Jean-Bernard Lasserre, editors, *Geometric and Numerical Foundations of Movements*, number 117 in Springer Tracts in Advanced Robotics, pages 133–159. Springer, New York, 2017.

[40] J-F Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.

[41] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

[42] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

[43] Coralia Cartis, Jaroslav M Fowkes, and Nicholas IM Gould. Branching and bounding improvements for global optimization algorithms with Lipschitz continuity properties. *Journal of Global Optimization*, 61(3):429–457, 2015.

[44] Coralia Cartis, Nicholas Ian Mark Gould, and Ph L Toint. Evaluation complexity of algorithms for nonconvex optimization. *MOS-SIAM Series on Optimization.*, 2022.

[45] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic overestimation methods for unconstrained optimization. *ora.ox.ac.uk*, 2007.

[46] 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.

[47] 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.

[48] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM Journal on Optimization*, 21(4):1721–1739, 2011.

[49] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Optimal Newton-type methods for nonconvex smooth optimization problems. *ERGO Technical Report*, 2011.

[50] 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.

[51] 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.

[52] Coralia Cartis, Nick I 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.

[53] Coralia Cartis, Nick IM Gould, and Ph L Toint. A concise second-order complexity analysis for unconstrained optimization using high-order regularized models. *Optimization Methods and Software*, 35(2):243–256, 2020.

[54] Coralia Cartis, Raphael Hauser, Yang Liu, Karl Welzel, and Wenqi Zhu. Efficient implementation of third-order tensor methods with adaptive regularization for unconstrained optimization: C. cartis et al. *Mathematical Programming Computation*, pages 1–72, 2026.

[55] Coralia Cartis and Lindon Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. *arXiv preprint arXiv:2102.12016*, 2021.

[56] Coralia Cartis and Wenqi Zhu. Cubic-quartic regularization models for solving polynomial subproblems in third-order tensor methods. *Mathematical Programming*, pages 1–53, 2025.

[57] Coralia Cartis and Wenqi Zhu. Second-order methods for quartically-regularised cubic polynomials, with applications to high-order tensor methods: C. cartis, w. zhu. *Mathematical Programming*, 215(1):669–715, 2026.

[58] Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences*, 25:536–538, 1847.

[59] Maolin Che, Yimin Wei, and Hong Yan. The computation of low multilinear rank approximations of tensors via power scheme and random projection. *SIAM Journal on Matrix Analysis and Applications*, 41(2):605–636, 2020.

[60] Haibin Chen, Hongjin He, Yiju Wang, and Guanglu Zhou. An efficient alternating minimization method for fourth degree polynomial optimization. *Journal of Global Optimization*, 82(1):83–103, 2022.

[61] Haibin Chen, Wenqi Zhu, and Coralia Cartis. Tensor-based dinkelbach method for computing generalized tensor eigenvalues and its applications. *Applied Numerical Mathematics*, 2026.

[62] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[63] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Independent component analysis based on higher-order statistics only. In *Proceedings of 8th Workshop on Statistical Signal and Array Processing*, pages 356–359. IEEE, 1996.

[64] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[65] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[66] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[67] Nikita Doikov and Yurii Nesterov. Local convergence of tensor methods. *Mathematical Programming*, 193(1):315–336, 2022.

[68] Nikita Doikov and Yurii E Nesterov. Inexact tensor methods with dynamic accuracies. In *ICML*, pages 2577–2586, 2020.

[69] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[70] Jean-Pierre Dussault. Arcq: a new adaptive regularization by cubics. *Optimization Methods and Software*, 33(2):322–335, 2018.

[71] Ivar Ekeland. On the variational principle. *Journal of Mathematical Analysis and Applications*, 47(2):324–353, 1974.

[72] Dan Feng and Thomas H Pulliam. Tensor-GMRES method for large systems of nonlinear equations. *SIAM Journal on Optimization*, 7(3):757–779, 1997.

[73] Henry B. Fine. On Newton's method of approximation. *Annals of Mathematics*, 17(3):97–100, 1916.

[74] Henry B Fine. On Newton's method of approximation. *Proceedings of the National Academy of Sciences*, 2(9):546–552, 1916.

[75] Roger Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[76] Jaroslav M Fowkes, Nicholas IM Gould, and Chris L Farmer. A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions. *Journal of Global Optimization*, 56(4):1791–1815, 2013.

[77] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, César A Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, et al. Near optimal methods for minimizing convex functions with Lipschitz p-th derivatives. In *Conference on Learning Theory*, pages 1392–1393. PMLR, 2019.

[78] PE Gill, W Murray, and MA Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 2005.

[79] Nicholas Gould and Jennifer Scott. A note on performance profiles for benchmarking software. *ACM Transactions on Mathematical Software (TOMS)*, 43(2):15, 2016.

[80] Nicholas IM Gould, Margherita Porcelli, and Philippe L Toint. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Computational optimization and applications*, 53:1–22, 2012.

[81] Nicholas IM Gould, Tyrone Rees, and Jennifer A Scott. Convergence and evaluation-complexity analysis of a regularized tensor-Newton method for solving nonlinear least-squares problems. *Computational Optimization and Applications*, 73(1):1–35, 2019.

[82] NIM Gould, T Rees, and JA Scott. A higher order method for solving nonlinear least-squares problems. *RAL Preprint RAL-P-2017–010, STFC Rutherford Appleton Laboratory*, 2017.

[83] Geovani N Grapiglia, Jinyun Yuan, and Ya-xiang Yuan. On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. *Mathematical Programming*, 152:491–520, 2015.

[84] Geovani Nunes Grapiglia and Yu Nesterov. Adaptive third-order methods for composite convex optimization. *SIAM Journal on Optimization*, 33(3):1855–1883, 2023.

[85] Geovani Nunes Grapiglia and Yurii Nesterov. Tensor methods for minimizing convex functions with Holder continuous higher-order derivatives. *arXiv preprint arXiv:1904.12559*, 2019.

[86] Geovani Nunes Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *Optimization Methods and Software*, pages 1–34, 2020.

[87] Geovani Nunes Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *Optimization Methods and Software*, 37(2):605–638, 2022.

[88] S Gratton and Ph L Toint. Adaptive regularization minimization algorithms with nonsmooth norms. *IMA Journal of Numerical Analysis*, 2022.

[89] Serge Gratton, Sadok Jerad, and Philippe L Toint. Yet another fast variant of Newton's method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 45(2):971–1008, 2025.

[90] Andreas Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[91] Mert Gürbüzbalaban and Michael L Overton. On Nesterov's nonsmooth Chebyshev–Rosenbrock functions. *Nonlinear Analysis: Theory, Methods & Applications*, 75(3):1282–1289, 2012.

[92] Nathan Halko, Per-Gunnar Martinsson, and Joel Tropp. Randomized low-rank approximation for symmetric indefinite matrices. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1255–1305, 2011.

[93] J William Helton and Jiawang Nie. Semidefinite representation of convex sets. *Mathematical Programming*, 122:21–64, 2010.

[94] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[95] David Hilbert. Über die darstellung definiter formen als summe von formenquadraten. *Mathematische Annalen*, 32(3):342–350, 1888.

[96] Reiner Horst and Hoang Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

[97] Yong Hsia, Ruey-Lin Sheu, and Ya-xiang Yuan. Theory and application of p-regularized subproblems. *Optimization Methods and Software*, 32(5):1059–1077, 2017.

[98] Florian Jarre. On Nesterov's smooth Chebyshev–Rosenbrock function. *Optimization Methods and Software*, 28(3):478–484, 2013.

[99] Leonid V. Kantorovich. On Newton's method for functional equations. *Doklady Akademii Nauk SSSR*, 59:1237–1240, 1948.

[100] Eleftherios Kofidis and Phillip A Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(3):863–884, 2002.

[101] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pages 1895–1904. PMLR, 2017.

[102] Masakazu Kojima. *Sums of squares relaxations of polynomial semidefinite programs*. Inst. of Technology, 2003.

[103] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[104] Tamara G Kolda and Jackson R Mayo. Shifted power method for computing tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1095–1124, 2011.

[105] Tamara G Kolda and Jackson R Mayo. An adaptive shifted power method for computing generalized tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1563–1581, 2014.

[106] M Kovcvara and M Stingl. Pennon: a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 2003.

[107] Dmitri E Kvasov and Yaroslav D Sergeyev. A univariate global search working with a set of Lipschitz constants for the first derivative. *Optimization Letters*, 3(2):303–318, 2009.

[108] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[109] Jean B Lasserre. The moment-SoS hierarchy. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3773–3794. World Scientific, 2018.

[110] Jean Bernard Lasserre. Optimisation globale et théorie des moments. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 331(11):929–934, 2000.

[111] Jean-Bernard Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.

[112] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

[113] JeanB Lasserre, Kim-Chuan Toh, and Shouguang Yang. A bounded degree SoS hierarchy for polynomial optimization. *EURO Journal on Computational Optimization*, 5(1-2):87–117, 2017.

[114] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[115] Kechen Li, Wenqi Zhu, Coralia Cartis, Tianbo Ji, and Shiwei Liu. SoS1: O1 and R1-like reasoning LLMs are sum-of-square solvers. *arXiv preprint arXiv:2502.20545*, 2025.

[116] Congzheng Liu and Wenqi Zhu. Newsvendor conditional value-at-risk minimisation: A feature-based approach under adaptive data selection. *European Journal of Operational Research*, 313(2):548–564, 2024.

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

[118] Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in neural information processing systems*, 24, 2011.

[119] Stefano Lucidi, Laura Palagi, and Massimo Roma. On some properties of quadratic programs with a convex quadratic constraint. *SIAM Journal on Optimization*, 8(1):105–122, 1998.

[120] Zhi-Quan Luo and Shuzhong Zhang. A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints. *SIAM Journal on Optimization*, 20(4):1716–1736, 2010.

[121] Terry Lyons and Andrew D McLeod. Higher order Lipschitz sandwich theorems. *arXiv preprint arXiv:2404.06849*, 2024.

[122] Boris Mariere, Zhi-Quan Luo, and Timothy N Davidson. Blind constant modulus equalization via convex optimization. *IEEE Transactions on Signal Processing*, 51(3):805–818, 2003.

[123] José Mario Martínez. Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM Journal on Optimization*, 4(1):159–176, 1994.

[124] José Mario Martínez and Marcos Raydan. Separable cubic modeling and a trust-region strategy for unconstrained minimization with impact in global optimization. *Journal of Global Optimization*, 63(2):319–342, 2015.

[125] José Mario Martínez and Marcos Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *Journal of Global Optimization*, 68:367–385, 2017.

[126] Leandro Martínez, Ricardo Andrade, Ernesto G Birgin, and José Mario Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of computational chemistry*, 30(13):2157–2164, 2009.

[127] Maike Meier, Yuji Nakatsukasa, Alex Townsend, and Marcus Webb. Are sketch-and-precondition least squares solvers numerically stable? *arXiv preprint arXiv:2302.07202*, 2023.

[128] Daniel K Molzahn and Ian K Hiskens. Sparsity-exploiting moment-based relaxations of the optimal power flow problem. *IEEE Transactions on Power Systems*, 30:3168–3180, 2015.

[129] Jorge J Moré, Burton S Garbow, and Kenneth E Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software (TOMS)*, 7(1):17–41, 1981.

[130] Jorge J Moré and Danny C Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[131] Theodore Samuel Motzkin. The arithmetic-geometric inequality. *Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965)*, 205:54, 1967.

[132] Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Technical report, CORE, 1985.

[133] Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming: Series A and B*, 39(2):117–129, 1987.

[134] AS Nemirovsky, DB Yudin, and ER DAWSON. Wiley-interscience series in discrete mathematics, 1983.

[135] Yu Nesterov. Squared functional systems and optimization problems. *High Performance Optimization*, 13:405–440, 2000.

[136] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Springer, 2004.

[137] Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186(1):157–183, 2021.

[138] Yurii Nesterov. Inexact high-order proximal-point methods with auxiliary search procedure. *SIAM Journal on Optimization*, 31(4):2807–2828, 2021.

[139] Yurii Nesterov. Superfast second-order methods for unconstrained convex optimization. *Journal of Optimization Theory and Applications*, 191(1):1–30, 2021.

[140] Yurii Nesterov. Inexact basic tensor methods for some classes of convex optimization problems. *Optimization Methods and Software*, 37(3):878–906, 2022.

[141] Yurii Nesterov. Inexact accelerated high-order proximal-point methods. *Mathematical Programming*, pages 1–26, 2023.

[142] Yurii Nesterov. Quartic regularity: Y. nesterov. *Vietnam Journal of Mathematics*, 53(3):553–575, 2025.

[143] Yurii Nesterov et al. Random walk in a simplex and quadratic optimization over convex polytopes. Technical report, CORE, 2003.

[144] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[145] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[146] Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[147] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13:271–369, 2004.

[148] Jiawang Nie. Certifying convergence of lasserre's hierarchy via flat truncation. *Mathematical Programming*, 142:485–510, 2013.

[149] Jiawang Nie. Optimality conditions and finite convergence of lasserre's hierarchy. *Mathematical programming*, 146:97–121, 2014.

[150] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

[151] Jorge Nocedal and Stephen J Wright. Nonlinear equations. *Numerical Optimization*, pages 270–302, 2006.

[152] Yannis Panagakis, Jean Kossaifi, Grigorios G Chrysos, James Oldfield, Mihalis A Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021.

[153] Panos M Pardalos and Stephen A Vavasis. Open questions in complexity theory for numerical optimization. *Mathematical programming*, 57(1):337–339, 1992.

[154] Taejun Park and Yuji Nakatsukasa. Randomized low-rank approximation for symmetric indefinite matrices. *arXiv preprint arXiv:2212.01127*, 2022.

[155] PA Parrilo and B Sturmfels. Algorithmic and quantitative real algebraic geometry. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60:83–99, 2003.

[156] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[157] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[158] Pablo A Parrilo and Bernd Sturmfels. Minimizing polynomial functions. *arXiv preprint math/0103170*, 2001.

[159] Pablo A Parrilo and Rekha R Thomas. *Sum of Squares: Theory and Applications*, volume 77. American Mathematical Soc., 2020.

[160] Imre Pólik and Tamás Terlaky. A survey of the S-lemma. *SIAM Review*, 49(3):371–418, 2007.

[161] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[162] Boris T Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181(3):1086–1096, 2007.

[163] Michael JD Powell. A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations*, 11:87–114, 1970.

[164] Liqun Qi, Zhong Wan, and Yu-Fei Yang. Global minimization of normal quartic polynomials based on global descent directions. *SIAM Journal on Optimization*, 15(1):275–302, 2004.

[165] Joseph Raphson. *Analysis Aequationum Universalis*. London, 1690.

[166] Bruce Reznick. Some concrete aspects of Hilbert's 17th problem. In *Contemporary Mathematics*, volume 253, pages 251–272. American Mathematical Society, 2000.

[167] Claus Scheiderer. Non-existence of degree bounds for weighted sums of squares representations. *Journal of Complexity*, 2006.

[168] Robert B Schnabel. Tensor methods for unconstrained optimization using second derivatives. *Mathematics of Computation*, 25(114):295–315, 1971.

[169] Robert B Schnabel and Ta-Tung Chow. Tensor methods for unconstrained optimization using second derivatives. *SIAM Journal on Optimization*, 1(3):293–315, 1991.

[170] Robert B Schnabel and Paul D Frank. Tensor methods for nonlinear equations. *SIAM Journal on Numerical Analysis*, 21(5):815–843, 1984.

[171] Krzysztof A Sikorski. *Optimal solution of nonlinear equations*. Oxford University Press, USA, 2001.

[172] Olha Silina and Jeffrey Zhang. An unregularized third order Newton method. *arXiv preprint arXiv:2209.10051*, 2022.

[173] Elias M Stein. *Singular integrals and differentiability properties of functions*. Princeton university press, 1970.

[174] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[175] RJ Vanderbei. Loqo: An interior point code for quadratic programming. *Optimization Methods and Software*, 1999.

[176] A Wächter and LT Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006.

[177] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.

[178] Chenwei Wang, Zhaoke Huang, Zelin Li, and Wenqi Zhu. Recover cell tensor: Diffusion-equivalent tensor completion for fluorescence microscopy imaging. *ICLR 2026*, 2026.

[179] Chenwei Wang, Renjie Xu, Yulin Huang, Jifang Pei, Chuan Huang, Wenqi Zhu, and Jianyu Yang. Limited-data sar atr causal method via dual-invariance intervention. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[180] Hongcheng Wang and Narendra Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, 76:217–229, 2008.

[181] Martin Weiser, Peter Deuflhard, and Bodo Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimisation Methods and Software*, 22(3):413–431, 2007.

[182] Tillmann Weisser, Jean B Lasserre, and Kim-Chuan Toh. Sparse-BSOS: a bounded degree SoS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, 10:1–32, 2018.

[183] Karl Welzel and Raphael A Hauser. Generalizing quasi-Newton updates to higher-order derivative tensors. *arXiv preprint arXiv:2301.11678*, 2023.

[184] Karl Welzel, Yang Liu, Raphael A Hauser, and Coralia Cartis. Local convergence of adaptively regularized tensor methods. *arXiv preprint arXiv:2510.25643*, 2025.

[185] Jorge Nocedal Stephen J Wright. Numerical optimization, 2006.

[186] Yangyang Xu and Wotao Yin. Block proximal gradient method for convex and nonconvex optimization. *SIAM Journal on Optimization*, 25(2):1388–1416, 2013.

[187] V. A. Yakubovich. S-procedure in nonlinear control theory. *Vestnik Leningrad University*, 1:62–77, 1971.

[188] Ya-xiang Yuan. A review of trust region algorithms for optimization. *IMA Journal of Numerical Analysis*, 11(1):1–34, 2000.

[189] Yang Zheng, Giovanni Fantuzzi, and Antonis Papachristodoulou. Sparse sum-of-squares (SoS) optimization: A bridge between DSOS/SDSOS and SoS optimization for sparse polynomials. In *2019 American Control Conference (ACC)*, pages 5513–5518. IEEE, 2019.

[190] Jinling Zhou, Xin Liu, Jiawang Nie, and Xindong Tang. A tight SDP relaxation for the cubic-quartic regularization problem. *arXiv preprint arXiv:2511.00168*, 2025.

[191] Wenqi Zhu and Coralia Cartis. Quartic polynomial sub-problem solutions in tensor methods for nonconvex optimization. In *NeurIPS 2022 Workshop*, 2022.

[192] Wenqi Zhu and Coralia Cartis. Global convergence of high-order regularization methods with sums-of-squares Taylor models. *arXiv preprint arXiv:2404.03035*, 2024.

[193] Wenqi Zhu and Coralia Cartis. Global optimality characterizations and algorithms for minimizing quartically-regularized third-order Taylor polynomials. *arXiv preprint arXiv:2504.20259*, 2025.

[194] Wenqi Zhu and Yuji Nakatsukasa. Convergence and near-optimal sampling for multivariate function approximations in irregular domains via vandermonde with Arnoldi. *IMA Journal of Numerical Analysis*, page draf055, 2025.