UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ
DEGLI STUDI
DI PERUGIA

iNδAM
Istituto Nazionale
di Alta Matematica

Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA
IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN MATEMATICA
CICLO XXX

Sede amministrativa Università degli Studi di Firenze
Coordinatore Prof. Graziano Gentili

Levenberg-Marquardt methods for the solution of noisy nonlinear least squares problems

Settore Scientifico Disciplinare MAT/08

Dottorando:
Elisa Riccietti
Elisa Riccietti

Tutore
Prof. Stefania Bellavia

Coordinatore
Prof. Graziano Gentili

Anni 2014/2017

Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA
IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN MATEMATICA
CICLO XXX

Sede amministrativa Università degli Studi di Firenze
Coordinatore Prof. Graziano Gentili

Levenberg-Marquardt methods for the solution of noisy nonlinear least squares problems

Settore Scientifico Disciplinare MAT/08

Dottorando:	**Tutore**
Elisa Riccietti	Prof. Stefania Bellavia

Coordinatore
Prof. Graziano Gentili

Anni 2014/2017

Acknowledgments
Ringraziamenti

I would like to thank Andreas Fischer and Luis Vicente for agreeing to review my thesis and for their useful comments and suggestions on this manuscript.

Vorrei ringraziare tutte le persone che mi sono state vicine in questo percorso.

Prima di tutto la mia relatrice Stefania per avermi introdotto al mondo della ricerca e per avermi insegnato come muovermi da sola in questo mondo, per la sua costante presenza e disponibilità, e per avermi consigliato nei momenti difficili e nelle scelte complicate. Ringrazio anche tutto il gruppo di ottimizzazione di Firenze, Margherita, Benedetta e Alessandra per il piacevole ambiente di lavoro.

Vorrei ringraziare anche il prof. Graziano Gentili per la sua disponibilitá e gentilezza e per il suo costante supporto con la burocrazia, le segretarie Tiziana e Francesca per le piacevoli chiacchierate, il prof. Sergio Vessella e la prof. Elisa Francini per le loro pazienti consulenze sui problemi inversi.

Vorrei ringraziare il prof. Andrea Arnone per avermi dato l'opportunità di lavorare su un interessante problema industriale, e di scoprire un diverso aspetto del mondo della ricerca. Ringrazio i ragazzi del gruppo di Ingegeneria Industriale, per le piacevoli giornate a Santa Marta, i caffè al ginseng, per aver cercato di farmi una cultura ingegneristica, e per avermi offerto una scrivania!

Ensuite, je voudrais remercier toute l'équipe APO de l'ENSEEIHT de Toulouse, pour m'avoir accueillie pendant six mois, surtout les Italiens, parmi lesquels, naturellement, je compte aussi Patrick! Je remercie également Charlie, Clément, Florent, Grégoire, Lucie et Romain pour les belles soirées passées ensemble, et Thibaut pour m'avoir aidé à pratiquer mon français. Un remerciement spécial va à Serge, pour les intéressantes discussions scientifiques pendant ma thèse, et pour le passionnant postdoc qu'il m'a proposé, mais aussi pour promouvoir les échanges internationaux, y compris de nature non académique.

A questo proposito ringrazio Theo, per essere arrivato inaspettatamente nella mia vita e averla stravolta con il suo sorriso ♡.

Infine ringrazio le mie amiche Silvia e Sofia, e tutta la mia famiglia per il continuo sostegno, soprattutto il mio cognato preferito Michele e mia sorella Eleonora. Dedico un pensiero speciale ai miei genitori Susanna e Paolo, che mi hanno sempre sostenuto, anche quando le mie scelte mi hanno portato lontano.

Abstract

In this thesis, we investigate the numerical resolution of noisy nonlinear least squares problems. We devise novel, specialized variants of Levenberg-Marquardt methods to solve two classes of noisy problems: ill-posed problems and large scale problems whose objective function is expensive to evaluate and can be replaced by cheaper noisy approximations. We propose three different approaches: a *regularizing Trust-Region* approach, an *elliptical regularizing Trust-Region* approach and a *Levenberg-Marquardt method for large scale problems with dynamic noise*.

The first two methods are intended to tackle ill-posed least squares problems with noisy data. Such problems are challenging, because continuous dependence on the data does not hold for them. It is therefore necessary to design ad hoc regularizing strategies for their stable solution. The *regularizing Trust-Region* approach aims at solving zero residual problems. We show how it represents an improvement over existing Levenberg-Marquardt methods in the literature and how it is more robust compared to them. The *elliptical regularizing Trust-Region* approach aims at solving small residual problems. To the best of our knowledge, there are no other existing methods in the literature designed to handle ill-posed least squares problems with nonzero residual. We theoretically prove regularizing and local convergence properties of these methods under mild assumptions, and then numerically validate them on different problems.

We then turn to large scale problems with an expensive objective function that can be replaced by cheaper approximations of increasing accuracy. Such approximations are used as objective functions of a sequence of noisy least squares problems. We design a novel *Levenberg-Marquardt method for large scale problems with dynamic noise* that computes a solution of the original problem by solving this sequence of noisy problems. The proposed method is able to handle noisy functions and gradients and can consequently solve the problem at a greatly reduced computational cost. We are not aware of any other method specially designed to solve least squares problems with noisy functions and gradients for which both local and global convergence are proved. We validate the numerical behaviour of the method on problems arising from data assimilation and real-life machine learning applications.

Contents

Acknowledgments / Ringraziamenti **v**

Abstract **vii**

Contents **ix**

List of Algorithms **xiii**

List of Tables **xv**

List of Figures **xvii**

Introduction **1**

I General background **7**

1 Least squares problems **9**

1.1 Linear least squares problems 11

1.1.1 Singular value decomposition and least squares problems . 12

1.1.2 Solution techniques 13

2 Numerical methods **15**

2.1 Newton's method 15

2.2 Trust-Region methods 16

2.2.1 Solution of the Trust-Region subproblem 19

2.2.1.1 Exact solution 19

2.2.1.2 Approximate solution 22

2.3 Gauss-Newton method 24

2.4 Levenberg-Marquardt method 25

2.4.1 Inexact Levenberg-Marquardt method 27

2.4.2 Local convergence and complexity 29

2.4.3 Trust-Region method for least squares 31

2.4.4 Tikhonov method 33

II Ill-posed nonlinear least squares problems 35

3 Introduction to Part II 37

4 Zero-residual problems 43

4.1 Regularizing Levenberg-Marquardt method for ill-posed problems . 49

4.2 Regularizing Trust-Region method 53

4.3 Local behaviour of the Trust-Region method 57

4.3.1 Noise free case 58

4.3.2 Noisy case 62

4.4 Numerical results 65

4.4.1 Test problems definition 66

4.4.2 Implementation of Algorithm 4.2 67

4.4.3 Validation of the regularizing properties of the Trust-Region method 69

4.4.4 Comparison with Levenberg-Marquardt method 71

4.4.5 Comparison with standard Trust-Region method 74

4.4.6 Choice of the quadrature formula 77

4.5 Chapter conclusion 78

5 Non-zero residual problems 81

5.1 Monotonic error decrease 85

5.2 The method 86

5.2.1 Choice of the Trust-Region radius 88

5.3 Convergence analysis 90

5.3.1 Noise free case 91

5.3.2 Noisy case 94

5.4 Constrained case 95

5.5 Numerical results 97

5.6 Chapter conclusion 105

6 Infinite dimensional setting 107

6.1 Procedures in the Hilbert setting 108

6.2 Convergence to a solution of the infinite dimensional problem . . . 110

6.3 A model problem 111

III Large scale noisy nonlinear least squares problems 115

7 Introduction to Part III 117

8 Levenberg-Marquardt method for problems with dynamic noise 121

8.1 Global convergence 125

8.2 Local convergence 131
8.3 Complexity 134
8.4 Chapter conclusion 136

9 Application to data assimilation and machine learning 137
9.1 Data assimilation problems 140
9.1.1 Data assimilation problem with subsampling 141
9.1.2 Data assimilation problem with non-diagonal covariance matrix. 145
9.2 Machine learning problems 147
9.2.1 CINA dataset 148
9.2.2 A real life machine learning problem 152
9.2.2.1 Parametric design of a family of centrifugal pumps 152
9.2.2.2 Classification problem resolution 156
9.3 Chapter conclusion 158

Conclusion 161

Appendix 165

Publications related to the thesis 169

References 171

List of Algorithms

2.1 k-th iteration of basic Trust-Region algorithm for problem (1.1) . . . 17

2.2 l-th iteration of Newton's method applied to $\Phi_2(\lambda) = 0$. 22

4.1 k-th iteration of the regularizing Levenberg-Marquardt method for problem (4.2) 49

4.2 k-th iteration of the regularizing Trust-Region method for problem (4.2) 55

5.1 k-th iteration of the elliptical regularizing Trust-Region method for problem (3.1) 91

5.2 k-th iteration of the elliptical regularizing Trust-Region method for problem (5.37) 96

8.1 Levenberg-Marquardt method for problem (1.2) using dynamic noise 123

List of Tables

4.1 Results obtained by the regularizing Trust-Region (RTR) method and the regularizing Levenberg-Marquardt (RLM) method with noise $\delta = 10^{-4}$ and different initial guesses. 70

4.2 Results obtained by the regularizing Trust-Region (RTR) method and the regularizing Levenberg-Marquardt (RLM) method with noise $\delta = 10^{-2}$ and different initial guesses. 71

9.1 Performance of LMN for test problem $\mu = 0.5$ and $K_0 = 2000$. 143

9.2 Performance of LMN for test problem $\mu = 0.5$ and $K_0 = 5000$. 143

9.3 Performance of LMN for test problem $\mu = 0.01$. 145

9.4 Performance of LMN for data assimilation problem with non-diagonal background-error covariance matrix. 146

9.5 Performance of LMN for machine learning test problem for different values of K_*. 149

9.6 Comparison of LMN with approximated and estimated noise. 152

9.7 Parametric design test problem. Comparison of FLM and LMN performance. 158

List of Figures

4.1 Convex hull of $a(x)$ and $b(x)$, Figure 3.1 in [97]. 44

4.2 Effect of q-condition on the step length. 47

4.3 Problem P1, $x_0 = 0e$, $\delta = 10^{-2}$. Upper part: Solution approximation computed with $\Delta_k = \frac{1-q}{\|B_k\|}\|g_k\|$ (dotted line) and true solution (solid line) (left) and monotonic decrease of the error $\|x_k^\delta - x^\dagger\|$ (right). Bottom part: Decrease of the residual $\|F(x_k) - y^\delta\|$, for $\Delta_k = \frac{1-q}{\|B_k\|}\|g_k\|$ (left) and adaptive choice of the Trust-Region radius (4.45) (right). 68

4.4 Regularizing Trust-Region applied to P2, $x_0 = 0e$, $\delta = 10^{-4}$. *Left*: values $q_k = \frac{||F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k||}{||F(x_k^\delta) - y^\delta||}$ (marked by an asterisk) and value of $q = 1.1/\tau$ (solid line) versus the iterations. *Right*: semilog plot of the error $||x_k^\delta - x^\dagger||$ versus the iterations (on the right). 72

4.5 Regularizing Trust-Region applied to P1, $x_0 = 0e$ (top left), P2, $x_0 = 0e$ (top right), P3, $x_0 = x_0(\alpha) = x_0(1.25)$ (lower left) and to P4, $x_0 = x_0(\beta, \chi) = x_0(0.5, 0)$ (lower right). Log plot of the error $||x_{k_*(\delta)}^\delta - x^\dagger||$ versus the noise level δ. 72

4.6 Regularizing Trust-Region (left) and regularizing Levenberg-Marquardt (right), true solution (solid line) and approximate solutions (dotted line). *Upper part*: P1, $\delta = 10^{-2}$, $x_0 = 0e$. *Lower part*: P3, $\delta = 10^{-2}$, $x_0 = x_0(\alpha) = x_0(1.25)$. 73

4.7 True solution (solid line) and approximate solutions (dotted line) computed by the regularizing Trust-Region method (on the left) and the regularizing Levenberg-Marquardt method (on the right). Upper part: problem P2, $\delta = 10^{-2}$, $x_0 = 0e$; lower part: problem P4, $\delta = 10^{-2}$, $x_0 = x_0(\beta, \chi) = x_0(0.5, 0)$. 74

4.8 Problem P4, $\delta = 10^{-2}$, $x_0 = x_0(\beta, \chi) = x_0(1.5, 0)$. True solution (solid line) and approximate solution (dotted line) computed by the regularizing Levenberg-Marquardt method for values of $q = 0.67, 0.70, 0.73, 0.87$. . 75

4.9 True solution (solid line) and approximate solutions (dotted line) computed by the regularizing Trust-Region method (on the left) and the standard Trust-Region method (on the right). (a)-(b) problem P1, $\delta = 10^{-2}, x_0 = 0e$; (c)-(d) problem P2, $\delta = 10^{-2}, x_0 = 0e$; (e)-(f) problem P3, $\delta = 10^{-2}$, $x_0 = x_0(1.25)$; (g)-(h) problem P4, $\delta = 10^{-2}$, $x_0 = x_0(0.5, 0)$. . . . 76

4.10 Problem P2, $x_0 = 1e$, e_I error inside the interval, e_E error in the endpoints. Upper part: $\delta = 0$; Lower part: $\delta = 10^{-2}$. Rectangular rule (left) and trapezoidal rule (right). 79

4.11 Solution approximation and true solution, $\delta = 10^{-2}$. Upper part: problem P2; Lower part: problem P4. Rectangular rule (left) and trapezoidal rule (right). 79

5.1 Effect of the generalized q-condition 5.5 on the step length. 88

5.2 Numerical evidence for Assumption 5.6: plot of $\|\nabla f(\tilde{x}) - \nabla f(x_k) - J(x_k)^T J(x_k)(\tilde{x} - x_k)\|$ (dashed line) and $(0.1\|\tilde{x} - x_k\| + \sigma)\|\nabla f(\tilde{x}) - \nabla f(x_k)\|$ (solid line) for $\tilde{x}$ randomly chosen in $B_\rho(x_k)$, $\rho = 0.3$, for problem R1 (top left, $\sigma = 4 \cdot 10^{-3}$), R2 (top right, $\sigma = 0.1$), R3 (bottom left, $\sigma = 0.48$), R4 (bottom right, $\sigma = 0.54$). 101

5.3 (a): Test problem R1, $\delta = 10^{-2}$. Obtained values of q_k. (b): Test problem R4, $\delta = 10^{-2}$. Dependence of the relative error between computed solution $x^\delta_{k^*(\delta)}$ and x^* on the parameter q. 102

5.4 Test R2, $\delta = 10^{-2}$. Upper part: The procedure is stopped when the discrepancy principle is satisfied, the norm of the error decreases monotonically (a). Lower part: The procedure is not stopped when the discrepancy principle is satisfied, the norm of the relative error increases (b) and a solution of the noisy problem is approached (c). 103

5.5 Test R2. Upper part: solution approximation x^* obtained with exact data (a). Lower part: solution approximations obtained with standard Trust-Region (b) and elliptical regularizing Trust-Region (c) for $\delta = 10^{-2}$. 104

5.6 Contour plots of solution approximations of problem R2 for different noise levels, $\delta = 0, 10^{-2}, 10^{-4}$. 104

5.7 (a): Problem R1. Comparison of solution approximations computed with $\delta = 10^{-3}, 10^{-2}$ and solutions x^* and $x^\dagger = [c(x_1), \dots, c(x_N)]^T$ with c in (5.39). (b): Problem R4. Reduction of the relative error $\frac{\|x^* - x^\delta_{k^*(\delta)}\|}{\|x^*\|}$ with the noise level δ. 105

5.8 Test problem R3 with $\delta = 10^{-2}$: plot of the true solution $x^\dagger$ and of the computed solution $x^\delta_{k^*(\delta)}$ for $\delta = 10^{-2}$ (a), regularization parameters λ_k (b). 105

9.1 Problem $\mu = 0.5$. Comparison of true state (solid line) and computed solution (dotted line) computed by FLM (up left), LMN with $K_0 = 5000$ and $\kappa_d = 10$ (up right), LMN with $K_0 = 2000$ (bottom left) and $K_0 = 5000$ (bottom right) with $\kappa_d = 10000$. 144

9.2 Left plot: $save_f$ (lines marked by a star) and $save_p$ (lines marked by a circle) for tests in Tables 9.1 (solid lines), 9.2 (dotted lines). Right plot: corresponding solution accuracy. 144

9.3 Problem $\mu = 0.5$, $K_0 = 2000$, $\kappa_d = 10$. Left: log plot of noise level versus iterations. Right: log plot of the RMSE versus iterations. 145

9.4 LMN applied to data assimilation problem with non-diagonal matrix. Left: solution approximation. Right: total number of `cgls` iterations. . 146

9.5 Values of n_k (dotted line) and number of linear iterations (dashed line) per outer iteration, for $K_* = 1.1$ (up left), $K_* = 2$ (up right) and $K_* = 3.5$ (bottom). 150

9.6 Left: values of $|X_k|$ versus iterations for different values of K_*. Right: Comparison of matrix-vector cost counter for FLM (solid line) and LMN with various choices of K_*. 151

9.7 Left: comparison of approximated noise level (solid line) and estimated noise level (dashed line) during run of LMN. Right: decrease in the model $m_k(x_k) - m_k(x_k + p_k^{LM})$ versus iterations (dashed line) compared to decrease of $\frac{1}{2}\lambda_k \|p_k^{LM}\|^2$ (solid line). 151

Introduction

Nonlinear least squares problems arise in many practical applications. They can be stated as

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|R(x)\|^2$$

where $R : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function. We consider the case $m \geq n$, which is the most common in applications as they arise, for example, from data-fitting issues. Classical well-studied iterative methods to solve this class of problems are Gauss-Newton and Levenberg-Marquardt methods [27, 80, 84].

In this thesis, we consider the case in which the exact values of the function f and of its derivatives are not known. This is the case in many problems of practical interest. Indeed, in many applications, only approximations to exact function values, up to a given accuracy, are available. For example, this may be due to the fact that the function depends on measured data, which are usually affected by errors, arising from instruments sensitivity or measurement errors. In this case, the exact function is unknown and it is not possible to recover it, by any means. In other applications, the function can actually be computed, but it is computationally demanding to do it. Then, it may be more convenient to avoid this computation when possible, at least at the initial stage of the iterative process, and replace the function with some cheaper approximation.

This thesis deals with the study and implementation of suitable methods that aim to recover an approximation to a solution x^* of a least squares problem, without using the exact values of f. Our methods would rather build a sequence of approximations approaching x^*, by considering only problems built employing the available approximations f_δ to f:

$$\min_{x\in\mathbb{R}^n} f_\delta(x),$$

that we will refer to as *noisy problems*, as opposed to the *unperturbed or original* problem that is given by the minimization of f.

We will focus on the class of Levenberg-Marquardt methods and devise approaches suitable to handle two different broad classes of noisy nonlinear least squares problems, that fit in the outlined framework.

1. *Ill-posed problems.* The first class is that of least squares problems arising from the fitting of measured data. For this kind of problems f_δ is the squared norm of the difference between the measured data y^δ and the model F we want to fit them with:
$$f_\delta(x) = \frac{1}{2}\|F(x) - y^\delta\|^2.$$
In practical contexts, the measured data are not exact measurements of the quantity that we would actually like to measure, but rather an approximation y^δ of the true data y, affected by noise, that is such that
$$\|y - y^\delta\| \le \delta,$$
where δ is the noise level. It is usually assumed that the noise level δ is known and represents a fixed quantity arising from measurement errors and instruments sensitivity. Problems arising in this context are usually ill-posed, in the sense that the solutions of the problem do not depend continuously on the data. Then, particular attention should be devoted to the development of stable methods for their solution. In fact, it is not possible to approximate the solutions of the original problem with those of the noisy problem, computed applying a classical method directly to it. Indeed, its solutions may be bad approximations to solutions of the unperturbed problem. Then, a method should aim at recovering partial information about the solution as stably as possible. The final accuracy of the solution cannot be expected to be better than what the noise allows for [55].

2. *Large scale noisy problems.* The second class of problems is that of large scale problems for which the exact values of the objective function cannot be employed along all the optimization process, either because the exact function is unknown or because its evaluation is expensive. This situation arises in many applications. This is for example the case when the exact objective function cannot be computed and only noisy approximations are available, or when the objective function evaluation is a sum over a large number of terms or the result of a computation whose accuracy can vary and must be specified in advance. In these latter cases, the evaluation of the function may be expensive, but it can also be made cheaper considering a subset of the addends or asking for a lower accuracy level. We consider any problem for which an exact evaluation of the function is computationally demanding and can be replaced by possibly cheaper approximations. As opposed to the first class of problems, the noise is not assumed to be limited to the data, so that the Jacobian matrix is also affected by noise. The noise is in general defined as the accuracy of the function approximations
$$|f_\delta(x) - f(x)| \le \delta,$$
and here we assume that it is possible to decrease it if needed.

Aim of the thesis

The thesis is devoted to the development of variants of Levenberg-Marquardt type methods to solve the two classes of problems presented above. We propose three different approaches, two for the first class and one for the second class of problems:

- *Regularizing Trust-Region*. We propose a Trust-Region approach for the stable solution of zero residual nonlinear ill-posed problems, which takes its origin from the Levenberg-Marquardt method presented in [50]. Our method is based on an adaptive choice of the regularization parameters that are chosen to satisfy a specific regularizing condition. A suitable choice of the Trust-Region radius is devised that guarantees an indirect and automatic choice of the parameters satisfying the regularizing condition.
- *Elliptical regularizing Trust-Region*. We propose an nonstationary iterated Tikhonov procedure to stably solve nonlinear ill-posed problems with small residual. We propose also an elliptical Trust-Region reformulation that allows for an automatic setting of the regularization parameters.
- *Levenberg-Marquardt method for large scale problems with dynamic noise*. We introduce an inexact Levenberg-Marquardt method aimed at solving a nonlinear least squares problem relying solely on approximations f_δ to f. The method builds a sequence of solutions approximations considering noisy problems, whose objective functions are approximations of known and increasing accuracy to the exact objective function. This feature is exploited by asking the lowest possible accuracy in the value of the objective that is sufficient to guarantee progress of the minimization, with the ultimate goal of saving computing time.

We study theoretical properties of the proposed methods, like convergence properties, complexity, and regularizing properties. In this context, one does not usually look for fast locally convergent methods as the need of regularizing and handling the noise requires to slow down the convergence to avoid approaching noisy solutions.

We also perform a numerical validation of the proposed procedures and we show numerical evidence of their properties on several examples of least squares problems, like Fredholm equations of the first kind, parameter identification problems and problems arising in geophysics, data assimilation and machine learning, one of which arises from a real life application in the domain of the parametric design of turbomachinery components.

Contributions of the thesis

- We analyze the practical implementation of the method in [50] that was not considered in the original paper or in related articles. Specifically we discuss how to compute the regularization parameters in a reliable way.

- The regularizing Trust-Region approach we propose represents an improvement over the method in [50], as it is shown to be more robust compared to it. The two methods indeed are based on similar conditions for the choice of the regularization parameter. However, the condition on which the Levenberg-Marquardt method in [50] is based may fail to be satisfied for iterates not close enough to a solution, while the condition we adopted can always be satisfied. It can be enforced by a suitable choice of the Trust-Region radius, while it is not straightforward to understand how to enforce it with a Levenberg-Marquardt method. The Trust-Region approach is also shown to be less-dependent on the free parameters of the method.

- The most part of the literature on ill-posed nonlinear least squares deals with zero residual problems, even if nonzero residual problems frequently appear in applications, especially when a natural phenomenon is represented through a mathematical model. While it is common especially in the literature on linear problems to incorporate the modelling error in the data error and solve the problem as a zero residual problem, our elliptical regularizing Trust-Region is an ad hoc method for nonzero residual problems. It has the advantage that an estimation of the modelling error for the computation of the regularization parameters is not required. To our knowledge, approaches designed to deal with ill-posed nonzero residual problems have never been proposed in the literature.

- This work represents also a contribution on the study of local convergence properties of Trust-Region methods. The local analysis of the methods for ill-posed problems is indeed performed under assumptions different and somehow weaker than those usually used in the literature.

- The Levenberg-Marquardt method for noisy problems provides a method to solve least squares problems with both noisy function and gradients. Several methods have been designed for noisy unconstrained minimization problems with approximated gradient and Hessian, both in the case in which it is possible to assume the noise to vary (namely in problems arising from machine learning) [5, 16] and in the case of fixed noise [70]. On the contrary, we are not aware of methods specially designed for noisy nonzero residual nonlinear least squares problems, for which both local and global convergence is proved.

- The proposed Levenberg-Marquardt method allows considerable savings in terms of function evaluations and matrix vector products compared to inexact Levenberg-Marquardt method employing the exact objective function and Jacobian.

Organization of the thesis

The thesis is divided into three parts.

Part I represents an introductory part, in which least squares problems and the methods under study are presented. A review of classical results in optimization theory are reported, to make the understanding of the methods and theory presented in the following chapters easier.

Part II is devoted to ill-posed nonlinear least squares problems. The zero residual case is considered in Chapter 4 and the nonzero residual case in Chapter 5. In Chapter 6 we briefly discuss the extension of the procedures proposed in the previous chapters to an infinite-dimensional Hilbert setting.

Part III is devoted to large scale noisy least squares problems. The proposed Levenberg-Marquardt method is presented. This part of the thesis has been realized in collaboration with Prof. Serge Gratton, during my six-months visit to INP-IRIT Toulouse [S1].

Notations

Given $f : \mathbb{R}^n \to \mathbb{R}$, we will denote with $\nabla f \in \mathbb{R}^n$ its gradient and with $\nabla^2 f \in \mathbb{R}^{n\times n}$ its Hessian matrix. Given $R : \mathbb{R}^n \to \mathbb{R}^m$, we will denote with $J \in \mathbb{R}^{m\times n}$ its Jacobian matrix.

If not differently specified, $\|\cdot\|$ denotes the Euclidean norm.

We denote with $\mathbb{R}^+ = \{x \in \mathbb{R} \,|\, x \geq 0\}$ and given $x^* \in \mathbb{R}^n$ with $B_r(x^*) = \{x \in \mathbb{R}^n \,|\, \|x - x^*\| \leq r\}$ a ball of radius r and centre x^*.

Given a matrix A, we will denote with A^+ its Moore-Penrose pseudoinverse, with $\texttt{rank}(A)$ its rank, with $\mathscr{R}(A)$ its range and with $\mathscr{R}(\mathscr{A})^\perp$ the orthogonal complement of $\mathscr{R}(\mathscr{A})$.

Given $m \geq n$ and a vector $v = [v_1, \dots, v_n]$, $\texttt{diag}(v) \in \mathbb{R}^{m\times n}$ denotes a matrix with elements v_i $i = 1, \dots, n$ on the diagonal, i.e. if $D = \texttt{diag}(v)$, $D_{ii} = d_i$ and the other entries are zero.

We denote with I the identity matrix. If a subscript is present, it specifies the dimension of the matrix, otherwise we assume I to have dimension $n \times n$.

Part I

General background

CHAPTER

1 Least squares problems

Least squares problems are a special case of unconstrained minimization problems. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, that is usually addressed as the *objective function*, an unconstrained minimization problem has the following form:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1.1}$$

A least squares problem corresponds to a specific choice of f, namely given $R_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \dots, m$, f is the sum of squares:

$$f(x) = \frac{1}{2} \sum_{i=1}^{m} R_i(x)^2.$$

If we denote as $R(x) = [R_1(x), R_2(x), \dots, R_m(x)]^T$, a least squares problem can be formulated as

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|R(x)\|^2, \tag{1.2}$$

with $R : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}^+$. If function R, and consequently function f, is nonlinear we gain a nonlinear least squares problem.

A solution of such problem is a minimizer, i.e. a point x^* in which the lowest possible value for the function is achieved: $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$. Such a point is called a *global minimizer*. However, finding a global minimizer is usually expensive and in many applications it is not even necessary to find the lowest possible value of the function. In most applications it is sufficient to find what is called a *local minimizer*, namely a point x^* for which it exists a neighbourhood $B_r(x^*)$, such that $f(x^*) \leq f(x)$ for all $x \in B_r(x^*)$. In the following, when we refer to a solution of a least squares problem like (1.2), we are then referring to a local minimizer. Local minimizers can be characterized by the following optimality conditions [85, Chapter 2]. We denote with ∇f and $\nabla^2 f$ the gradient and the Hessian matrix of f, respectively.

Theorem 1.1 (First-Order Necessary Conditions). *If x^* is a local minimizer and f is continuously differentiable in an open neighbourhood of x^*, then $\nabla f(x^*) = 0$.*

Theorem 1.2 (Second-Order Necessary Conditions). *If x^* is a local minimizer and $\nabla^2 f$ exists and is continuous in an open neighbourhood of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.*

Theorem 1.3 (Second-Order Sufficient Conditions)**.** *Suppose that $\nabla^2 f$ is continuous in an open neighbourhood of x^*, and that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then, x^* is a strict local minimizer of f.*

All the points satisfying $\nabla f(x^*) = 0$ are called *first order critical points* or *stationary points*. Once a solution x^* of (1.2) is found, the *residual* at the solution is defined as the value of the function at the local minimizer: $f(x^*) = \frac{1}{2}\|R(x^*)\|^2$. If a point for which $f(x^*) = 0$ exists, (1.2) is said a *zero residual* problem. In this case x^* is also a solution to the nonlinear system $R(x) = 0$. If such a point does not exist, then $f(x) > 0$ for all $x \in \mathbb{R}^n$ and (1.2) is said to be a *nonzero residual* problem.

We consider the case in which $m \geq n$, that is called the overdetermined case. This is the most common case, as it arises in many data fitting applications.

Example 1.4 (Data fitting applications). *In data fitting applications one wants to approximate an unknown function $\psi(z)$, having at disposal m empirical data y_i, $i = 1,\dots,m$. The data are empirical measurements of the unknown function for different values of the parameter z. Taking into account that the data are usually affected by errors, the set of data can be thought as a set of couples $\{z_i, y_i\}$ where $y_i \simeq \psi(z_i)$. For example f can be the temperature depending on the time of the day z. Then, the unknown function ψ is approximated through a model $m(x,z)$, depending on some parameters $x \in \mathbb{R}^n$. The best values for them should be found to get a model that predicts the function in the best way. This is done employing the available data. For each available data value a residual R_i is defined as the difference between the model prediction and the data value itself:*

$$R_i(x) = m(x, z_i) - y_i, \qquad i = 1,\dots,m.$$

and the best value for the parameters x can be found solving a least squares problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|R(x)\|^2$$

for $R(x) = [R_1(x),\dots,R_m(x)]^T$. If at least one of the parameters x appears nonlinearly in the model we obtain a nonlinear least squares problem. This is the case for example when data are fitted to a Gaussian curve:

$$m(x,z) = ae^{\frac{(z-b)^2}{2c^2}}.$$

In this example $x = [a,b,c]^T$ are the unknown parameters to be found.

A least squares problem is a really special case of problem (1.1), as it has a strong structure. For example we can derive special expressions for the derivatives of f, assuming it to be twice continuously differentiable. Indeed, if we denote with $J(x) \in \mathbb{R}^{m \times n}$ the Jacobian matrix of $R(x)$, the gradient of f can be expressed as [27, Chapter 10]

$$\nabla f(x) = J(x)^T R(x). \tag{1.3}$$

Similarly the Hessian is given by

$$\nabla^2 f(x) = J(x)^T J(x) + S(x) = J(x)^T J(x) + \sum_{i=1}^{m} R_i(x)\nabla^2 R_i(x). \tag{1.4}$$

Notice that term $S(x)$ contains the second derivatives $\nabla^2 R_i$ of R. Its norm depends both on the nonlinear residual $R(x)$ and on the magnitude of such derivatives.

Least squares problems are a special case of (1.1), and they could therefore be solved by general optimization methods. However other methods have been specially devised that are more efficient, as they exploits the special structure of the problem. In many cases they achieve better than linear convergence, sometimes even quadratic convergence, even though they do not need implementation of second derivatives of R [27, Chapter 10]. Well-known methods specially designed for the solution of such problems are Gauss-Newton and Levenberg-Marquardt methods, that will be introduced in Chapter 2.

1.1 Linear least squares problems

A special case of (1.2) is given by linear least squares, that are obtained when function R is linear: $R(x) = Ax - b$, for $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Then the following problem is considered:

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|Ax - b\|^2. \tag{1.5}$$

This represents an important occurrence, as the solution of nonlinear problems reduces to the solution of a sequence of linear subproblems. In this case

$$\nabla f(x) = A^T(Ax - b) \qquad \text{and} \qquad \nabla^2 f(x) = A^T A. \tag{1.6}$$

Notice that the second term in (1.4) does not appear in this case, as the second derivatives of R are zero. As f in (1.5) is convex, the points for which $\nabla f(x) = 0$ are all global minimizers. They can then be found from (1.6) as the solutions of the following linear system of equations, that are called *normal equations*:

$$A^T A x = A^T b. \tag{1.7}$$

In case A has full rank, $A^T A$ is positive definite and f is strictly convex. In this case (1.7) has a unique solution, which is the global minimizer for (1.5). If A is rank deficient, matrix $A^T A$ is singular, but system (1.7) will still have a solution because of the equivalence with (1.5). In this case an infinite number of solutions exists, while the so-called *minimum norm solution*, i.e. the solution of minimal norm, will still be unique. This can be characterized by means of the pseudoinverse of matrix A, that we introduce in the next section.

1.1.1 Singular value decomposition and least squares problems

For a real matrix many different decompositions can be defined. Among them, one that is particularly useful in various contexts is the singular value decomposition (SVD). It can be defined thanks to the results stated in the following theorem:

Theorem 1.5 (Theorem 2.5.2 [44])**.** *Given a real matrix $A \in \mathbb{R}^{m\times n}$, it exist orthogonal matrices*

$$U = (u_1,\dots,u_m) \in \mathbb{R}^{m\times m}, \qquad V = (v_1,\dots,v_n) \in \mathbb{R}^{n\times n}$$

such that

$$U^T AV = \mathtt{diag}([\varsigma_1,\dots,\varsigma_\nu]) \in \mathbb{R}^{m\times n}, \qquad \nu = \min\{m,n\},$$

with $\varsigma_1 \geq \cdots \geq \varsigma_\nu \geq 0$.

The values $\varsigma_1,\dots,\varsigma_\nu$ are called the *singular values* of matrix A and the SVD of matrix A is defined as

$$A = U\Sigma V^T, \qquad \Sigma = \mathtt{diag}([\varsigma_1,\dots,\varsigma_\nu]) \in \mathbb{R}^{m\times n}. \tag{1.8}$$

This decomposition reveals a great deal about the matrix. For example if $\varsigma_1 \geq \cdots \geq \varsigma_\ell > \varsigma_{\ell+1} = \cdots = \varsigma_\nu = 0$, then the rank of matrix A is ℓ,

$$\mathcal{K}er(A) = span\{v_{\ell+1},\dots,v_n\}, \quad \mathcal{R}(A) = span\{u_1,\dots,u_\ell\},$$

where $\mathcal{K}er(A)$ and $\mathcal{R}(A)$ are respectively the null space and the range space of A. The SVD expansion can be written as [44, §2.5]

$$A = \sum_{i=1}^{\ell} \varsigma_i u_i v_i^T. \tag{1.9}$$

Also it holds [44, §2.5]

$$\|A\|_2 = \varsigma_1, \qquad \min_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \varsigma_n \; (m \geq n).$$

The SVD decomposition can be used also to define the Moore-Penrose pseudoinverse of a matrix, which generalizes the concept of inverse of a matrix to the rank deficient case and to rectangular matrices [44, §5.5.4]. Given a matrix $A \in \mathbb{R}^{m\times n}$, whose SVD in given by (1.8), the Moore-Penrose pseudoinverse $A^+ \in \mathbb{R}^{n\times m}$ is

$$A^+ = V\Sigma^+ U^T, \qquad \Sigma^+ = \mathtt{diag}\left(\left[\frac{1}{\varsigma_1},\dots,\frac{1}{\varsigma_\ell},0,\dots,0\right]\right) \in \mathbb{R}^{n\times m}, \qquad \ell = rank(A).$$

It is the unique minimal Frobenius norm solution to the problem [44, §5.5.4]

$$\min_{X\in\mathbb{R}^{n\times m}} \|AX - I_m\|_F.$$

If $\texttt{rank}(A) = n$, then $A^+ = (A^TA)^{-1}A^T$ and $A^+A = I$, while if $m = n = \texttt{rank}(A)$, then $A^+ = A^{-1}$.

The pseudoinverse can be equivalently defined as the unique matrix X that satisfies the four following conditions:

$$\begin{aligned} &1)\, AXA = A \qquad && 3)\, (AX)^T = AX \\ &2)\, XAX = X \qquad && 4)\, (XA)^T = XA. \end{aligned}$$

These conditions amount to the requirement that AA^+ and A^+A be orthogonal projections onto $\mathscr{R}(A)$ and $\mathscr{R}(A^T)$, respectively. Indeed, $AA^+ = U_1U_1^T$ where $U_1 = (u_1, \dots, u_\ell) \in \mathbb{R}^{m\times\ell}$ and $A^+A = V_1V_1^T$, where $V_1 = (v_1, \dots, v_\ell) \in \mathbb{R}^{n\times\ell}$ [44, §5.5.4].

The Moore-Penrose pseudoinverse can be used to characterize solutions to linear least squares problems. Indeed, the minimum norm solution x^* to (1.5) and the optimal residual can be expressed as [44, §5.5.3].

$$x^* = A^+b = \sum_{i=1}^{\ell} \frac{u_i^Tb}{\varsigma_i} v_i, \qquad \frac{1}{2}\|Ax^* - b\|_2^2 = \frac{1}{2}\|(I - AA^+)b\|_2^2 = \frac{1}{2}\sum_{i=\ell+1}^{m} (u_i^Tb)^2. \quad (1.10)$$

1.1.2 Solution techniques

In this section we briefly give some indications to solution techniques for linear least squares problems. For a more detailed description we refer to [44, §5]. To solve (1.5) in case of small dimensional problems, one could form and solve system (1.7) through the Cholesky factorization of A^TA. The main advantage of this approach is speed. It can be convenient also when $m \gg n$ and it is practical to store A^TA but not A. However, this will not yield an accurate solution when A is ill-conditioned, considering that the condition number of A^TA is the square of the condition number of A. Other solution methods are possible, for which the error in the computed solution is proportional to the condition number of A rather then to that of A^TA, like the QR-based approaches [85, §10.2]. In case of large scale problems, specific variants of the Conjugate Gradient (CG) method, like CGLS, are used. They avoid to form matrix A^TA and just rely on multiplications by A and A^T for which it is possible to devise also matrix-free implementations [60, 87]. These are implementations for which computing and storing A is not needed, provided that the operators computing the products A and A^T times a vector are available.

CHAPTER

2 Numerical methods

In this section we introduce the methods we are going to consider in the thesis. As least squares problems are a special case of unconstrained minimization, we will first focus on solution methods for this latter class. Then, approaches arising from them, specially designed for least squares, will be presented. Particularly, we will briefly introduce Newton's method and then Trust-Region schemes, for the globalization of approaches for unconstrained minimization. Finally, we present Gauss-Newton and Levenberg-Marquardt methods.

These are all iterative procedures, that starting from an initial guess x_0 build a sequence $\{x_k\}$ approximating the sought solution. As the aim is to minimize the function, at each iteration k we look for a direction along which it is possible to achieve a decrease in the function. Namely, given the current approximation x_k, the new approximation is defined as $x_{k+1} = x_k + p_k$ where p_k is called the *step*, that should be such that $f(x_k + p_k) < f(x_k)$. At each iteration the objective function is approximated by a model easy to minimize and the step is defined as the model minimizer. Given an iterative method designed to solve problem (1.1), we expect it to produce a sequence $\{x_k\}$ converging to a stationary point of f. Then, we expect the sequence to be such that $\lim_{k\to\infty} \nabla f(x_k) = 0$. If this is obtained regardless of the starting point x_0 the method is said to be *globally convergent*. Usually the second order sufficient conditions are not checked at the solution, then one cannot exclude convergence to a saddle point. However convergence to a maximum can be excluded, as the function is decreased at each step.

2.1 Newton's method

Newton's method is a fast approach designed to solve unconstrained problems (1.1) for a nonlinear twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ [27, Chapter 5]. Given the current iterate x_k, the new solution approximation is built as $x_{k+1} = x_k + p_k^N$, where p_k^N is the Newton step, that is computed as follows.

At each iteration information on the objective function is used to build a quadratic model for it. From Taylor's theorem for every x and p it exists $z \in (x, x+p)$ such that:

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(z) p. \tag{2.1}$$

This suggests to build the model for function f at iteration k as

$$m_k^N(x_k + p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p, \tag{2.2}$$

so that m_k^N and f are in agreement up to second order at the current iterate x_k. The step is then sought to minimize the quadratic model and, whenever $\nabla^2 f(x_k)$ is positive definite, it can be found solving the Newton's equations:

$$\nabla^2 f(x_k) p = -\nabla f(x_k),$$

that is equivalent to looking for a point such that $\nabla m_k^N(x_k + p) = 0$. The method is well-defined in a neighbourhood of a minimum that satisfies second order sufficient conditions. In this case the Hessian is positive definite, then is invertible and the step is well-defined. Newton's method indeed, is intended primarily as a local method to be used when x_k is close enough to a minimizer, even if useful adaptations has been devised to handle also the case of non positive definite Hessians [27, §5.5]. It can be proved that the method is fast locally convergent, as stated in the following theorem.

Theorem 2.1 (Theorem 3.5 [85])**.** *Suppose that f is twice continuously differentiable and $\nabla^2 f(x)$ is Lipschitz continuous in a neighbourhood of a solution x^* at which the sufficient conditions are satisfied. Then*

- *if the starting point is sufficiently close to x^* , the sequence of iterates converges to x^*;*
- *the rate of convergence is quadratic, i.e. $\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2$ for a suitable $c > 0$.*
- *the sequence of gradient norms $\|\nabla f(x_k)\|$ converges quadratically to zero.*

However, if a good starting point is not available the method may not be convergent at all. In next section we introduce a possible strategy to make it globally convergent.

2.2 Trust-Region methods

Let us consider unconstrained problems of the form (1.1) for a nonlinear twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ and let x^* be a stationary point for (1.1). Trust-Region methods are globally convergent iterative methods intended to their solution. For what reported in this section we refer to [22]. At each iteration, given the current approximation x_k to x^*, a new iterate is built that lays in a neighbourhood $\mathscr{B}_k$ of x_k:

$$\mathscr{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k\}, \tag{2.3}$$

for $\|\cdot\|_k$ a norm that may be iteration dependent and Δ_k a positive value called the *Trust-Region radius*. A sequence $\{x_k\}$ converging to a first order critical point is generated. In Algorithm 2.1 we sketch the k-th iteration of the method.

Algorithm 2.1 k-th iteration of basic Trust-Region algorithm for problem (1.1)

Initialization: Given x_k, $\mathscr{B}_k$, $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_1 \leq \gamma_2 < 1$.
Compute $f(x_k)$.

1. **Model definition:** Define a model $m_k^{TR}(x_k + p)$ in $\mathscr{B}_k$.
2. **Step calculation:** Compute a step p_k^{TR} that "sufficiently reduces" the model and such that $x_k + p_k^{TR} \in \mathscr{B}_k$.
3. **Acceptance of the trial point:** Compute $f(x_k + p_k^{TR})$ and define

$$\rho_k(p_k^{TR}) = \frac{f(x_k) - f(x_k + p_k^{TR})}{m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k^{TR})}. \tag{2.4}$$

If $\rho_k(p_k^{TR}) \geq \eta_1$, then set $x_{k+1} = x_k + p_k^{TR}$, otherwise define $x_{k+1} = x_k$.

4. **Trust-Region radius update:** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k(p_k^{TR}) > \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k(p_k^{TR}) \in [\eta_1, \eta_2], \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k(p_k^{TR}) < \eta_1. \end{cases}$$

Let's now give more insight into the steps of the procedure.

At step 1 a model for the objective function f is defined. Usually, motivated by (2.1), given the current approximation x_k a quadratic model is employed, which is defined as

$$m_k^{TR}(x_k + p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p, \tag{2.5}$$

where B_k is a a symmetric matrix that approximates the Hessian of $f(x_k)$. In case $B_k = \nabla^2 f(x_k)$ the model used is Newton model (2.2) and the method is called Newton Trust-Region method, otherwise it is called a Quasi-Newton Trust-Region method.

Then, a step p_k^{TR} is computed to define the new solution approximation: $x_{k+1} = x_k + p_k^{TR}$. The model is expected to be a good approximation to the function only in a neighbourhood of the current iterate, because $m_k^{TR}(x_k + p) - f(x_k + p) = O(\|p\|^2)$, and so the approximation error is small if the step is small. Then, we restrict the search for the step to a neighbourhood $\mathscr{B}_k$ of the current solution approximation, called the Trust Region. To enforce this, the so-called Trust-Region constraint $p \in \mathscr{B}_k$ is considered.

The Trust Region is usually a ball $B_{\Delta_k}(x_k)$ of radius $\Delta_k > 0$ and centre x_k. This motivates why Δ_k is called the Trust-Region radius. In this case $\|\cdot\|_k$ in (2.3) is the Euclidean norm for all k and the Trust-Region constraint becomes $\|p\| \leq \Delta_k$. However, other choices are also possible, for example elliptical and box-shaped Trust Regions may also be used, see also [85, §4.5]. In case of an elliptical Trust Region a scaling matrix D is inserted in the constraint: $\|Dp\| \leq \Delta_k$. Matrix D is usually diagonal with positive elements, and can be iteration dependent. It is used when the function is more sensitive to the value of some components than to the others, to balance this. Namely, elements D_{ii} will be large for the components i to which the function is more sensitive. Box-shaped Trust Regions are usually used when the problem is subject also to box constraints. These are obtained employing the infinity norm, rather than the Euclidean one in the Trust-Region constraint.

Assuming to choose a spheric Trust Region, at step 2 p_k^{TR} is computed solving the following constrained subproblem, known as the *Trust-Region subproblem*:

$$\begin{aligned} &\min_{p\in\mathbb{R}^n} m_k^{TR}(x_k+p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p \\ &\text{s.t. } \|p\| \le \Delta_k. \end{aligned} \tag{2.6}$$

Notice that in (2.6) both the objective function and constraint (which can be rewritten as $p^T p \le \Delta_k^2$) are quadratic. Minimizing f reduces to considering a sequence of such subproblems.

The step can be chosen as the exact solution of (2.6). However, to obtain convergence and good practical behaviour it is enough to find an approximate solution p_k^{TR}, that "sufficiently reduces" the model. We will give more insight in Section 2.2.1 into the solution of the Trust-Region subproblem and we will give conditions to define a sufficient reduction.

Got such a step, we are sure of getting a decrease in the model, but we also want to be sure of having a decrease in the objective function. Then, at step 3 we measure the accordance between model and function through the ratio (2.4) between the *actual reduction* $f(x_k) - f(x_k + p_k^{TR})$, the reduction achieved in the function, and the *predicted reduction* $m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k^{TR})$, that is the reduction predicted by the model. Note that since the step p_k^{TR} is obtained by minimizing the model m_k^{TR} over a region that includes the step $p = 0$, the predicted reduction will always be nonnegative. Thus if $\rho_k(p_k^{TR})$ is negative, the new objective value $f(x_k + p_k^{TR})$ is greater than the current value $f(x_k)$, so the step must be rejected. On the other hand, if $\rho_k(p_k^{TR})$ is positive the function has been decreased and the step can be accepted.

The ratio (2.4) measures the agreement between the model and the function over the step: the larger the ratio, the better the agreement. Then, at step 4 (2.4) it is used also to update the Trust-Region radius for the next iteration. Three cases may occur. If the ratio is large, ideally close to 1 ($\rho_k(p_k^{TR}) > \eta_2$), there is good agreement between the function and the model, so it is safe to expand the Trust Region for the next iteration. If $\rho_k(p_k^{TR})$ is positive but not close to 1 ($\rho_k(p_k^{TR}) \in [\eta_1, \eta_2]$), it is safer not to expand the Trust Region, that is then left unchanged. If it is close to zero or negative ($\rho_k(p_k^{TR}) < \eta_1$), the model is not a good approximation to the function in the ball, and the Trust Region is shrunk. These three cases correspond to three different kinds of iterations: *very successful iterations*, *successful iterations*, and *unsuccessful iterations*. To obtain a method with good practical behaviour two cases would suffice, i.e. distinguishing between successful and unsuccessful iterations is sufficient. Indeed, this amounts to the choice $\eta_1 = \eta_2$ and $\gamma_1 = \gamma_2$ in Algorithm 2.1.

Notice also that the one at step 4, is not the only possible update for the radius. Other updates have been proposed in the literature, for example in [34, 36, 38, 39, 113] updates are considered for which the resulting Trust-Region radius converges to zero as k goes to infinity. See [111] for a review on recent developments on Trust-Region methods.

The fact that the model is minimized in a ball and the rules employed for the update of its radius, ensure global convergence properties to Trust-Region schemes. This property makes Trust-Region methods appealing, for the possibility of ensuring global convergence properties to fast locally convergent methods (like Newton's method), by using steps produced by such methods in the Trust-Region framework. Global convergence can be also achieved with an approximate solution of the Trust-Region subproblem, as we will state in Theorem 2.6.

In next section we focus on the solution of the Trust-Region subproblem.

2.2.1 Solution of the Trust-Region subproblem

Let us consider here the solution of the Trust-Region subproblem (2.6). We will consider first its exact solution and then state the conditions that an approximate solution should satisfy to guarantee convergence of the method.

2.2.1.1 Exact solution

Exact solutions are characterized by the following theorem, that states KKT conditions for a constrained problem like (2.6). For all the results reported in this section we refer to [85, §4.3], see also [22, §7].

Theorem 2.2 (Theorem 4.1, [85])**.** *A vector $p \in \mathbb{R}^n$ is a global solution of*

$$\min_{\|p\| \le \Delta} m(p) = f + g^T p + \frac{1}{2} p^T B p,$$

if and only if there is a scalar λ such that $p = p(\lambda)$ and the following conditions are satisfied:

$$(B + \lambda I) \text{ is positive semidefinite} \tag{2.7a}$$

$$(B + \lambda I) p(\lambda) = -g, \tag{2.7b}$$

$$\lambda(\Delta - \|p(\lambda)\|) = 0, \tag{2.7c}$$

$$\|p(\lambda)\| \le \Delta, \tag{2.7d}$$

$$\lambda \ge 0. \tag{2.7e}$$

Note that from (2.7b) $\lambda p(\lambda) = -Bp(\lambda) - g = -\nabla m(p(\lambda))$ that is, the step is collinear with the negative gradient of the model and then it is normal to its contours.

From (2.7a) $\lambda = 0$ is feasible only if B is positive semidefinite, otherwise it must hold $\lambda > 0$. In this latter case, from complementarity condition (2.7c), the solution of (2.7b) must lie on the Trust Region boundary. Namely $\|p(\lambda)\| = \Delta$ and the Trust

Region is said to be *active*. On the other hand, if B is positive semidefinite two different situations may happen. The solution may either lie on the Trust Region boundary and λ may be strictly positive, or the step may be strictly inside the region, $\|p(\lambda)\| < \Delta$ and then it must hold $\lambda = 0$. In this latter case the Trust Region is said to be *inactive* and $Bp = -g$. In case B is positive definite the minimum is unique, and if Newton-model is considered, the case $\lambda = 0$ corresponds to taking the full Newton step.

Theorem 2.2 suggests a strategy to find the step. If B is positive semidefinite, one can first try to solve $Bp = -g$. If the norm of the minimum norm solution does not exceed the Trust-Region radius, this is the step we were looking for. Otherwise we define $p(\lambda) = -(B+\lambda I)^{-1}g$ for λ sufficiently large that $B+\lambda I$ is positive definite and we seek a value $\lambda > 0$ such that

$$\|p(\lambda)\| = \Delta.$$

This is a scalar nonlinear equation in the variable λ that is usually called *secular equation* [22, §7.3.3]. This one-dimensional root-finding problem can be faced with Newton's method for one-dimensional problems. In the next lemma we show that a value of λ with all the desired properties exists.

Lemma 2.3 (§4.3 in [85]). *Let $B \in \mathbb{R}^{n\times n}$ be a symmetric matrix and $g \in \mathbb{R}^n$. Then it exists $\lambda \geq 0$ such that $B+\lambda I$ is positive definite and if $p(\lambda)$ is defined in (2.7b), then there exist orthogonal vectors $q_1,\dots,q_n$ such that*

$$p(\lambda) = -\sum_{j=1}^{n} \frac{q_j^T g}{\sigma_j + \lambda} q_j, \qquad \|p(\lambda)\|^2 = \sum_{j=1}^{n} \frac{(q_j^T g)^2}{(\sigma_j + \lambda)^2}, \tag{2.8}$$

where $\sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_n$ are the eigenvalues of B.

Proof. If $\lambda > -\min\{\sigma_1,\dots,\sigma_n\} = -\sigma_1$ then $B+\lambda I$ is positive definite. As B is a symmetric matrix, there exists an orthogonal matrix Q such that $B = Q\Lambda Q^T$ for $\Lambda = \texttt{diag}([\sigma_1,\dots,\sigma_n]) \in \mathbb{R}^{n\times n}$ the diagonal matrix of the eigenvalues of B. Then, as $B+\lambda I$ is positive definite

$$p(\lambda) = -Q(\Lambda+\lambda I)^{-1}Q^T g = -\sum_{j=1}^{n} \frac{q_j^T g}{\sigma_j + \lambda} q_j,$$

with q_j the j-th column of Q. By orthonormality of the q_j we also get the second relation. □

From Lemma 2.3 we can deduce the following corollary.

Corollary 2.4 (§4.3 in [85]). *Let the assumptions of Lemma 2.3 hold. If $\lambda > -\sigma_1$ then $\sigma_j + \lambda > 0$ for all $j = 1,\dots,n$ and $\|p(\lambda)\|$ is a continuous, nonincreasing function of λ on the interval $(-\sigma_1, \infty)$. Moreover*

$$\lim_{\lambda\to\infty} \|p(\lambda)\| = 0, \qquad \textit{and} \quad \lim_{\lambda\to-\sigma_1} \|p(\lambda)\| = \infty \qquad \textit{if } q_1^T g \neq 0.$$

Then we can conclude that:

- If B is positive definite and $\|B^{-1}g\| \le \Delta$ then $\lambda = 0$ and there is no need to solve the secular equation.
- If B is positive definite but $\|B^{-1}g\| > \Delta$ it exists unique λ^* in $(0,\infty)$ such that $\|p(\lambda^*)\| = \Delta$.
- If B is indefinite and $q_1^T g \neq 0$ Corollary 2.4 ensures that we can find a $\lambda^* \in (-\sigma_1, \infty)$ such that $\|p(\lambda^*)\| = \Delta$.

The case $q_1^T g = 0$ is known as the hard case [22, §7.3], [85, §4.3]. We do not consider it here, as in this thesis we will always deal with positive semidefinite B, then this case is not of interest. We then describe here how to numerically solve the secular equation in the other cases.

One can apply the root-finding Newton's method to

$$\Phi_1(\lambda) = \|p(\lambda)\| - \Delta = 0. \tag{2.9}$$

The disadvantage of this approach is that in case B is indefinite, when λ is greater but close to $-\sigma_1$, Φ_1 is a highly nonlinear function, and therefore Newton's method would be unreliable or slow [85, §4.3]. However we can reformulate (2.9) as

$$\Phi_2(\lambda) = \frac{1}{\Delta} - \frac{1}{\|p(\lambda)\|} = 0. \tag{2.10}$$

Function Φ_2 is an analytic function [22, §7.3.3] and it is nearly linear for λ slightly greater than $-\sigma_1$. Newton's method will perform well provided that it maintains $\lambda > -\sigma_1$. Then it is better to apply Newton's method to Φ_2, which generates the following sequence:

$$\lambda_{l+1} = \lambda_l + \left(\frac{\Phi_2(\lambda_l)}{\Phi_2'(\lambda_l)}\right).$$

Let's consider the Cholesky factorization $B + \lambda I = R^T R$ and set $R^T w = p$. Taking into account (2.8) and the fact that

$$\begin{aligned}\frac{d}{d\lambda}\left(\frac{1}{\|p(\lambda)\|}\right) &= \frac{d}{d\lambda}\left((\|p(\lambda)\|^2)^{-\frac{1}{2}}\right) = -\frac{1}{2}\left(\|p(\lambda)\|^2\right)^{-\frac{3}{2}}\frac{d}{d\lambda}(\|p(\lambda)\|^2)\\ &= \frac{1}{2}\left(\|p(\lambda)\|^2\right)^{-\frac{3}{2}} 2\sum_{j=1}^{n}\frac{(q_j^T g)^2}{(\sigma_j+\lambda)^3},\end{aligned}$$

we get, using also $B + \lambda I = Q(\Lambda + \lambda I)Q^T$ and (2.7b), that

$$\Phi_2'(\lambda) = -\frac{1}{\|p(\lambda)\|^3}\sum_{j=1}^{n}\frac{(q_j^T g)^2}{(\sigma_j+\lambda)^3},$$

$$\|w\|^2 = \|R^{-T}p\|^2 = p^T(B+\lambda I)^{-1}p = g^T Q(\Lambda+\lambda I)^{-3}Q^T g = \sum_{j=1}^{n}\frac{(q_j^T g)^2}{(\sigma_j+\lambda)^3}.$$

Then, we derive the practical implementation of the procedure in Algorithm 2.2.

The main cost of this procedure is given by the Cholesky factorization at step 1. However, usually it is not necessary to look for a highly accurate solution of (2.10).

Algorithm 2.2 l-th iteration of Newton's method applied to $\Phi_2(\lambda) = 0$.

```
Initialization: Given $\lambda_l$, $\Delta > 0$.
1. Factor $B + \lambda_l I = R^T R$.
2. Solve $\begin{cases} R^T R p_l = -g, \\ R^T w_l = p_l. \end{cases}$
3. Set $\lambda_{l+1} = \lambda_l + \left(\frac{\|p_l\|}{\|w_l\|}\right)^2 \left(\frac{\|p_l\| - \Delta}{\Delta}\right)$.
```

2.2.1.2 Approximate solution

As we have previously stated, it is sufficient to solve (2.6) approximately to get global convergence of the method. An approximate solution of (2.6) is a step that lies within the Trust Region and gives a "sufficient reduction" in the model. This reduction can be quantified in the following way. As the model locally decreases at the fastest rate in the direction of the negative gradient $-\nabla f(x_k)$, that is the steepest descend direction, it makes sense to analyze the decrease got in this direction, i.e the decrease got along the Cauchy arc [22, §6.3]:

$$\{x \,|\, x = x_k - t\nabla f(x_k),\ t \geq 0,\ x \in B_{\Delta_k}(x_k)\}.$$

It is possible to minimize the model exactly on the Cauchy arc [22, §6.3]. The resulting unique point is called the *Cauchy point* that we denote by x_k^C. Then, it can be defined as

$$x_k^C = x_k - t_k^C \nabla f(x_k) = \underset{\substack{t \geq 0 \\ x_k - t\nabla f(x_k) \in B_{\Delta_k}(x_k)}}{\arg\min} \quad m_k^{TR}(-t\nabla f(x_k)) \tag{2.11}$$

where $p_k^C = -t_k^C \nabla f(x_k)$ is called the *Cauchy step*. It holds [22, Theorem 6.3.1]

$$t_k^C = \begin{cases} \frac{\Delta_k}{\|\nabla f(x_k)\|} & \text{if } \nabla f(x_k)^T B_k \nabla f(x_k) \leq 0, \\ \min\left\{\frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)}, \frac{\Delta_k}{\|f(x_k)\|}\right\} & \text{if } \nabla f(x_k)^T B_k \nabla f(x_k) > 0. \end{cases}$$

In the following theorem we show a lower bound for the decrease obtained in the model employing this point.

Theorem 2.5 (Theorem 6.3.1 [22])**.** *If the Cauchy point is defined as in* (2.11)*, it holds*

$$m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k^C) \geq \frac{1}{2}\|\nabla f(x_k)\| \min\left[\Delta_k, \frac{\|\nabla f(x_k)\|}{\|B_k\|}\right]. \tag{2.12}$$

Condition (2.12) is called *sufficient Cauchy decrease* because, as we show in the following theorem, a fraction of that decrease is sufficient to obtain global convergence of the method.

Theorem 2.6 (Theorem 4.6 [84])**.** *Suppose that f is bounded below on the level set $\mathscr{L} = \{x \in \mathbb{R}^n$ s.t. $f(x) \leq f(x_0)\}$ and Lipschitz continuously differentiable in a neighbourhood of $\mathscr{L}$. Suppose further that $\|B_k\|$ is bounded above for all k and that the* *approximate solutions $p = p_k^{TR}$ computed at step 2 of Algorithm 2.1 satisfy for all k*

$$m_k^{TR}(x_k) - m_k^{TR}(x_k + p) \geq \theta \|\nabla f(x_k)\| \min \left[\Delta_k, \frac{\|\nabla f(x_k)\|}{\|B_k\|} \right] \tag{2.13}$$

for some positive θ. Then we have that the sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies

$$\lim_{k\to\infty} \nabla f(x_k) = 0. \tag{2.14}$$

We remark that condition (2.13) is satisfied for all p_k such that $\|p_k\| \leq \Delta_k$ and that achieve at least some fixed fraction θ_2 of the reduction achieved by the Cauchy step, i.e. if

$$m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k) \geq \theta_2(m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k^C)).$$

In this case (2.13) is satisfied with $\theta = \theta_2/2$. In particular it holds for the exact solution p^* of (2.6) with $\theta = \frac{1}{2}$ [85, Theorem 4.4].

Then, to make it clear what is looked for at step 2 of Algorithm 2.1, we can give the following definition, arising from all the previously stated results.

Definition 2.7. *We say that p sufficiently reduces model m_k^{TR} if it provides the sufficient Cauchy decrease, i.e. if it exists $\theta > 0$ such that (2.13) holds.*

Remark 2.8. *A special case of Trust-Region approaches is that of Trust-Region Newton schemes. In such approaches $B_k = \nabla^2 f(x_k)$, i.e. the exact Hessian is used in the model m_k^{TR}, when x_k is close to a solution that satisfies second-order conditions. These approaches are usually designed in such a way that asymptotically the Trust-Region constraint results to be inactive, so that the Trust-Region bound eventually does not interfere with the convergence and fast local convergence of Newton's method is recovered. Indeed, it is possible to prove that for any algorithm of the form of Algorithm 2.1 the Trust-Region constraint eventually becomes inactive. This holds provided that the algorithm produces steps that asymptotically become closer and closer to the pure Newton step, whenever the true Newton step is well inside the Trust Region. This is the case also when the subproblem is solved inexactly, if the step satisfies the Cauchy decrease according to Definition 2.7. The result is stated in the following Theorem:*

Theorem 2.9 (Theorem 4.9 [84])**.** *Let f be twice Lipschitz continuously differentiable in a neighbourhood of a point x^* at which second order sufficient conditions are satisfied. Suppose the sequence $\{x_k\}$ converges to x^* and that for all k sufficiently large, the Trust-Region algorithm based on the choice $B_k = \nabla^2 f(x_k)$ in the model m_k^{TR}, chooses steps p_k that satisfy the Cauchy-decrease (2.13) and are asymptotically similar to Newton steps p_k^N whenever $\|p_k^N\| \leq \frac{1}{2}\Delta_k$, that is, $\|p_k - p_k^N\| = O(\|p_k^N\|)$. Then the Trust-Region constraint becomes inactive for all k sufficiently large and the sequence $\{x_k\}$ converges superlinearly to x^*.*

In the next sections we turn to consider methods specially designed for least squares problems: Gauss-Newton and Levenberg-Marquardt methods. They are designed as suitable modifications of the methods just presented.

2.3 Gauss-Newton method

Gauss-Newton method is a modification of Newton's method, specially designed to handle nonlinear least squares problems.

Like Newton's approach, Gauss-Newton method builds a model for function f at each iteration, but taking into account the special form of the objective function. Then, recalling (1.3) and (1.4), model (2.2) for f as in (1.2) becomes

$$\frac{1}{2}\|R(x_k)\|^2 + (J(x_k)^T R(x_k))^T p + \frac{1}{2} p^T (J(x_k)^T J(x_k) + S(x_k)) p.$$

Gauss-Newton model is obtained approximating the Hessian $J(x_k)^T J(x_k) + S(x_k) \simeq B_k = J(x_k)^T J(x_k)$, dropping the term $S(x_k)$ which contains the second derivatives of $R(x)$ at x_k:

$$m_k^{GN}(x_k + p) = \frac{1}{2}\|R(x_k)\|^2 + (J(x_k)^T R(x_k))^T p + \frac{1}{2} p^T J(x_k)^T J(x_k) p.$$

This approximation is convenient because usually the portion $J(x_k)^T J(x_k)$ of the Hessian will already be available since $J(x_k)$ must be calculated to get $\nabla f(x_k)$ [85, §10.3]. We can notice also that

$$m_k^{GN}(x_k + p) = \frac{1}{2}\|J(x_k)p + R(x_k)\|^2, \tag{2.15}$$

i.e. f is approximated by the squared norm of an affine model of R. Then, using just an affine model for R we get a second order model for f, with approximated Hessian. At each iteration the step p_k^{GN} will be computed minimizing m_k^{GN}, that is solving a linear least squares problem, as explained in Section 1.1. The normal equations are, cf. (1.7),

$$J(x_k)^T J(x_k) p = J(x_k)^T R(x_k). \tag{2.16}$$

The success of the method clearly depends on the quality of the Hessian approximation. It depends in particular on whether the term that is discarded is a large part of the Hessian or not. Anyway, the term $J(x_k)^T J(x_k)$ is often more important then the other one, especially close to a solution x^*. This happens either because the residuals $R_i(x^*)$ are close to affine near the solution x^*, and then $\nabla^2 R_i(x^*)$ is small, or because of small residuals, i.e. $R_i(x^*)$ itself is small [85, Chapter 10]. $\|S(x^*)\|$ indeed, can be viewed as an absolute combined measure of the nonlinearity and residual size of the problem [27, Chapter 10]. Convergence of the method actually depends on the relation between $\|S(x^*)\|$ and $\|J(x^*)^T J(x^*)\|$. It can be proved that if $\|S(x^*)\|$ is small relative to $\|J(x^*)^T J(x^*)\|$, the Gauss-Newton method is locally q-linearly convergent, but if $\|S(x^*)\|$ is too large it may be not be convergent at all [27, Chapter 10].

Assuming Lipschitz continuity of the Hessian, defining $x_{k+1} = x_k + p_k^{GN}$, it can be shown that

$$\|x_{k+1} - x^*\| \simeq \|(J(x^*)^T J(x^*))^{-1} S(x^*)\| \, \|x_k - x^*\| + O(\|x_k - x^*\|^2),$$

[85, §10.3]. Hence if $\|(J(x^*)^T J(x^*))^{-1} S(x^*)\| << 1$ we can expect rapid local convergence and even quadratic convergence if $S(x^*) = 0$, i.e. for example for zero residual problems.

We report here more rigorous convergence results, stated in Theorem 10.2.1 and Corollary 10.2.2 in [27].

Theorem 2.10 (Theorem 10.2.1 [27])**.** *Let $R : \mathbb{R}^n \to \mathbb{R}^m$, f be defined in (1.2) twice continuously differentiable in an open convex set $D \subseteq \mathbb{R}^n$. Assume that J is Lipschitz continuous in D with constant γ and $\|J(x)\| \leq \alpha$ for all $x \in D$. Assume that it exists $x^* \in D$ such that $\nabla f(x^*) = 0$ and let $\tilde{\lambda}$ be the smallest eigenvalue of $J(x^*)^T J(x^*)$. Assume also that it exists $\sigma \geq 0$ such that*

$$\|(J(x) - J(x^*))^T R(x^*)\| \leq \sigma \|x - x^*\|, \quad \textit{for all} \quad x \in D.$$

If $\sigma < \tilde{\lambda}$ (and so $J(x^*)$ has full rank), for any $c \in (1, \tilde{\lambda}/\sigma)$, there exists $\epsilon > 0$ such that for all $x_0 \in B_\epsilon(x^*)$ the sequence generated by the Gauss-Newton method is well defined, converges to x^* and it holds

$$\|x_{k+1} - x^*\| \leq \frac{c\sigma}{\tilde{\lambda}} \|x_k - x^*\| + \frac{c\alpha\gamma}{2\tilde{\lambda}} \|x_k - x^*\|^2,$$
$$\|x_{k+1} - x^*\| \leq \frac{c\sigma + \tilde{\lambda}}{2\tilde{\lambda}} \|x_k - x^*\| < \|x_k - x^*\|.$$

Theorem 2.11 (Corollary 10.2.2 [27])**.** *Let the assumptions of Theorem 2.10 be satisfied. If $R(x^*) = 0$ there exists $\epsilon > 0$ such that for all $x_0 \in B_\epsilon(x^*)$ the sequence generated by the Gauss-Newton method is well defined and converges q-quadratically to x^*.*

In general Gauss-Newton is not a globally convergent approach but it can be made so by coupling it with line-search or Trust-Region approaches. In this thesis we will focus just on this latter class of globalization strategies, which have been introduced in Section 2.2. In the next section we report on the Levenberg-Marquardt method, that can be viewed as the globally convergent extension of Gauss-Newton method got by coupling it with Trust-Region approaches.

2.4 Levenberg-Marquardt method

Levenberg-Marquardt method was first introduced by Levenberg [76] and Marquardt [78]. It is an algorithm for the solution of nonlinear least squares problems (1.2). It can be regarded as an improvement over the Gauss-Newton method, aimed at solving its two main weaknesses: it is not necessarily a globally convergent approach, which is crucial for the practical use when a good starting point is not known, and it is not well defined when the Jacobian is rank-deficient.

Both weaknesses can be overcome adding a regularization term in the Gauss-Newton model (2.15). Levenberg-Marquardt method is still an iterative method and at each iteration, given a regularization parameter $\lambda_k > 0$, the step is found solving the following minimization problem:

$$\min_{p\in\mathbb{R}^n} m_k^{LM}(x_k+p) = \frac{1}{2}\|J(x_k)p + R(x_k)\|^2 + \frac{\lambda_k}{2}\|p\|^2. \tag{2.17}$$

This can be stated also as a linear least squares problem:

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}\left\| \begin{bmatrix} J(x_k) \\ \sqrt{\lambda_k} I \end{bmatrix} p + \begin{bmatrix} R(x_k) \\ 0 \end{bmatrix} \right\|^2,$$

whose normal equations are given by, cf. (1.7),

$$(J(x_k)^T J(x_k) + \lambda_k I)p = -J(x_k)^T R(x_k). \tag{2.18}$$

Remark 2.12. *Notice the strict connection between the Levenberg-Marquardt step and the Trust-Region step defined in* (2.7b)*. The first one is obtained choosing* $B = J(x_k)^T J(x_k)$ *and* $g = J(x_k)^T R(x_k)$ *in* (2.7b)*. In Levenberg-Marquardt methods parameter* λ_k *is given a-priori, while in Trust-Region approaches it comes from relation* (2.7c)*, namely it corresponds to the Lagrange multiplier in the KKT conditions* (2.7)*. In this case* B *is always positive semidefinite and* $g \in \mathscr{R}(B)$*.*

Due to this strong relation, Lemma 2.3 also holds for the Levenberg-Marquardt step. It shows how the length of the step is influenced by the choice of the free parameter λ_k. Particularly the norm of the step is decreasing as λ increases.

The addition of an arbitrary strictly positive regularizing term makes the method well defined even in case of rank-deficient Jacobian and in case of ill-conditioned matrix the conditioning of the system is improved. However, a wiser choice of the parameter can provide additional properties to the method. For example, the method can get regularizing properties and become suitable to the solution of ill-posed inverse problems. In this context it is more commonly known as non-stationary iterated Tikhonov method, that we will describe in Section 2.4.4.

The approach can also be made globally convergent [80, 86, 90]. In Section 2.4.3 it is shown how this can be gained exploiting the strong connection between Levenberg-Marquardt and Trust-Region methods, that we have highlighted in Remark 2.12. Indeed, the approach can be implemented through a Trust-Region strategy and the free parameters λ_k are indirectly selected from the Trust-Region radius choice. An alternative, originally proposed in [78, 86], is to update the parameters directly, with an update that mimics the one of the Trust-Region radius. The advantage is that the linear system (2.18) is easier to solve than the Trust-Region subproblem. Based on the reduction of the objective function f, λ_k is increased in case of unsuccessful iterations and decreased in case of successful ones.

Remark 2.13. *Note that the Trust-Region radius* Δ_k *and the Levenberg-Marquardt parameters* λ_k *are 'inversely related'. Namely, a reduction of* Δ_k *means a reduction in the norm of the step and conversely an increase in* Δ_k *allows bigger steps. On* *the contrary, if λ_k is increased the norm of the step is decreased, see (2.8). Then, the update of λ_k is 'reversed' compared to the update of the Trust-Region radius, i.e. in case of successful steps Δ_k is increased and λ_k is reduced and vice-versa for the unsuccessful case.*

Often, the update of the parameter and the step acceptance are still based on the ratio $\rho_k(p_k^{LM}) = \frac{f(x_k)-f(x_k+p_k^{LM})}{m_k^{LM}(x_k)-m_k^{LM}(x_k+p_k^{LM})}$. Given $0 < \eta_0 < \eta_1 < \eta_2 < 1$ and $0 < \gamma_1 < 1 < \gamma_0$, one set [86, 103, 114]

$$x_{k+1} = \begin{cases} x_k + p_k^{LM} & \text{if } \rho_k(p_k^{LM}) \geq \eta_0, \\ x_k & \text{if } \rho_k(p_k^{LM}) < \eta_0, \end{cases} \tag{2.19}$$

and

$$\lambda_{k+1} = \begin{cases} \gamma_1 \lambda_k & \text{if } \rho_k(p_k^{LM}) > \eta_2, \\ \lambda_k & \text{if } \rho_k(p_k^{LM}) \in [\eta_1, \eta_2], \\ \gamma_0 \lambda_k & \text{if } \rho_k(p_k^{LM}) < \eta_1. \end{cases} \tag{2.20}$$

However other updates are possible. For example many papers in the literature are concerned with the study of parameters update to get fast local convergence under mild assumptions. The most common choice for zero residual problems is to relate λ_k to the magnitude of the nonlinear residual. Possible choices are $\lambda_k = \|R(x_k)\|^2$ [110, 68], $\lambda_k = \|R(x_k)\|^\beta$ for $\beta \in (0,2]$ [8, 40], $\lambda_k = \mu_k \|R(x_k)\|^\beta$ with $0 \leq \beta \leq 1$ [33, 102] with μ_k updated adaptively as

$$\mu_{k+1} = \begin{cases} \max\{\gamma_1 \mu_k, m\} & \text{if } \rho_k(p_k^{LM}) > \eta_2, \\ \mu_k & \text{if } \rho_k(p_k^{LM}) \in [\eta_1, \eta_2], \\ \gamma_0 \mu_k & \text{if } \rho_k(p_k^{LM}) < \eta_1, \end{cases} \tag{2.21}$$

where m is a small positive constant to prevent the parameter from being too small. In [37] the parameter is chosen as a combination of $\|R(x_k)\|$ and $\|J(x_k)^T R(x_k)\|$.

2.4.1 Inexact Levenberg-Marquardt method

A Levenberg-Marquardt method is said to be inexact when the subproblem (2.17) is not solved exactly. Indeed, to get a convergent method it is not necessary to solve the subproblem with high accuracy and it is sufficient to find an approximate solution. The study of the inexactness of the method investigates the level of inexactness that can be allowed in the subproblem solution without affecting either the global convergence or a given rate of local convergence.

Regarding the global convergence, as for Trust-Region methods it is possible to define an approximate solution by means of the sufficient Cauchy decrease. In this case the Cauchy point is defined as [11]

$$x_k^c = x_k - t_k^c \nabla f(x_k) = \arg\min_{t \geq 0} m_k^{LM}(-t \nabla f(x_k)) \tag{2.22}$$

and $p_k^c = -t_k^c \nabla f(x_k)$ is again called the Cauchy step. It is possible to see that

$$t_k^c = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T (B_k + \lambda_k I) \nabla f(x_k)}.$$

We can then give the following definition:

Definition 2.14. *We say that a step p approximately minimizes model m_k^{LM} if it achieves the sufficient Cauchy decrease, i.e. if it provides at least as much reduction in m_k^{LM} as that achieved by the Cauchy point (2.22):*

$$m_k^{LM}(x_k) - m_k^{LM}(x_k + p) \geq \theta \frac{\|J(x_k)^T R(x_k)\|^2}{\|J(x_k)\|^2 + \lambda_k}, \qquad \theta > 0. \tag{2.23}$$

For the Cauchy step p_k^c (2.23) holds with $\theta = \frac{1}{2}$. To find a step achieving the Cauchy decrease it is sufficient to solve the normal equations (2.18) approximately, i.e. to compute a step p such that

$$(J(x_k)^T J(x_k) + \lambda_k I)p = -J(x_k)^T R(x_k) + r_k, \tag{2.24}$$

for a residual vector r_k such that $\|r_k\| > 0$. The resulting step is usually called an inexact step. It can be computed applying an iterative method to the normal equations without reaching high accuracy, but rather stopping the iterative process as soon as the norm of the residual vector goes under a certain threshold. Indeed, if the norm of the residual vector is small enough, p achieves the Cauchy decrease, as stated in the next lemma.

Lemma 2.15 ([11], Lemma 4.1.)**.** *The inexact Levenberg-Marquardt step p_k^{LM} computed as*

$$(J(x_k)^T J(x_k) + \lambda_k I)p_k^{LM} = -J(x_k)^T R(x_k) + r_k$$

for a residual r_k satisfying

$$\|r_k\| \leq \epsilon_k \|J(x_k)^T R(x_k)\|, \qquad 0 \leq \epsilon_k \leq \sqrt{\theta_2 \frac{\lambda_k}{\|J(x_k)\|^2 + \lambda_k}}, \tag{2.25}$$

for some $\theta_2 \in (0,1)$, achieves the Cauchy decrease (2.23), with $\theta = (1-\theta_2) \in (0,1)$.

Proof. We can rewrite the predicted reduction as

$$\begin{aligned} m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM}) &= -(J(x_k)^T R(x_k))^T p_k^{LM} - \frac{1}{2}(-J(x_k)^T R(x_k) + r_k)^T p_k^{LM} \\ &= -\frac{1}{2}(J(x_k)^T R(x_k) + r_k)^T p_k^{LM} \\ &= \frac{1}{2}(J(x_k)^T R(x_k) + r_k)^T (J(x_k)^T J(x_k) + \lambda_k I)^{-1} (J(x_k)^T R(x_k) - r_k). \end{aligned}$$

Since $J(x_k)^T J(x_k)$ is positive semidefinite,

$$r_k^T (J(x_k)^T J(x_k) + \lambda_k I)^{-1} r_k \leq \frac{\|r_k\|^2}{\lambda_k} \leq \frac{\epsilon^2 \|J(x_k)^T R(x_k)\|^2}{\lambda_k}$$

and

$$(J(x_k)^T R(x_k))^T (J(x_k)^T J(x_k) + \lambda_k I)^{-1} J(x_k)^T R(x_k) \geq \frac{\|J(x_k)^T R(x_k)\|^2}{\|J(x_k)\|^2 + \lambda_k}$$

Then we conclude that

$$\begin{aligned} m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM}) &\geq \left(\frac{1}{\|J(x_k)\|^2 + \lambda_k} - \frac{\epsilon^2}{\lambda_k} \right) \|J(x_k)^T R(x_k)\|^2 \\ &\geq (1 - \theta_2) \frac{\|J(x_k)^T R(x_k)\|^2}{\|J(x_k)\|^2 + \lambda_k} \end{aligned}$$

□

Thanks to this result, the iterative method can be stopped as soon as (2.25) is satisfied. Usually few iterations are sufficient to reach this desired accuracy. This allows considerable computational savings, especially for large scale problems.

It can be investigated also the level of inexactness in the subproblems of Levenberg–Marquardt methods that is possible without loosing a given superlinear convergence rate. In these results the residual norm of the linear subproblems is related to the regularization term λ_k.

A well known theory for the inexactness level in Newton's method for squared nonlinear systems exists in case it is assumed that the Jacobian of R at a solution x^* has full rank [26]. This theory has been extended for Levenberg-Marquardt methods also to the rank-deficient case [24, 41, 35]. Rates of convergence have been proved for inexact methods under assumptions weaker than the nonsingularity of the Jacobian, that we will present in the next section.

2.4.2 Local convergence and complexity

In this section we consider local convergence analysis of the Levenberg-Marquardt method. We assume that the initial iterate is close to a solution x^*. If the nonlinear residual is zero, and $J(x^*)$ has full column rank, Levenberg-Marquardt method maintains fast local convergence of Gauss-Newton method [70, Theorem 3.3.4]. The local convergence of the method can however be established also under assumptions weaker than the classical assumption of the nonsingularity of the Jacobian. A widely assumed condition is the so-called *local error bound condition*.

Definition 2.16. $\|R(x)\|$ *is said to provide a local error bound for the system* $R(x) = 0$ *if there exist positive constants* b, c *such that*

$$b \, \mathtt{dist}(x, X^*) \leq \|R(x)\| \qquad \forall x \in B_c(x^*),$$

for some $x^* \in X^*$, *with* X^* *the solution set of* $R(x) = 0$ *and* $\mathtt{dist}(x, X^*) = \min_{x^* \in X^*} \|x - x^*\|$.

This definition is suitable only for zero residual problems, but both square and rectangular systems are allowed. The error bound condition weakens and generalizes the classical regularity condition. Indeed, if $J(x^*)$ is nonsingular, $\|R(x)\|$

provides a local error bound for $R(x)=0$ on some neighbourhood of x^* [69, Lemma 4.3.1], while the converse is not necessarily true [110]. Hence a local error bound condition is weaker than the nonsingularity of $J(x^*)$. Moreover, it allows also the case of nonisolated solutions. Indeed, while the nonsingularity of the Jacobian implies that solutions are isolated (i.e. if $R(x)=0$ is a square system of equations and $J(x)$ is invertible at a solution x^*, then x^* is isolated), the error bound condition might be fulfilled also at nonisolated solutions. This condition plays a decisive role in the design and local convergence analysis of other Newton-type methods as well [32, 42, 49].

Local convergence of Levenberg-Marquardt method under the error bound assumption has been considered in a large number of papers. To cite a few we mention [24, 34, 36, 40, 41, 68, 110, 112], on this topic see also [7] and references therein. The first paper in which superlinear rate for the Levenberg-Marquardt method under the error bound condition is shown is [110]. It is proved that the generated sequence $\{x_k\}$ converges to a solution and that $\mathtt{dist}(x_k, X^*)$ tends to zero quadratically. The result has been extended in [40] where it is shown that the sequence $\{x_k\}$ converges to a solution quadratically and also in [33] where both global convergence and local quadratic convergence are proved for singular square systems of nonlinear equations. Zhang in [112] provides a unifying framework for methods in [24, 40, 110], considering an update for the parameter such that $c_2\mathtt{dist}(x_k, X^*)^\beta \le \lambda_k \le c_3\mathtt{dist}(x_k, X^*)^\beta$ for constants $c_2, c_3 > 0$ and $0 < \beta \le 2$. He proves that the sequence generated by the method converges to the solution of the original equation system superlinearly and the exact order of convergence rate is $\min\{1+\beta, 2\}$ if $\|R(x)\|$ provides a local error bound for the system of nonlinear equations. The results are generalized to nonlinear equations systems with nonnegative constraints.

The above results have been then extended also to inexact methods. In this context the choice of the regularization parameter plays an important role for the goal of obtaining large inexactness levels while maintaining a given superlinear rate of convergence. A contribution on this topic is given by [41], where a robust Levenberg-Marquardt method is studied, i.e. a method based on a choice of regularization parameters of magnitude as large as possible, without decreasing the quadratic convergence rate of the exact case. It is shown that the choice $\lambda_k = \|R(x_k)\|$ enables inexactness levels of the order of $\|R(x_k)\|^2$, showing that for robust Levenberg-Marquardt method the level of inexactness allowed in the subproblems can be increased significantly, for example compared to the results in [24, 35].

The notion of error bound condition can be extended also to the case of constrained problems, simply considering in Definition 2.16 the intersection between the feasible set and the considered neighbourhood of the solution.

The constrained case has been considered in [68], where the quadratic convergence is proved. This result has been then extended also to inexact constrained Levenberg-Marquardt methods in [8, 68]. Inexact constrained methods are considered also in [32], where a new family of methods is introduced, that comprises also the Levenberg-Marquardt method in [8, 68]. Quadratic convergence rate is proved under assumptions weaker than those in [8, 68], that include neither differentiability of R nor the local uniqueness of the solutions.

The nonzero residual case is less studied. It has been considered in [63], for rank-deficient problems, where it is proved that the method converges q-linearly if the residual is small enough, under assumptions somewhat stronger than the standard assumptions for convergence of the Levenberg- Marquardt algorithm for full-rank problems, motivated by the considered applications. Recently the nonzero residual case has been studied also in [10], where the global and local convergence of a novel Levenberg-Marquardt method is studied under the assumption that $\|R(x)-R(x^*)\|$ provides a local error bound condition for (1.2).

A topic that is strictly related to the study of rates of convergence of a method is the complexity analysis. Providing a global complexity bound for a method means providing an upper bound to the number of iterations required to get an approximate solution such that $\|\nabla f(x)\| \leq \epsilon$, where ϵ is a given positive constant. For Levenberg-Marquardt methods this has been considered for example in [102, 103, 114], where it is shown that the complexity bound of the Levenberg-Marquardt method is $O(\epsilon^{-2})$, that is also the complexity bound for Newton, gradient and Trust-Region methods.

2.4.3 Trust-Region method for least squares

As least squares are a special case of unconstrained minimization, the Trust-Region scheme can also be used to solve those problems. This can be achieved choosing the model at step 1 of Algorithm 2.1 as the Gauss-Newton model (2.15) and letting the rest of the procedure unmodified, as it is described in Section 2.2. Then, in this case subproblem (2.6) takes the following form:

$$\begin{aligned}\min_{p\in\mathbb{R}^n} m_k^{TR}(x_k+p) &= \|J(x_k)p+R(x_k)\|^2\\ &= \tfrac{1}{2}\|R(x_k)\|^2+(J(x_k)^TR(x_k))^Tp+\tfrac{1}{2}p^TJ(x_k)^TJ(x_k)p \qquad (2.26)\\ \text{s.t. } \|p\| \leq \Delta_k. &\end{aligned}$$

From Theorem 2.2 the resulting step satisfies system (2.7b), with $B = J(x_k)^TJ(x_k)$, that is always positive semidefinite, and $g = J(x_k)^TR(x_k) \in \mathscr{R}(B)$. In this case (2.7b) yields exactly the normal equations (2.18) for Levenberg-Marquardt method.

Then the Trust-Region method can be thought of as a Levenberg-Marquardt method in which the free parameters are automatically set from the choice of the Trust-Region radius, as they represent the Lagrangian multipliers of the subproblem. In this case λ_k may also be zero. Then, when the Trust-Region radius is adjusted through the rules at step 4 of Algorithm 2.1, a sequence of dynamically adjusted parameters $\{\lambda_k\}$ is built. Moré in [80] gave a robust and efficient implementation of this version of the method.

In the following lemma we restate the result in Lemma 2.3 on the norm of the step $p(\lambda)$ solution of (2.26), taking into account the special form of B and g and considering also the norm of the affine model

$$M_k(p) = J(x_k)p + R(x_k). \tag{2.27}$$

We employ the singular value decomposition of the Jacobian for the proof, see Section 1.1.1.

Lemma 2.17. *[9, Lemma 4.2] Suppose $\|J(x_k)^T R(x_k)\| \neq 0$ and let $p(\lambda)$ be the minimum norm solution of (2.18) with $\lambda \geq 0$. Suppose furthermore that $J(x_k)$ is of rank ℓ and its singular value decomposition is given by $U_k \Sigma_k V_k^T$ where Σ_k is the diagonal matrix with entries $\varsigma_1, \dots, \varsigma_v$ on the diagonal, with $v = \min\{n, m\}$. Then, denoting $r = [r_1, r_2, \dots, r_m]^T = U_k^T R(x_k)$, we have that*

$$\|p(\lambda)\|^2 = \sum_{i=1}^{\ell} \frac{\varsigma_i^2 r_i^2}{(\varsigma_i^2 + \lambda)^2}, \tag{2.28}$$

$$\|R(x_k) + J(x_k)p(\lambda)\|^2 = \sum_{i=1}^{\ell} \frac{\lambda^2 r_i^2}{(\varsigma_i^2 + \lambda)^2} + \sum_{i=\ell+1}^{m} r_i^2. \tag{2.29}$$

Proof. Taking into account (2.18), the step can be defined as

$$p(\lambda) = -(J(x_k)^T J(x_k) + \lambda I)^+ J(x_k)^T R(x_k).$$

Using the singular value decomposition of $J(x_k)$ it follows

$$p(\lambda) = -V_k(\Sigma_k^T \Sigma_k + \lambda I)^+ \Sigma_k^T r. \tag{2.30}$$

As V_k has orthogonal columns,

$$\|p(\lambda)\| = \|(\Sigma_k^T \Sigma_k + \lambda I)^+ \Sigma_k^T r\| = \left\| \left[\frac{\varsigma_1}{\varsigma_1^2 + \lambda} r_1, \dots, \frac{\varsigma_\ell}{\varsigma_\ell^2 + \lambda} r_\ell, 0 \dots, 0 \right]^T \right\|$$

and (2.28) follows. Employing again the singular value decomposition of $J(x_k)$ and considering that $U_k^T U_k = I_m \in \mathbb{R}^{m \times m}$, we get

$$\begin{aligned} R(x_k) + J(x_k)p(\lambda) &= R(x_k) - U_k \Sigma_k (\Sigma_k^T \Sigma_k + \lambda I)^+ \Sigma_k r \\ &= U_k(I_m - \Sigma_k(\Sigma_k^T \Sigma_k + \lambda I)^+ \Sigma_k^T) r. \end{aligned}$$

Then,

$$\begin{aligned} \|R(x_k) + J(x_k)p(\lambda)\| &= \|(I_m - \Sigma_k(\Sigma_k^T \Sigma_k + \lambda I)^+ \Sigma_k^T) r\| \\ &= \left\| \left[\left(1 - \frac{\varsigma_1^2}{\varsigma_1^2 + \lambda}\right) r_1, \dots, \left(1 - \frac{\varsigma_\ell^2}{\varsigma_\ell^2 + \lambda}\right) r_\ell, 1, \dots, 1 \right]^T \right\|, \end{aligned}$$

and also (2.29) holds. □

As we have already observed after Lemma 2.3, (2.28) shows that the step-length is decreasing with λ. On the other hand, taking derivatives in (2.29), we notice that $\|R(x_k) + J(x_k)p(\lambda)\|$ is increasing with λ.

Finally, in the following lemma we show a property of model (2.27), that will be useful for the analysis in the following sections.

Lemma 2.18. *Let $M_k(p)$ be defined as in (2.27). Then for $p_k = p(\lambda_k)$ solution to (2.18) with $\lambda_k > 0$ it holds*

$$p_k = p(\lambda_k) = -J(x_k)^T (J(x_k)J(x_k)^T + \lambda_k I)^{-1} R(x_k), \tag{2.31}$$

$$M(p_k) = \lambda_k (J(x_k)J(x_k)^T + \lambda_k I)^{-1} R(x_k). \tag{2.32}$$

Proof. From the singular value decomposition of $J(x_k)$ it is possible to verify that for a positive λ_k it holds:

$$(J(x_k)^T J(x_k) + \lambda_k I)^{-1} J(x_k)^T = J(x_k)^T (J(x_k)J(x_k)^T + \lambda_k I)^{-1}.$$

Then, from (2.18) we get (2.31), that yields also

$$\begin{aligned} M_k(p_k) = &- J(x_k)J(x_k)^T (J(x_k)J(x_k)^T + \lambda_k I)^{-1} R(x_k) + R(x_k) \\ = &- J(x_k)J(x_k)^T (J(x_k)J(x_k)^T \lambda_k I)^{-1} R(x_k) \\ &+ (J(x_k)J(x_k)^T + \lambda_k I)(J(x_k)J(x_k)^T + \lambda_k I)^{-1} R(x_k) \\ = &\lambda_k (J(x_k)J(x_k)^T + \lambda_k I)^{-1} R(x_k). \end{aligned}$$

□

As for Trust-Region methods for unconstrained minimization, we can prove global convergence properties also for least squares problems, as stated in the following theorem.

Theorem 2.19 (Theorem 10.3 [85])**.** *Let $\eta_1 < \frac{1}{4}$ in Algorithm 2.1. Suppose that $\mathscr{L} = \{x \in \mathbb{R}^n \text{ s.t. } f(x) \le f(x_0)\}$ is bounded and that R_j, $j = 1, \dots, m$ are Lipschitz continuously differentiable in a neighbourhood of the level set $\mathscr{L}$. Suppose also that the approximate solution p_k^{TR} of (2.26) satisfies for each k the Cauchy decrease*

$$m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k^{TR}) \ge \theta \|J^T(x_k)R(x_k)\| \min \left[\Delta_k, \frac{\|J^T(x_k)R(x_k)\|}{\|J(x_k)^T J(x_k)\|} \right]$$

for some $\theta > 0$. Then we have that the sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = \lim_{k\to\infty} \|J(x_k)^T R(x_k)\| = 0. \tag{2.33}$$

2.4.4 Tikhonov method

With a proper choice of the regularizing parameters the Levenberg-Marquardt method can also be used to handle ill-posed problems. In the context of ill-posed inverse problems it is better known as nonstationary iterated Tikhonov method [53, 64]. Given a sequence of regularizing parameters $\{\lambda_k\}$, nonstationary iterated Tikhonov method solves a sequence of regularized problems (2.17). In Tikhonov method, in addition to the regularization parameter λ_k a regularizing matrix may be added too. Given a symmetric and positive definite matrix $L_k \in \mathbb{R}^{n\times n}$, at k-th iteration the following regularized problem is solved:

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}\|J(x_k)p + R(x_k)\|^2 + \frac{\lambda_k}{2}\|L_k p\|^2 \tag{2.34}$$

For $L_k = I$ we get the Levenberg-Marquardt method presented above. It has been shown that for ill-posed linear problems the choice of a matrix different from the identity improves the solution approximation [17, 29].

Parameter λ_k is usually addressed as the regularization parameter. The main difficulty in employing the Tikhonov method is to set it. It is a crucial choice, at it affects the qualities of the solution approximation, but it is in general difficult to make an appropriate a-priori choice. Usually a dynamically adjusted parameter is then employed. Specifically, in ill-posed inverse problem context the parameter is often defined to solve a specific condition that guarantees to the method regularizing properties [50, 52]. In next chapter we will provide a rigorous definition of regularizing properties and introduce possible choices of λ_k to guarantee them.

Part II

Ill-posed nonlinear least squares problems

CHAPTER

3 Introduction to Part II

In this part of the thesis we consider a particular class of nonlinear inverse problems, that of ill-posed problems [31, 55]. When using the term inverse problem, it is implicitly assumed that the considered problem is connected to another one, and the formulation of the one involves that of the other. Usually the problem that was studied first or that is simpler is called direct and the other inverse. However, when there is a real-world problem behind the mathematical problem studied the distinction is clear. The inverse problem is defined as the one that starts with the results to calculate the causes, for example the process of calculating from a set of observations the causal factors that produced them. Possible inverse problems are the calculation of the evolution of a system backwards in time, or the identification of physical parameters from observations of the evolution of the system (parameter identification problems). Such problems can be formulated as nonlinear least squares problems [31].

Here we adopt the following generic formulation. We assume $F:\mathbb{R}^n \to \mathbb{R}^m$ nonlinear and continuously differentiable, $m \geq n$ and $y \in \mathbb{R}^m$. We state the problem as

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)-y\|^2, \tag{3.1}$$

which is a special case of (1.2) with $R(x) = F(x)-y$. F is usually called forward or observation operator, which describes the explicit relationship between the data and the model parameters [31]. We assume (3.1) to be ill-posed [67, 105]. Hadamard was the first to introduce the idea of well-posedness of a problem and to give a rigorous definition, that can be used as well to define an ill-posed problem [31]:

Definition 3.1 (Hadamard's well-posedness)**.** *An inverse problem is said to be well-posed if all the following conditions hold:*

- *existence: for all admissible data a solution exists;*
- *uniqueness: for all admissible data the solution is unique;*
- *stability: the solution depends continuously on the data.*

If just one of these conditions does not hold, the problem is said to be ill-posed.

Many ill-posed problems are typically formulated as continuous operator equations or least squares problems defined on an infinite dimensional Hilbert space. For example, all continuous problems with compact operator are ill-posed, as the inverse of a compact operator cannot be continuous if defined on an infinite dimensional space. Typical examples of continuous ill-posed problems are Fredholm equations of the first kind or parameter identification problems [21, 48, 109]. Many finite-dimensional ill-posed problems actually arise from the discretization of those infinite-dimensional problems, and they inherit their ill-posedness.

In practical applications, the main difficulty in the solution of such problems is the lack of stability. Indeed, one has typically access to some measured data y^δ, which are noisy representations of the true data y such that

$$\|y - y^\delta\| \le \delta, \tag{3.2}$$

for some positive δ that is called the *noise level*. The noise level is assumed to be known and fixed, as it is the case when it arises from measured data.

One has then to cope with a noisy problem of the form

$$\min_{x \in \mathbb{R}^n} f_\delta(x) = \frac{1}{2}\|F(x) - y^\delta\|^2. \tag{3.3}$$

Due to the noise in the data, one does not have access to exact values of the objective function f, or to its derivatives.

Even if the perturbations on the data are small, they may be severely amplified if stability property does not hold, leading to a relative error in the solution that is much higher than the relative error in the data.

To understand better which problems arise in the solution of an ill-posed problem, let us consider a linear problem as an example [55, p.36]:

$$Ax = b, \qquad A \in \mathbb{R}^{m\times n},\, b \in \mathbb{R}^m. \tag{3.4}$$

In connection with discrete ill-posed systems, a peculiar characteristic of the singular value decomposition is very often found [55]: the singular values ς_i decrease gradually to zero, without significant gap in the spectrum and an increase in the size of A increases the number of small singular values. Then, generally the matrix in a ill-posed system is severely ill-conditioned and the ill-conditioning increases with the size of the problem. Also, as ς_i decreases u_i and v_i become more oscillatory, i.e. they are vectors with many sign changes.

The decay rate of the singular values is so fundamental for the behaviour of ill-posed problems that it is usually used to characterize the degree of ill-posedness of the problem [31, p.40], [55, 61, 62].

If we consider the singular value decomposition of A in (1.8), we can write

$$x = \sum_{i=1}^{n} (v_i^T x) v_i \qquad Ax = \sum_{i=1}^{n} \varsigma_i (v_i^T x) u_i$$

and the minimum norm solution x^* of (3.4) is given in (1.10). We notice that high-frequency components in x are more damped in Ax compared to the low frequency ones, so that the multiplication by A guarantees a smoothing effect. The inverse problem on the other hand has the opposite effect, it amplifies the high-frequency oscillations in the right hand side by a factor ς_i^{-1}, see (1.10), that is usually really big. Moreover, if we assume that the data are affected by additive noise, the problem becomes then $Ax = b + \epsilon$, with $\epsilon \in \mathbb{R}^m$ being a perturbation on the data b. Its solution x_ϵ is given by

$$x_\epsilon = A^+(b+\epsilon) = \sum_{i=1}^{n} \frac{u_i^T(b+\epsilon)}{\varsigma_i} v_i = x^* + \sum_{i=1}^{n} \frac{u_i^T \epsilon}{\varsigma_i} v_i. \tag{3.5}$$

The last term in (3.5) represents the difference between the solution x^* and that obtained with noisy data. Due to the presence of tiny singular values ς_i at the denominator, the error in the data is amplified in the solution.

Analogously, if we consider a nonlinear system, the system's ill-posedness will be reflected in the Jacobian matrix J, that may be severely ill-conditioned, with ill-conditioning again due to smallest singular values close to zero. It may also be potentially singular, or tending to the singularity as approaching the solution.

Similar considerations can be made also when continuous ill-posed problems are considered, employing the singular values expansion in place of the singular value decomposition. Indeed, the singular value decomposition of a matrix arising from the discretization of an operator is closely related to its singular values expansion [55].

All of this, makes it impossible to seek a solution of (3.1) by solving a problem of the form (3.3) with classical methods usually employed for well-posed problems. The sequence generated by such a method would indeed converge to one of its solutions which, as we said, may be arbitrarily far from those of the original problem. Moreover, in the analysis of these methods it is usually assumed to have a finite bound on the norm of the inverse of the Jacobian of F around a solution, which usually it is not possible to do in the context of ill-posed problems.

Then, specific methods must be devised that attempt to solve (3.1) in a stable way. These are called *regularizing methods*. We consider a special class of regularizing methods, the *iterative regularizing methods*. These are iterative approaches that through both the construction of the iterates x_k^δ and the choice of a suitable stopping criterion achieve the following regularizing properties [67, 104].

Definition 3.2. *An iterative method is said to be an iterative regularizing method if it provides the following properties, assuming that the iterations are stopped at index $k_*(\delta)$:*

- $x^\delta_{k_*(\delta)}$ is an approximation to a solution of (3.1);
- $x^\delta_{k_*(\delta)}$ converges to a solution of (3.1) as δ tends to zero;
- in the noise-free case, convergence to a solution of (3.1) occurs.

A great part of the literature on nonlinear ill-posed inverse problems deals with the solution of ill-posed operator equations in infinite dimensional Hilbert spaces [31, 67, 104]. Regularizing adaptations of existing procedures classically used for the solution of well-posed problems are widely studied. Both first [54, 88, 91] and second order methods [50, 51, 65, 66, 92, 101, 104] have been proposed to cope with them. See [67] for a wide overview of iterative regularizing methods for nonlinear ill-posed operator equations.

Here, we focus on discrete ill-posed problems and we propose suitable modifications of Levenberg-Marquardt schemes to make them regularizing methods. The case of zero residual problems has been investigated in the seminal papers [50, 52, 107, 113]. However, this study deserves further theoretical and numerical insights as well as an extension to the nonzero residual case.

Indeed, while the zero residual case is well-studied, we are aware only of [3] that considers convergence rates of Tikhonov method admitting also the nonzero residual case. However, nonzero residual problems arise in many applications. Usually indeed, jointly to observation errors also modelling errors are present, so that it is not realistic to assume that the data are attainable and one must admit the case $y \notin \mathscr{R}(F)$, even when exact data are considered [3, 6, 21]. This is the case when a mathematical model approximating a true distribution is fit to given data [25] or of parameter estimation problems [6, 21]. These problems are indeed usually formulated as least squares problems. See [3, Example 4.3] for an example of an ill-posed problem for which a solution of the zero residual problem cannot exist, but it exists a solution of the problem's least squares reformulation.

In the linear case it is common to consider the modelling errors as part of the noise in the data. The same algorithms as for zero residual problems are used, with a proper a-posteriori parameter choice, based on an estimate on the noise level that comprises both noise in the data and modelling errors [83]. However, it is generally difficult to estimate this last contribution. For this reason, we propose an ad hoc method for ill-posed least squares problems with nonzero residual, that does not need the estimation of modelling errors in order to choose the regularization parameter.

In this part of the thesis we consider then both zero and nonzero residual problems and methods where the regularization parameter is adaptively selected. This automatic update is a very desirable property in a regularizing method, as the choice of the regularization parameters is always difficult and critical.

This part is then divided into two main chapters.

Chapter 4 is devoted to the study of the zero residual case. We consider the regularizing Levenberg-Marquardt method proposed by M. Hanke in [50] and we discuss some issues related to its practical implementation, that have not been addressed in [50] or in related papers. Then, inspired by this procedure, we present a regularizing Trust-Region approach and we discuss its convergence and regularizing properties. In our approach the regularization parameter is adaptively chosen thanks to a specific rule for selecting the Trust-Region radius at each iteration. The convergence analysis is conducted under assumptions weaker than the invertibility of the Jacobian and different from those usually employed in the literature. Then this study also gives more insight into the study of Trust-Region methods.

We underline that Trust-Region methods for ill-posed problems are studied in [107, 113] too. Like our method, these are based on a specific condition for the selection of the regularization parameters that guarantees the desired regularizing properties. However, while in our approach parameters satisfying such condition are adaptively chosen via the Trust-Region radius choice, in [107] the satisfaction of the condition is assumed while in [113] it is explicitly enforced rejecting the step whenever it does not hold and reducing the trust- region radius.

Numerical tests are performed that show in practice the properties theoretically studied. In the tests slow convergence of the method is observed. Then in the numerical results section an adaptive rule for choosing the Trust-Region radius, different from that theoretically analysed but based on the same ideas, is presented to improve the rate of convergence. Tests made with this choice of the radius show the increased robustness of the proposed method compared to that presented in [50].

In Chapter 5 we consider the more general case of nonzero residual problems. We devise suitable extensions to this case of the assumptions and conditions employed in the previous chapter. We propose a nonstationary iterated Tikhonov procedure and a suitable elliptical Trust-Region reformulation that allows an automatic setting of the free regularization parameters, that guarantees regularizing properties to the method. Convergence and regularizing properties are proved for problems with small residual under mild conditions.

Finally, in Chapter 6 we briefly discuss the extension of our procedure to an infinite dimensional Hilbert setting. Our analysis is threefold. First we point out that all the results presented in the first two chapters can easily be extended to an infinite dimensional Hilbert setting. Then, we consider also a sequence of solutions of problems got by the projection of an infinite dimensional problem onto a sequence of finite dimensional spaces of increasing dimension. We discuss the convergence of such sequence to a solution of the infinite dimensional problem. Finally we present a model problem for which we show that the assumptions of our method hold.

Notations. In this chapters both noisy and unperturbed problems will be considered. Then, for seek of clarity, we will denote with x_k the iterates generated when the unperturbed problem (3.1) is considered and x_k^δ those generated when the noisy problem (3.3) is taken into account. By $x_0^\delta = x_0$ we denote an initial guess which may incorporate a-priori knowledge of an exact solution. We will denote m_k^{LM} the Levenberg-Marquardt model and p_k^{LM} the corresponding step; m_k^{TR} the Trust-Region model and simply p_k the corresponding step, for ease of notation. The symbol $\|\cdot\|$ indicates the Euclidean norm. A closed ball of radius r around a vector x is denoted as $B_r(x)$. The Jacobian matrix of $F(x)$ is denoted as

$J(x)$. Moreover we will denote $\nabla f(x)$ the gradient of $f(x)$ and

$$g_k = \nabla f_\delta(x_k) = J(x_k^\delta)^T(F(x_k^\delta) - y^\delta) \qquad\qquad B_k = J(x_k^\delta)^T J(x_k^\delta). \tag{3.6}$$

Either x^* or $x^\dagger$ will be used to denote a solution of (3.1). We denote with $F_j(x)$ and y_j^δ the j-th component of $F(x)$ and y^δ, respectively. We indicate the singular value decomposition of $J(x_k^\delta)$ as $U_k \Sigma_k V_k^T$ and the singular values as $\varsigma_1, \dots, \varsigma_\ell$, where ℓ is the rank of $J(x_k^\delta)$.

CHAPTER

4 Zero-residual problems

The content of this chapter has been the object of a publication in [J1]. We consider a special case of problem (3.1), that in which it can be assumed that a solution $x^\dagger$ exists such that $F(x^\dagger) = y$. As (3.1) has zero residual, the resulting system

$$F(x) = y \tag{4.1}$$

will be compatible. Then, we consider the solution of (4.1), that will provide a global minimum for (3.1).

Assuming to have at disposal just noisy data y^δ as in (3.2), in practice it is necessary to handle a problem of the form

$$F(x) = y^\delta. \tag{4.2}$$

When this problem is considered, the following assumption on F is commonly made, both in case of first and second order methods [50, 67, 97]:

Assumption 4.1. *Given an initial guess x_0, there exist positive r and c such that system (4.1) is solvable in $B_r(x_0)$, and*

$$\|F(x) - F(\tilde{x}) - J(x)(x-\tilde{x})\| \le c\|x-\tilde{x}\|\,\|F(x)-F(\tilde{x})\|, \quad x,\tilde{x} \in B_{2r}(x_0). \tag{4.3}$$

Condition (4.3) is known as the *tangential cone condition* and it is a requirement on the Taylor remainder of function F. It is motivated by the following observations. If F is continuously differentiable and J is Lipschitz continuous in a neighbourhood of x_0, it follows that for x and $\tilde{x}$ in that neighbourhood

$$F(\tilde{x}) - F(x) - J(x)(\tilde{x}-x) = \int_0^1 [J(x+t(\tilde{x}-x)) - J(x)](\tilde{x}-x)\,dt.$$

By Lipschitz continuity of J we obtain

$$\|F(\tilde{x}) - F(x) - J(x)(\tilde{x}-x)\| \le \frac{L}{2}\|\tilde{x}-x\|^2, \tag{4.4}$$

where L is the Lipschitz constant of J. However, in the context of ill-posed problems, this condition is not strong enough to prove regularization properties of

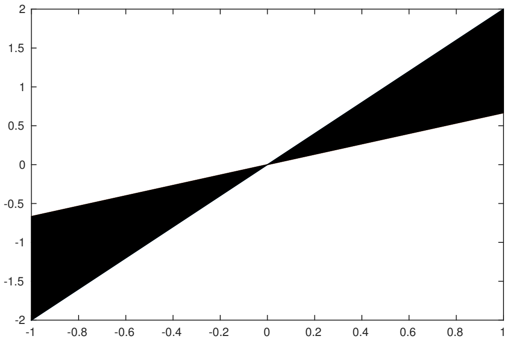


Figure 4.1: *Convex hull of $a(x)$ and $b(x)$, Figure 3.1 in [97].*

iterative regularization methods [67, §2.1]. In fact, the left hand side can be much smaller than the right hand side for certain pairs of points $\tilde{x}$ and x, whatever close to each other they are, and (4.4) carries too little information about the local behaviour of F around x to draw conclusions about convergence. For example, in case of rank-deficient Jacobian, it may happen that $\tilde{x}-x$ belongs to the null space of $J(x)$ and $F(\tilde{x}) = F(x)$. In this case, the bound on the right is too rough and we need to impose the stronger condition (4.3). It has been proved that it holds for many examples of continuous ill-posed nonlinear operator equations in a Hilbert setting, see for example [21, 98].

We can also give a geometrical interpretation of condition (4.3) [97]. We consider a slightly more general version of the condition, that is

$$\|F(x) - F(\tilde{x}) - J(x)(x-\tilde{x})\| \le \eta \|F(x) - F(\tilde{x})\|, \quad x, \tilde{x} \in B_r(x_0),$$

and we focus on the scalar case, then J is simply the first derivative of the function that we denote by F'. We assume $F : \mathbb{R} \to \mathbb{R}$ and we fix $\tilde{x} \in \mathbb{R}$ satisfying $F'(\tilde{x}) > 0$ and $\eta = 0.5$. The condition means that the graph of F lies entirely between $a(x) = F(\tilde{x}) + \frac{2}{3}F'(\tilde{x})(x-\tilde{x})$ and $b(x) = F(\tilde{x}) + 2F'(\tilde{x})(x-\tilde{x})$. For fixed $\tilde{x}$ the convex hull of $a(x)$ and $b(x)$ forms a cone with vertex $(\tilde{x}, F(\tilde{x}))$ around the tangent $F(\tilde{x}) + F'(\tilde{x})(x-\tilde{x})$, see Figure 4.1. We can conclude that for a continuously differentiable function $F : \mathbb{R} \to \mathbb{R}$, the condition with $\eta = 0.5$ and for fixed $\tilde{x}$ is satisfied if the graph of F is contained in the convex hull of $a(x)$ and $b(x)$.

Our method takes the step from the regularizing Levenberg-Marquardt method proposed by M. Hanke in [50, 52]. Indeed, problem (4.1) can be reformulated as a least squares problem and Levenberg-Marquardt method can be employed for its solution. The method by Hanke, assuming Assumption 4.1 to hold in a neighbourhood of an initial guess x_0 close enough to some solution $x^\dagger$ of (4.1), is able to compute a stable approximation to $x^\dagger$ or to some other solution of the unperturbed problem (4.1) close to $x^\dagger$.

This task is achieved through two key ingredients: an implicit step size control and an appropriate stopping criterion.

As we have shown in Section 2.4, for Levenberg-Marquardt methods the length of the step can be controlled through the choice of the regularizing parameter

λ. Through a specific choice of the parameter many goals can be achieved. For example, the method can be made globally convergent. However, if problem (4.1) is ill-posed, and the scalars λ_k are limited to promote convergence of the procedure [78, 80] the solution of (4.1) may be significantly misinterpreted [67, 105]. Then, in [50] a suitable condition has been individuated to compute the regularization parameters, to make the method regularizing, according to Definition 3.2, under the assumption that an initial guess x_0 in a neighbourhood of $x^\dagger$ is available. The condition is related to *Morozov discrepancy principle*, [48, p.44]. Let us consider an ill-posed linear system

$$Az = b, \qquad b \in \mathscr{R}(\mathscr{A}).$$

Assume that the data are affected by noise, so that just b^δ is available such that

$$\|b - b^\delta\| \leq \delta \leq \|b^\delta\|,$$

for a given noise level δ. Notice that the second inequality does not represent a further restriction, if it does not hold it means that the data are hopelessly corrupted and any analysis is ill-advised [48]. The problem with noisy data can be handled by Tikhonov method for linear problems, i.e. given $\lambda > 0$, regularizing the least squares reformulation of the problem as

$$\|Az - b^\delta\|^2 + \lambda\|z\|^2.$$

Let define $z^\delta(\lambda) = (A^TA + \lambda I)^{-1}A^Tb^\delta$, the solution of the regularized problem. Morozov in [81] asserts that the quality of the results of a computation should be comparable to the quality of the data. He suggests then to choose the free parameter λ to have a residual of the same order of the noise:

$$\|Az^\delta(\lambda) - b^\delta\| \sim \delta. \tag{4.5}$$

In the nonlinear case, at each iteration the nonlinear function is approximated by a linear model:

$$F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda).$$

Hanke imposes the following condition to define the regularization parameters at each iteration:

$$\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\| = q\|F(x_k^\delta) - y^\delta\|, \tag{4.6}$$

for some fixed $q \in (0,1)$. Then, this condition is coupled with a suitable stopping criterion. Namely, the iterative process is stopped at iteration $k_*(\delta)$, satisfying

$$\|y^\delta - F(x_{k_*(\delta)}^\delta)\| \leq \tau\delta < \|y^\delta - F(x_k^\delta)\|, \;\; 0 \leq k < k_*(\delta), \tag{4.7}$$

with $\tau > 1$ appropriately chosen [50], i.e. as soon as the residual reaches the noise level.

If we couple (4.6) and (4.7) we obtain that the computed solution approximation $x_{k_*(\delta)}^\delta$ satisfies

$$\|F(x_{k_*(\delta)}^\delta) - y^\delta + J(x_{k_*(\delta)}^\delta)p_{k_*(\delta)}\| \simeq \delta,$$

as in the linear case. The choice of a good stopping criterion is crucial. The generated sequence of solution approximations indeed is built considering the noisy problem (4.2). Then, it is necessary to stop the process before convergence is reached, to avoid approaching a noisy solution. In the solution of nonlinear ill-posed problems, a semi-convergence phenomenon is indeed usually observed. At the beginning of the optimization process the error $\|x_k^\delta - x^\dagger\|$ is decreasing, but eventually it starts to increase again. This means that, thanks to the regularizing properties of the method, at first a solution of the unperturbed problem is approached. But then, if the procedure is not stopped, the sequence begins to approach a solution of the noisy problem, and gets far from the sought point. Then it is crucial to understand when the process should be stopped. If (4.7) is employed as a stopping criterion, it means that also the residual of the original problem has reached the noise level. Indeed

$$\|y - F(x_{k_*(\delta)}^\delta)\| \le \|y^\delta - y\| + \|y^\delta - F(x_{k_*(\delta)}^\delta)\|.$$

As stated by Morozov, one cannot expect to gain a better result.

It is important to notice that while (4.5) always has a solution, [48, Theorem 3.3.1], the same does not hold for condition (4.6), unless a starting guess close enough to the sought solution is available, as we will show in the following. If it happens that (4.6) does not have a solution, it is not easy to choose a proper regularizing parameter. A non ad hoc choice could lead the method to loose its regularizing properties. Then in this thesis we consider the following variant of condition (4.6):

$$\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\| \ge q\|F(x_k^\delta) - y^\delta\|, \tag{4.8}$$

that in the following we will address as *q-condition*. It was first proposed in [50, Remark p. 6], but it was neither analyzed, nor employed for numerical computation. We will prove that it always has a solution and enforcing it we obtain a method that shares its regularizing properties with the method proposed by Hanke.

In the context of nonlinear problems, conditions such as (4.6) or (4.8) can be seen as a constraint on the length of the step. Their effect on it is illustrated in Figure 4.2, where we plot $\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\|$ (top) and $\|p(\lambda)\|$ (bottom) varying λ. As we will prove in Lemma 4.3, the norm of the model varies between $\|F(x_k^\delta) - y^\delta\|$ (dotted line) and the norm of the projection of the residual on the orthogonal complement of the range of $J(x_k^\delta)$ (dash-dot line). By imposing those conditions, in case (4.6) admits solution, the regularization parameter λ is forced to be equal or greater then the value λ_k^q satisfying (4.6), avoiding too small values that correspond to large steps, as it is shown at the bottom of Figure 4.2. The underlying idea is that too big steps could lead us too quickly towards a solution of the noisy problem.

We notice that whereas it is easy to understand how to compute a λ satisfying (4.6), it is not evident how to enforce (4.8). (4.6) indeed is a nonlinear scalar equation and it can be solved with Newton's method, that can perform well if applied to a suitable reformulation of the equation, as we will see in Section 4.1. On

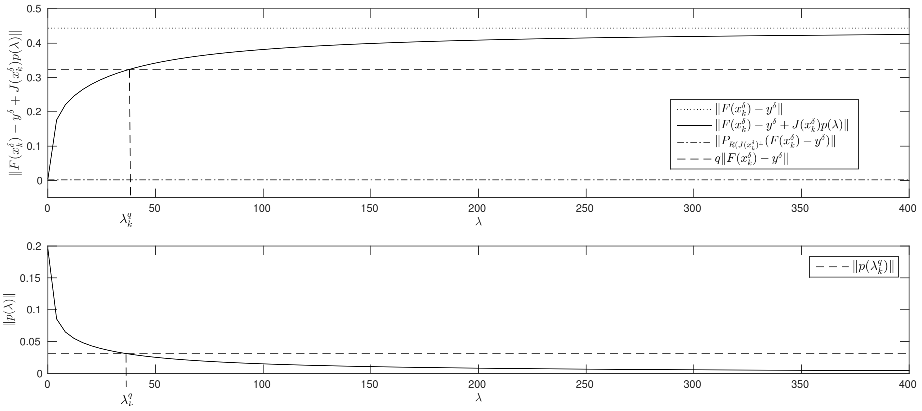


Figure 4.2: *Effect of q-condition on the step length.*

the other hand (4.8) is an inequality, then it is not clear how to select one among the parameter's values that satisfy the condition. Therefore, in this thesis we introduce and analyze a Trust-Region implementation of a Levenberg-Marquardt method, as described in Section 2.4.3. Our approach is based on a peculiar update of the Trust-Region radius, that guarantees an indirect choice of parameters λ that satisfy (4.8). The advantage of this approach compared to a 'pure' Levenberg-Marquardt method is then that the update of the radius allows us to obtain (4.8) without the need of selecting λ directly, that would not be a straightforward choice.

The resulting method, as a standard Trust-Region procedure, enforces a monotonic decrease of the value of the function

$$f_\delta(x) = \frac{1}{2}\|F(x) - y^\delta\|^2, \tag{4.9}$$

at the iterates x_k^δ, but also shares the same regularizing properties as the method by Hanke. The analysis is conducted under assumptions analogous to Assumption 4.1 used in [50].

Moreover, this work represents a contribution also in the local convergence theory for Trust-Region methods. Indeed, in the literature, when Trust-Region schemes are analyzed, assumptions stronger than (4.3) have been made. Typically, if squared systems are considered, local convergence properties of Trust-Region strategies are analyzed under assumptions which involve the inverse of J in a neighbourhood of a solution $x^\dagger$. It can be shown that condition (4.3) is weaker than the non singularity of the Jacobian in a neighbourhood of the solution, as we show in the following Lemma. The proof follows the lines of [69, Lemma 4.3.1].

Lemma 4.2. *Let x^* be a solution to $F(x) = y$ for $F : \mathbb{R}^n \to \mathbb{R}^n$. Let J be the Jacobian of F and assume that J is Lipschitz continuous in a neighbourhood of such solution and let $J(x^*)$ be nonsingular. Then it exists $r > 0$ such that (4.3) holds for $x, \tilde{x} \in B_r(x^*)$.*

Proof. If $J(x^*)$ is nonsingular, it exists $\rho > 0$ such that $J(x)$ is nonsingular for all $x \in B_\rho(x^*)$. Let $r < \min\{\rho, \frac{1}{8L\|J(x^*)^{-1}\|}\}$ for L the Lipschitz constant of J and let

$x, \tilde{x}, y \in B_r(x^*)$. From Lemma 4.3.1 in [69] $\|J(y)^{-1}\| \le 2\|J(x^*)^{-1}\|$ in $B_r(x^*)$. From this and the Lipschitz continuity of J it holds

$$\|I - J(\tilde{x})^{-1}J(y)\| = \|J(\tilde{x})^{-1}(J(\tilde{x}) - J(y))\| \le \|J(\tilde{x})^{-1}\|L\|y - \tilde{x}\| < 4rL\|J(x^*)^{-1}\| < \frac{1}{2}, \tag{4.10}$$

where the last inequality follows from the definition of r. Then,

$$\begin{aligned} J(\tilde{x})^{-1}(F(x) - F(\tilde{x})) &= J(\tilde{x})^{-1}\int_0^1 J(\tilde{x} + t(x - \tilde{x}))(x - \tilde{x})\,dt \\ &= (x - \tilde{x}) - \int_0^1 (I - J(\tilde{x})^{-1}J(\tilde{x} + t(x - \tilde{x})))(x - \tilde{x})\,dt. \end{aligned}$$

Setting $y = \tilde{x} + t(x - \tilde{x})$ in (8.29), because $\tilde{x} + t(x - \tilde{x}) \in B_r(x^*)$ for $0 \le t \le 1$ it follows

$$\|J(\tilde{x})^{-1}(F(x) - F(\tilde{x}))\| \ge \|x - \tilde{x}\|(1 - \int_0^1 \|I - J(\tilde{x})^{-1}J(\tilde{x} + t(x - \tilde{x}))\|) \ge \frac{\|x - \tilde{x}\|}{2}.$$

From (4.4) and the previous result it follows

$$\begin{aligned} \|F(x) - F(\tilde{x}) - J(x)(x - \tilde{x})\| &\le \frac{L}{2}\|x - \tilde{x}\|^2 \le L\|x - \tilde{x}\|\,\|J(\tilde{x})^{-1}\|\,\|F(x) - F(\tilde{x})\| \\ &\le c\|x - \tilde{x}\|\,\|F(x) - F(\tilde{x})\|, \end{aligned}$$

that is (4.3) holds in a neighbourhood of x^*. □

More recently (see Section 2.4.2 and references therein) the convergence analysis has been carried out assuming the so-called local error-bound condition (cf. Definition (2.16)) and Lipschitz continuity of the Jacobian in a neighbourhood of x^*, rather than the stronger nonsingularity of the Jacobian at a solution. We can relate condition (4.3) to the error bound condition if we restrict (4.3) to hold for $\tilde{x} = x^*$ and for x in a neighbourhood of x^*. Then, condition (4.3) becomes

$$\|F(x) - F(x^*) - J(x)(x - x^*)\| \le c\|x - x^*\|\,\|F(x) - F(x^*)\|, \tag{4.11}$$

for x in a neighbourhood of x^*. Condition (4.11) is weaker than the error bound condition. More precisely, if $\|F(x_k) - y\|$ provides a local error bound condition for (4.1), it exists $\gamma > 0$ such that $\|x_k - x^*\| \le \gamma\|F(x_k) - y\|$ which from (4.4) implies (4.11).

Like the error bound condition, our assumption allows the presence of non-isolated solutions. Then, here local convergence properties are established under conditions different than the conditions usually used in the literature for the local analysis.

This chapter is organized as follows. In Section 4.1 we describe the main features of the regularizing Levenberg-Marquardt method proposed by M. Hanke in [50]. We will focus especially on condition (4.6) and address the existence of a solution and its numerical resolution, that has not been considered in [50] or in related papers. In Section 4.2 we introduce our regularizing version of Trust-Region method and in Section 4.3 we study the local convergence properties. A comparative numerical analysis of all the procedures studied is reported in Section 4.4.

4.1 Regularizing Levenberg-Marquardt method for ill-posed problems

In the Levenberg-Marquardt method proposed in [50], as in standard Levenberg-Marquardt approaches, at the k-th iteration given $x_k^\delta \in \mathbb{R}^n$ and $\lambda_k > 0$, the model is defined as in (2.17). In this case it holds $R(x) = F(x) - y^\delta$, then the minimization problem becomes:

$$\min_{p\in\mathbb{R}^n} m_k^{LM}(x_k^\delta + p) = \frac{1}{2}\|J(x_k^\delta)p + F(x_k^\delta) - y^\delta\|^2 + \frac{\lambda_k}{2}\|p\|^2. \tag{4.12}$$

The step p_k^{LM} taken minimizes m_k^{LM}, and therefore it is the solution of the following linear system (cf. (2.18) and (3.6)):

$$(B_k + \lambda_k I)p = -g_k. \tag{4.13}$$

The new iterate is defined as $x_{k+1}^\delta = x_k^\delta + p_k^{LM}$. To achieve regularizing properties at each iteration the regularization parameter λ_k is chosen as the solution λ_k^q, if it exists, of the nonlinear scalar equation (4.6).

We sketch the k-th iteration of the procedure in Algorithm 4.1, assuming that a solution of (4.6) exists.

Algorithm 4.1 k-th iteration of the regularizing Levenberg-Marquardt method for problem (4.2)

```
Input: $x_k^\delta$, $q \in (0,1)$, $y^\delta$.
1. Compute $B_k = J(x_k^\delta)^T J(x_k^\delta)$ and $g_k = J(x_k^\delta)^T (F(x_k^\delta) - y^\delta)$.
2. Compute $\lambda_k^q$ satisfying (4.6).
3. Compute the solution $p_k^{LM}$ of subproblem (4.12) for $\lambda_k = \lambda_k^q$.
4. Set $x_{k+1}^\delta = x_k^\delta + p_k^{LM}$.
```

Notice that the step acceptance is not based on the ratio between the actual and the predicted reduction, since the regularizing term here is not intended to promote global convergence but to regularize the method. In the following lemma existence of a solution of (4.6) is discussed. The results reported in the Lemma can be found in [67, §4.1]. Suitable assumptions for λ_k^q are provided to be uniquely determined from the condition and an upper bound for it is derived. All of this is established recalling the results in Lemma 2.17, that states the behaviour of $\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\|$ as a function of λ for a Levenberg-Marquardt method.

Lemma 4.3. *Suppose $\|g_k\| = \|\nabla f_\delta(x_k)\| \neq 0$. Let $p(\lambda)$ be the minimum norm solution of (4.13) with $\lambda \geq 0$, $\mathscr{R}(J(x_k^\delta))^\perp$ be the orthogonal complement of the range $\mathscr{R}(J(x_k^\delta))$ of $J(x_k^\delta)$, and $P_{\mathscr{R}(J(x_k^\delta))^\perp}$ be the orthogonal projector onto $\mathscr{R}(J(x_k^\delta))^\perp$. Then*

(i) Equation (4.6) is solvable if and only if

$$\|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\| \leq q\|F(x_k^\delta) - y^\delta\|. \tag{4.14}$$

(ii) If

$$\|F(x_k^\delta) - y^\delta + J(x_k^\delta)(x^\dagger - x_k^\delta)\| \le \frac{q}{\theta_k}\|F(x_k^\delta) - y^\delta\| \tag{4.15}$$

for some $\theta_k > 1$, then (4.14) is satisfied and (4.6) has a unique solution λ_k^q such that

$$\lambda_k^q \in \left(0, \frac{q}{1-q}\|B_k\|\right]. \tag{4.16}$$

Proof. (i) We employ for the proof the results in Lemma 2.17, taking into account that in this case $R(x) = F(x) - y^\delta$ and $r_i = (U_k^T(F(x_k^\delta) - y^\delta))_i$.

First, taking into account that from the singular value expansion of $J(x_k^\delta)$

$$P_{\mathscr{R}(J(x_k^\delta))^\perp} = (I - J(x_k^\delta)J(x_k^\delta)^+)(F(x_k^\delta) - y^\delta) = U_k \begin{bmatrix} 0, 0 \\ 0, I_{m-\ell} \end{bmatrix} U_k^T(F(x_k^\delta) - y^\delta),$$

with $I_{m-\ell}$ the identity matrix of size $m - \ell$, it follows $\|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\|^2 = \sum_{i=\ell+1}^m r_i^2$. Equation (2.29) implies

$$\begin{aligned} \lim_{\lambda\to 0} \|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\| &= \|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\|, \\ \lim_{\lambda\to\infty} \|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\| &= \sqrt{\sum_{i=1}^{\ell} r_i^2 + \sum_{i=\ell+1}^{m} r_i^2} = \|F(x_k^\delta) - y^\delta\|. \end{aligned}$$

Thus, as from (2.29) $\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\|$ is an increasing function of λ, we conclude that (4.6) admits a solution if and only if $\|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\| \le q\|F(x_k^\delta) - y^\delta\|$. Also, if the solution exists, it is unique.

(ii) Trivially, as $J(x_k^\delta)(x - x_k^\delta) \in \mathscr{R}(J(x_k^\delta))$ for any x, it holds

$$\begin{aligned} \|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\| &= \|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta + J(x_k^\delta)(x - x_k^\delta))\| \\ &\le \|F(x_k^\delta) - y^\delta + J(x_k^\delta)(x - x_k^\delta)\|. \end{aligned}$$

Hence,

$$\begin{aligned} \|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta) - y^\delta)\| &\le \|F(x_k^\delta) - y^\delta + J(x_k^\delta)(x^\dagger - x_k^\delta)\| \\ &\le \frac{q}{\theta_k}\|F(x_k^\delta) - y^\delta\| < q\|F(x_k^\delta) - y^\delta\|, \end{aligned}$$

Then, from (i) (4.6) admits a solution λ_k^q which is positive and unique. Let's now bound it. From (2.32) and (4.6)

$$\begin{aligned} q\|F(x_k^\delta) - y^\delta\| &= \lambda_k^q\|(J(x_k^\delta)J(x_k^\delta)^T + \lambda_k^q I)^{-1}(F(x_k^\delta) - y^\delta)\| \\ &\ge \frac{\lambda_k^q}{\|B_k\| + \lambda_k^q}\|F(x_k^\delta) - y^\delta\| \end{aligned}$$

which yields (4.16). □

Then, from this lemma we conclude that it is not always guaranteed to have a solution of (4.6), unless (4.15) holds. In the following we will see that this is satisfied when a starting guess close enough to the desired solution is available. Indeed, the analysis in [50] is conducted under the following assumption:

Assumption 4.4. *Let x_0, c and r as in Assumption 4.1, $x^\dagger$ be a solution of (4.1) and x_0 satisfy*

$$\begin{aligned} \|x_0 - x^\dagger\| &< \min\left\{\frac{q}{c}, r\right\}, & \text{if } \delta = 0, \qquad (4.17)\\ \|x_0 - x^\dagger\| &< \min\left\{\frac{q\tau - 1}{c(1+\tau)}, r\right\}, & \text{if } \delta > 0, \qquad (4.18)\end{aligned}$$

where $\tau > 1/q$.

It can be proved that whenever x_k^δ belongs to $B_{2r}(x_0)$ and $\|x_k^\delta - x^\dagger\| < \|x_0 - x^\dagger\|$, Assumption 4.1 implies that inequality (4.15) is satisfied for some $\theta_k > 1$, and consequently there exists a solution to (4.6), cf. Lemmas 4.14 and 4.17.

Then, under Assumptions 4.1 and 4.4, Hanke proves that the Levenberg-Marquardt method in Algorithm 4.1 generates an approximation $x^\delta_{k_*(\delta)}$ satisfying the stopping criterion (4.7) and that the sequence $\{x^\delta_{k_*(\delta)}\}$ converges to a solution of (4.1) as δ tends to zero. This result is stated in the following theorem.

Theorem 4.5. *Let Assumptions 4.1 and 4.4 hold and x_k^δ be the Levenberg-Marquardt iterates generated by Algorithm 4.1. For noisy data, suppose $k < k_*(\delta)$ where $k_*(\delta)$ is defined in (4.7). Then, any iterate x_k^δ belongs to $B_{2r}(x_0)$. With exact data, the sequence $\{x_k\}$ converges to a solution of (4.1). With noisy data, the stopping criterion (4.7) is satisfied after a finite number $k_*(\delta)$ of iterations and the sequence of approximations $\{x^\delta_{k_*(\delta)}\}$ converges to a solution of (4.1) as δ tends to zero.*

Proof. See [50], Theorem 2.2 and Theorem 2.3. □

Let us now focus on a specific issue concerning the implementation of the method which, to our knowledge, has not been addressed either in [50] or in related papers. The numerical solution of (4.6) requires the application of a root-finder method and Newton's method is the most efficient procedure, though in general it requires the knowledge of an accurate approximation to the solution. On the other hand, nonlinear equations which are monotone and convex (or concave) on some interval containing the root are particularly suited to the application of Newton's method, as stated in the following theorem:

Theorem 4.6 (Theorem 4.8 [58])**.** *Let G be defined and twice continuously differentiable on the closed finite interval $[a,b]$, and let the following conditions be satisfied:*

1. $G(a)G(b) < 0$,
2. $G'(x) \neq 0, x \in [a,b]$,
3. *$G''(x)$ is either ≥ 0 or ≤ 0 for all $x \in [a,b]$,*
4. *$\left|\frac{G(c)}{G'(c)}\right| \leq b - a$ for c the endpoint of $[a,b]$ at which $|G'(x)|$ is smaller.*

Then Newton's method converges to the (only) solution of $G(x) = 0$ for any choice of $x \in [a,b]$.

Equation (4.6) does not have such properties but we can reformulate it as an equivalent equation with strictly decreasing and concave function in $[\lambda_k^q, \infty)$. Thus, Newton's method applied to the reformulated equation converges globally to λ_k^q whenever the initial guess overestimates such a root.

Lemma 4.7. *Suppose $\|F(x_k^\delta) - y^\delta\| \neq 0$, and that (4.6) has positive solution λ_k^q. Let*

$$\psi(\lambda) = \frac{\lambda}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\|} - \frac{\lambda}{q\|F(x_k^\delta) - y^\delta\|} = 0. \tag{4.19}$$

Then, Newton's method applied to (4.19) converges monotonically and globally to the root λ_k^q of (4.6) for any initial guess in the interval $[\lambda_k^q, \infty)$.

Proof. Trivially, solving (4.6) is equivalent to finding the positive root of equation (4.19). We now show that $\psi(\lambda)$ is strictly decreasing in $[\lambda_k^q, \infty)$ and concave on $(0, \infty)$. By (2.29),

$$\frac{\lambda}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\|} = \left(\sqrt{\sum_{i=1}^{\ell} \left(\frac{r_i}{\varsigma_i^2 + \lambda} \right)^2 + \sum_{i=\ell+1}^{m} \left(\frac{r_i}{\lambda} \right)^2} \right)^{-1}, \tag{4.20}$$

and this function is concave on $(0, \infty)$, cf. [18, Lemma 2.1]. Thus, ψ is concave on $(0, \infty)$ and trivially $\psi'(\lambda)$ is strictly decreasing.

Now we show that $\psi'(\lambda_k^q)$ is negative; thus, using the monotonicity of $\psi'(\lambda)$, we obtain that $\psi(\lambda)$ is strictly decreasing in $[\lambda_k^q, \infty)$. Differentiation of $\psi(\lambda)$ and (4.6) give

$$\begin{aligned} \psi'(\lambda_k^q) &= \frac{(\lambda_k^q)^3}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda_k^q)\|^3} \left(\sum_{i=1}^{\ell} \frac{r_i^2}{(\varsigma_i^2 + \lambda_k^q)^3} + \sum_{i=\ell+1}^{m} \frac{r_i^2}{(\lambda_k^q)^3} \right) - \frac{1}{q\|F(x_k^\delta) - y^\delta\|} \\ &= \frac{(\lambda_k^q)^2}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda_k^q)\|^3} \left(\sum_{i=1}^{\ell} \frac{r_i^2 \lambda_k^q}{(\varsigma_i^2 + \lambda_k^q)^3} + \sum_{i=\ell+1}^{m} \left(\frac{r_i}{\lambda_k^q} \right)^2 - \frac{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda_k^q)\|^2}{(\lambda_k^q)^2} \right). \end{aligned}$$

Moreover, using (4.20), it holds

$$\begin{aligned} \psi'(\lambda_k^q) &= \frac{(\lambda_k^q)^2}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda_k^q)\|^3} \left(\sum_{i=1}^{\ell} \frac{r_i^2 \lambda_k^q}{(\varsigma_i^2 + \lambda_k^q)^3} - \sum_{i=1}^{\ell} \left(\frac{r_i}{\varsigma_i^2 + \lambda_k^q} \right)^2 \right) \\ &= -\frac{(\lambda_k^q)^2}{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda_k^q)\|^3} \sum_{i=1}^{\ell} \frac{r_i^2 \varsigma_i^2}{(\varsigma_i^2 + \lambda_k^q)^3}, \end{aligned}$$

i.e. $\psi'(\lambda_k^q)$ is negative. The claimed convergence of Newton's method follows from the result given in Theorem 4.6. Indeed, the theorem's assumptions hold in $[a, b]$ with $a = \lambda_k^q - \epsilon$ for $\epsilon > 0$ and $b > \lambda_k^q$. The first three items follow directly from the monotonicity results proved on ψ and ψ', while the fourth can be proved as follows. In our case $c = a$ from the monotonicity of ψ'. From the mean value theorem, it exists $\xi \in (a, \lambda_k^q)$ such that

$$\psi(a) = \psi(\lambda_k^q) - \psi'(\xi)(\lambda_k^q - a) = \psi'(\xi)(a - \lambda_k^q).$$

From this relation and again from the monotonicity of ψ' we obtain

$$\frac{|\psi(a)|}{|\psi'(a)|} \le \frac{|\psi(a)|}{|\psi'(\xi)|} \le |a - \lambda_k^q| \le |a - b|,$$

and item 4. is proved. □

For the practical evaluation of $\psi(\lambda)$ and $\psi'(\lambda)$ we refer to Section 2.2.1.1 and references therein.

In the next section we present our proposal of a regularizing Trust-Region method such that, differently from the approach sketched in Algorithm 4.1, selects the step to satisfy the q-condition (4.8) in an adaptive way. We underline that with this approach λ_k, which is implicitly defined by the Trust-Region approach, is well-defined even if x_k^δ is not close to a solution.

4.2 Regularizing Trust-Region method

We propose here a Trust-Region method inspired by the Levenberg-Marquardt method presented in the previous section. Deviating from what is reported in Section 2.4.3, the choice of the Trust-Region radius update is not just aimed at making the process globally convergent. Our main objective will indeed be that of providing a nonlinear step size control that enforces both the monotonic reduction of f_δ and the q-condition (4.8), to ensure to the method also regularizing properties.

The peculiarity of our approach is that, enforcing condition (4.8), it provides strictly positive parameters λ_k, as it is needed to regularize the Gauss Newton system (2.15). From (2.7c) it follows that it must hold, for $p_k = p(\lambda_k)$, $\|p_k\| = \Delta_k$. Namely the Trust Region is active along all the optimization process, differently from standard Trust-Region methods. As stated in Remark 2.8, with standard radius update strategies the Trust Region towards the end of the process results to be inactive.

At each iteration our method takes a step p_k solving the following Trust-Region subproblem:

$$\begin{aligned} &\min_p m_k^{TR}(x_k + p) = \|J(x_k^\delta)p + F(x_k^\delta) - y^\delta\|^2 \\ &\text{s.t. } \|p\| \le \Delta_k. \end{aligned} \tag{4.21}$$

with Δ_k appropriately chosen to let the step satisfy (4.8). We remind that from Theorem 2.2 a solution of (4.21) satisfies (4.13) with λ_k solution of $\lambda(\|p(\lambda)\| - \Delta_k) = 0$.

We first show that (4.8) always has a solution and we characterize the parameters λ such that $p(\lambda)$ satisfies it.

Lemma 4.8. *Assume $\|g_k\| = \|\nabla f_\delta(x_k)\| \neq 0$. Let $p(\lambda)$ be the minimum norm solution of (4.13) with $\lambda \ge 0$ and $P_{\mathscr{R}(J(x_k^\delta))^\perp}$ be the orthogonal projector onto $\mathscr{R}(J(x_k^\delta))^\perp$. Then, equation (4.8) is satisfied for any $\lambda \ge 0$ whenever (4.14) does not hold. Otherwise, it is satisfied for any $\lambda \ge \lambda_k^q$ where λ_k^q is the solution to (4.6).*

Proof. If (4.14) does not hold,

$$q\|F(x_k^\delta)-y^\delta\| \le \|P_{\mathscr{R}(J(x_k^\delta))^\perp}(F(x_k^\delta)-y^\delta)\| \le \|F(x_k^\delta)-y^\delta+J(x_k^\delta)p(\lambda)\|,$$

and (4.8) is satisfied for all $\lambda \ge 0$. If (4.14) holds, from Lemma 4.3 it exists a solution λ_k^q of (4.6). Then, (4.8) is satisfied for all $\lambda \ge \lambda_k^q$, as $\|F(x_k^\delta)-y^\delta+J(x_k^\delta)p(\lambda_k^q)\|$ is an increasing function of λ from (2.29). □

We are now ready to show that an appropriate bound on the Trust-Region radius size can be found that ensures the resulting step p_k to guarantee (4.8).

Lemma 4.9. *Let p_k solve the Trust-Region problem (4.21). If*

$$\Delta_k \le \frac{1-q}{\|B_k\|}\|g_k\|, \tag{4.22}$$

then p_k satisfies the q-condition (4.8).

Proof. By Lemma 4.8 we know that the q-condition is satisfied either for $\lambda \ge 0$, or for any $\lambda \ge \lambda_k^q$. In the former case, the claim trivially holds. In the latter case, by (4.13) it follows

$$\|p(\lambda_k^q)\| \ge \frac{\|g_k\|}{\|B_k+\lambda_k^q I\|},$$

and by (4.16) it holds

$$\|B_k+\lambda_k^q I\| \le \|B_k\| + \frac{q}{1-q}\|B_k\| = \frac{\|B_k\|}{1-q}.$$

From Theorem 2.2 it exists λ_k satisfying conditions 2.7 such that $p_k = p(\lambda_k)$. By construction $\|p_k\| \le \Delta_k$, and if (4.22) holds we obtain

$$\|p_k\| = \|p(\lambda_k)\| \le \frac{1-q}{\|B_k\|}\|g_k\| \le \frac{\|g_k\|}{\|B_k+\lambda_k^q I\|} \le \|p(\lambda_k^q)\|.$$

Since $\|p(\lambda)\|$ is monotonically decreasing from Theorem 2.17, it follows $\lambda_k \ge \lambda_k^q > 0$ and condition (4.8) is satisfied. □

We stress that the bound (4.22) provides a practical rule for choosing the Trust-Region radius and enforcing the q-condition (4.8). Conversely, in papers [107, 113], where Trust-Region methods for ill-posed problems are studied, such a condition is respectively assumed to be satisfied, and explicitly enforced rejecting the step whenever it does not hold. We also notice that from (4.22) we have that the Trust-Region radius converges to zero as $\|g_k\|$ goes to zero. Here, the radius converging to zero helps to maintain an active Trust Region. This feature may jeopardize a fast local rate of convergence, but a too fast convergence may bring the sequence too quickly towards a solution of a noisy problem.

The result in Lemma 4.9 suggests the Trust-Region iteration described in Algorithm 4.2.

Algorithm 4.2 is well-defined, provided that the following assumption is met.

Algorithm 4.2 k-th iteration of the regularizing Trust-Region method for problem (4.2)

```
**Input:** Given $x_k^\delta$, $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < C_{\min} < C_{\max}$, $q \in (0,1)$, $y^\delta$.
1. Compute $B_k = J(x_k^\delta)^T J(x_k^\delta)$ and $g_k = J(x_k^\delta)^T (F(x_k^\delta) - y^\delta)$.
2. Choose $\Delta_k \in \left[ C_{\min} \|g_k\|, \min\left\{ C_{\max}, \frac{1-q}{\|B_k\|} \right\} \|g_k\| \right]$.
3. Repeat
   3.1 Compute the solution $p_k$ of the Trust-Region problem (4.21).
   3.2 Compute $\rho_k(p_k) = \frac{f_\delta(x_k) - f_\delta(x_k + p_k)}{m_k^{TR}(x_k) - m_k^{TR}(x_k + p_k)}$, with $f_\delta$ defined in (4.9).
   3.3 If $\rho_k(p_k) < \eta$, then set $\Delta_k = \gamma \Delta_k$.
Until $\rho_k(p_k) \geq \eta$.
4. Set $x_{k+1}^\delta = x_k^\delta + p_k$.
```

Assumption 4.10. *There exists a positive constant κ_J such that*

$$\|J(x)\| \leq \kappa_J,$$

for any x belonging to the level set $\mathscr{L} = \{x \in \mathbb{R}^n \text{ s.t. } f_\delta(x) \leq f_\delta(x_0)\}$.

First, step 2 is well defined for suitable choices of $C_{\min}$. In fact, as long as $C_{\min} < \dfrac{1-q}{\kappa_J^2}$, it holds $C_{\min} < \dfrac{1-q}{\|B_k\|}$ for all k. We point out that a suitable choice of this parameter requires an estimation of the bound κ_J. In the numerical results section we will propose an alternative Trust-Region radius choice that does not require neither the computation of B_k nor the estimation of κ_J, but that is shown to provide in practice the same regularizing properties of the choice at step 2 of Algorithm 4.2. Parameter $C_{\max}$ is just a safeguard to prevent the constant multiplying $\|g_k\|$ from becoming too big in case $\|B_k\|$ is small. However, even in case $C_{\max} < \frac{1-q}{\|B_k\|}$, (4.22) holds.

Second, due to well-known properties of Trust-Region methods, Assumption 4.10 guarantees that the step p_k is found within a finite number of attempts, whenever $\|g_k\| \neq 0$ [22, Theorem 7.3.4].

Global convergence of the Trust-Region method in Algorithm 4.2 in absence of noise is stated in the following theorem:

Theorem 4.11. *Suppose that $\delta = 0$ and that f is bounded below on the level set $\mathscr{L} = \{x \in \mathbb{R}^n \text{ s.t. } f(x) \leq f(x_0)\}$ and Lipschitz continuously differentiable in a neighbourhood of $\mathscr{L}$. Suppose further that $\|B_k\| \leq \beta$ for all k. Then we have that the sequence $\{x_k\}$ generated by Algorithm 4.2 satisfies*

$$\lim_{k \to \infty} g_k = 0.$$

Proof. The proof follows from adaptations of the proofs of Theorem 4.5 and 4.6 in [84]. Specifically, the proof of Theorem 4.5 can be modified as follows. Trivially, as subproblem (4.21) is solved exactly, the Cauchy decrease (2.13) is achieved. Let us assume that it exists $\epsilon > 0$ and an index $K > 0$ such that $\|g_k\| \geq \epsilon$ for all $k \geq K$. We first notice that from the updating rule at step 2 of Algorithm 4.2

$$\Delta_k \geq C_{\min}\epsilon \tag{4.23}$$

for all $k \geq K$. Let us assume that it exists an infinite sequence $\mathscr{K}$ of iterates such that $\rho_k(p_k) \geq \eta$ for all $k \geq \mathscr{K}$. Then, for $k \in \mathscr{K}$ and $k \geq \mathscr{K}$, from (2.13) it follows:

$$f(x_k) - f(x_{k+1}) \geq \eta(m_k^{TR}(p_k) - m_k^{TR}) \geq \eta\theta\epsilon \min\left[\Delta_k, \frac{\epsilon}{\beta}\right], \tag{4.24}$$

with θ defined in (2.13). Since f is bounded below, it follows from this inequality that

$$\lim_{k\in\mathscr{K}, k\to\infty} \Delta_k = 0,$$

contradicting (4.23). Hence such a sequence cannot exist, then we must have $\rho_k(p_k) < \eta$ for all k sufficiently large. In this case Δ_k will eventually be reduced at every iteration and again the Trust-Region radius will converge to zero, which again contradicts (4.23). Hence, the original assertion must be false giving $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$. The stronger result $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ follows repeating the arguments in Theorem 4.6 and employing the results we have just proven. □

This theorem allows us to establish global convergence of the sequence generated by Algorithm 4.2 in case of exact data and in case f is Lipschitz continuously differentiable. By the step acceptance rule at step 3 of Algorithm 4.2, the sequence $\{\|F(x_k^\delta) - y^\delta\|\}$ is monotonically decreasing and bounded below by zero; hence it is convergent. The result in Theorem 4.11 implies that any accumulation point of the sequence $\{x_k^\delta\}$ is a stationary point of f_δ. Then, in case of exact data, we conclude that if there exists an accumulation point of $\{x_k\}$ solving (4.1), then any accumulation point of the sequence solves (4.1). In the case of noisy data, the process is stopped before the convergence is reached, to avoid approaching a solution of the noisy problem. In this case we are not considering the global convergence issue. If data are affected by noise indeed it is not reasonable to think to be able to achieve convergence to the noise free solution from an arbitrary starting guess. In fact, in this context the starting guess must contain some information on the true solution. Global converge issues to our knowledge have been only faced in [66], where a multilevel approach is considered. Such an approach allows to first solve the problem on very coarse grid on which the problem is well-posed and to get closer to a true solution, so that a regularizing method on finer grids will then be able to converge.

What we will able to prove in the following section is that if one of the iterates of the generated sequence gets close enough to the true solution, there exists an iterate $x_{k_*(\delta)}^\delta$ such that the discrepancy principle (4.7) is met.

Then summarizing:

- *Noise free case*: we have a globally convergent approach with an adaptive choice of λ_k that is well defined even far from the solution. Moreover rank-deficient Jacobian matrices are allowed and therefore nonisolated solutions.

- *Noisy case*: We will see in the next sections that we obtain local regularizing properties as in [50] but with an overall more robust method as the choice of the scalar q is less crucial (see numerical results section).

4.3 Local behaviour of the Trust-Region method

In this section we analyze the local properties of the Trust-Region method, namely we show the behaviour of the iterates generated by Algorithm 4.2 when, for some k, x_k^δ is sufficiently close to a solution $x^\dagger$ of (4.1). For instance, this occurs with exact data when the accumulation points of $\{x_k\}$ solve (4.1) and k is sufficiently large.

We show that the proposed Trust-Region method shares the same local regularizing properties as the regularizing Levenberg-Marquardt method. To this purpose, we analyze the local properties of the method under the same assumptions made for the Levenberg-Marquardt method. We suppose that there exists an iteration index $\bar{k}$ such that the iterate $x_{\bar{k}}^\delta$ satisfies the following assumptions, that are the counterpart of Assumptions 4.1 and 4.4.

Assumption 4.12. *Suppose that for some iteration index $\bar{k}$ there exist positive r and c such that system (4.1) is solvable in $B_r(x_{\bar{k}}^\delta)$, and*

$$\|F(x)-F(\tilde{x})-J(x)(x-\tilde{x})\| \le c\|x-\tilde{x}\|\,\|F(x)-F(\tilde{x})\|, \quad x,\tilde{x} \in B_{2r}(x_{\bar{k}}^\delta), \tag{4.25}$$

with $\bar{k} < k_*(\delta)$ if the data are noisy, where $k_*(\delta)$ is defined in (4.7). Moreover, letting $x^\dagger$ be a solution of (4.1), suppose that $x_{\bar{k}}^\delta$ satisfies

$$\begin{aligned} \|x_{\bar{k}}-x^\dagger\| &< \min\left\{\frac{q}{c}, r\right\}, & \text{if } \delta = 0, \quad (4.26)\\ \|x_{\bar{k}}^\delta-x^\dagger\| &< \min\left\{\frac{q\tau-1}{c(1+\tau)}, r\right\}, & \text{if } \delta > 0. \quad (4.27)\end{aligned}$$

where $\tau > 1/q$.

Here we report a result that holds both in the noisy and in the noisy free case. Then, the local behaviour will be analyzed, distinguishing between the noise free (Section 4.3.1) and the noisy case (Section 4.3.2).

We show that if condition (4.15) is satisfied at iteration k, for that iteration the error between the current approximation and the solution $x^\dagger$ decreases. We will show in the following sections that both in the noise free and in the noisy case, it is possible to enforce (4.15) in a neighbourhood of the solution.

Lemma 4.13. *Assume that equation (4.15) is fulfilled for some $\theta_k > 1$, and let $x_{k+1}^\delta = x_k^\delta + p_k$ with $p_k = p(\lambda_k)$ satisfying (4.13) and (4.8). Then it holds*

$$\|x_k^\delta - x^\dagger\|^2 - \|x_{k+1}^\delta - x^\dagger\|^2 > \frac{2(\theta_k - 1)}{\theta_k \lambda_k}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2. \tag{4.28}$$

Proof. The proof parallels that of [67, Proposition 4.1]. Let $M_k(p) = F(x_k^\delta) - y^\delta + J(x_k^\delta)p$ and $e_k = x^\dagger - x_k^\delta$. It holds:

$$\begin{aligned}\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 &= 2 < x_{k+1}^\delta - x_k^\delta, x_k^\delta - x^\dagger > + \|x_{k+1}^\delta - x_k^\delta\|^2 \\ &= -2 < p_k, e_k > + \|p_k\|^2.\end{aligned}$$

From (2.31) and (2.32) with $R(x) = F(x) - y^\delta$ it follows

$$< p_k, e_k > = < -(J(x_k^\delta)J(x_k^\delta)^T + \lambda_k I)^{-1}(F(x_k^\delta) - y^\delta), J(x_k^\delta)e_k > = -\frac{1}{\lambda_k} < M_k(p_k), J(x_k^\delta)e_k > .$$

After an easy algebraic manipulation (adding $\pm(F(x_k^\delta) - y^\delta), \pm J_{\delta_k}(x_k)p_k$ to $J(x_k^\delta)e_k$) and taking into account the definition of $M_k(p)$, we obtain

$$< p_k, e_k > = -\frac{1}{\lambda_k} < M_k(p_k), M_k(e_k) > + \frac{1}{\lambda_k} < M_k(p_k), M_k(p_k) > - \frac{1}{\lambda_k} < M_k(p_k), J(x_k^\delta)p_k > .$$

Notice that from (2.31) and (2.32) $J(x_k^\delta)^T M_k(p_k) = -\lambda_k p_k$. Then,

$$\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 \le \frac{2}{\lambda_k}\|M_k(p_k)\|\|M_k(e_k)\| - \frac{2}{\lambda_k}\|M_k(p_k)\|^2 - \|p_k\|^2.$$

From (4.15) and (4.8) it follows that

$$\begin{aligned}\|M_k(e_k)\| &= \|F(x_k^\delta) - y^\delta + J(x_k^\delta)(x^\dagger - x_k^\delta)\| \\ &\le \frac{1}{\theta_k}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\| = \frac{1}{\theta_k}\|M_k(p_k)\|,\end{aligned} \tag{4.29}$$

which yields

$$\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 < \frac{2}{\lambda_k}\left(\frac{1}{\theta_k} - 1\right)\|M_k(p_k)\|^2$$

and the thesis follows. □

4.3.1 Noise free case

In this section we focus on the noise free case. We assume that $\delta = 0$ and we drop the symbol δ from the generated sequence, the data y and the function. We show that the norm of the error $\|x_k - x^\dagger\|$ decreases in a monotonic way for $k \ge \bar{k}$, the sequence $\{x_k\}$ remains in a neighbourhood of $x_{\bar{k}}$ and converges to a solution of (4.1).

First, we show that it is possible to enforce condition (4.15) if k is big enough.

Lemma 4.14. *Suppose that Assumptions 4.10 and 4.12 hold and let $\delta = 0$. Then, for $k \geq \bar{k}$ it exists $\theta_k > 1$ such that (4.15) holds.*

Proof. From (4.25), choosing $\tilde{x} = x^\dagger$ and $x = x_{\bar{k}}$ it follows that

$$\|F(x_{\bar{k}}) - y - J(x_{\bar{k}})(x_{\bar{k}} - x^\dagger)\| \leq c\|x_{\bar{k}} - x^\dagger\|\,\|F(x_{\bar{k}}) - y\|.$$

Then, condition (4.15) is satisfied at $k = \bar{k}$ with $\theta_{\bar{k}} = \dfrac{q}{c\|x_{\bar{k}} - x^\dagger\|} > 1$ from (4.26). Since Lemma 4.13 holds for $k = \bar{k}$, (4.28) implies $\|x_{\bar{k}+1} - x^\dagger\| < \|x_{\bar{k}} - x^\dagger\|$ and, as a consequence, $x_{\bar{k}+1}$ belongs to $B_{2r}(x_{\bar{k}})$ and to $B_r(x^\dagger)$. Repeating the above arguments, by induction we can prove that condition (4.15) holds for $k > \bar{k}$, with

$$\theta_k = \frac{q}{c\|x^\dagger - x_k\|} > 1. \tag{4.30}$$

□

The following results show the local behaviour of the regularizing Trust-Region method. We prove that the iterates x_k with $k > \bar{k}$ remain into the ball $B_r(x^\dagger)$ and the regularizing parameters are bounded above.

Lemma 4.15. *Suppose that Assumptions 4.10 and 4.12 hold and let $\delta = 0$. Then, Algorithm 4.2 generates a sequence $\{x_k\}$ such that, for $k \geq \bar{k}$,*

(i) x_k belongs to $B_{2r}(x_{\bar{k}}) \cap B_r(x^\dagger)$, $\|x_{k+1} - x^\dagger\| < \|x_k - x^\dagger\|$ and $\theta_{k+1} > \theta_k$.

(ii) There exists a constant $\bar{\lambda} > 0$ such that $\lambda_k \leq \bar{\lambda}$.

Proof. *(i)* From Lemma 4.14 it follows that condition 4.15 is satisfied for all $k \geq \bar{k}$. Consequently, Lemma 4.13 implies that $\{\|x_k - x^\dagger\|\}_{k=\bar{k}}^{\infty}$ is a monotonic decreasing sequence and, as a consequence, x_k belongs to $B_{2r}(x_{\bar{k}}) \cap B_r(x^\dagger)$ for all $k \geq \bar{k}$. Notice also that from this, the sequence $\{\theta_k\}_{k=\bar{k}}^{\infty}$, with θ_k given in (4.30), is monotonic increasing.

(ii) From Lemma 4.14 condition (4.15) is satisfied for all $k \geq \bar{k}$. Then, from Lemma 4.8 $\lambda_k > 0$. Then, by (2.7c) the Trust Region must be active. From (4.13)

$$\Delta_k = \|p_k\| = \|(B_k + \lambda_k I)^{-1} g_k\| \leq \frac{\|g_k\|}{\lambda_k}. \tag{4.31}$$

Thus our claim follows if $\Delta_k/\|g_k\|$ is larger than a suitable threshold, independent from k. Let us provide such a bound, by estimating the value of Δ_k which guarantees condition $\rho_k(p_k) \geq \eta$.

If this condition is fulfilled for the value of Δ_k fixed in step 2 of Algorithm 4.2, then $\Delta_k/\|g_k\| \geq C_{\min}$. Otherwise, the Trust-Region radius is progressively reduced, and we provide a bound for the value of Δ_k at termination of step 3 of Algorithm 4.2 in the case where $f(x_k + p_k) > m_k^{TR}(p_k)$.

This occurrence represents the most adverse case. In fact if $f(x_k + p_k) \leq m_k^{TR}(p_k)$ then $\rho_k(p_k) \geq 1 > \eta$ and the repeat loop terminates for a Trust-Region radius greater than or equal to the one estimated below. Trivially,

$$1 - \rho_k(p_k) = \frac{f(x_k + p_k) - m_k^{TR}(p_k)}{f(x_k) - m_k^{TR}(p_k)}, \tag{4.32}$$

and

$$\begin{aligned}
f(x_k+p_k)-m_k^{TR}(p_k) &= \frac{1}{2}\|F(x_k+p_k)-y\|^2-\frac{1}{2}\|F(x_k)-y+J(x_k)p_k\|^2 = \\
&= \frac{1}{2}\|F(x_k+p_k)-y\pm F(x_k)\pm J(x_k)p_k\|^2-\frac{1}{2}\|F(x_k)-y+J(x_k)p_k\|^2 \\
&\le \frac{1}{2}\|F(x_k+p_k)-F(x_k)-J(x_k)p_k\|^2+\frac{1}{2}\|F(x_k)-y+J(x_k)p_k\|^2 \\
&\quad +\|F(x_k+p_k)-F(x_k)-J(x_k)p_k\|\|F(x_k)-y+J(x_k)p_k\| \\
&- \frac{1}{2}\|F(x_k)-y+J(x_k)p_k\|^2 \\
&= \frac{1}{2}\|F(x_k+p_k)-F(x_k)-J(x_k)p_k\|^2 \\
&\quad +\|F(x_k+p_k)-F(x_k)-J(x_k)p_k\|\|F(x_k)-y+J(x_k)p_k\|.
\end{aligned}$$

By (4.25) and the mean value theorem [85, p.630], it holds

$$\|F(x_k+p_k)-F(x_k)-J(x_k)p_k\| \le c\|p_k\|\|F(x_k+p_k)-F(x_k)\| \le c\kappa_J\|p_k\|^2.$$

Consequently, as $\Delta_k \le C_{\max}\|g_k\|$ and $\|F(x_k)-y+J(x_k)p_k\| \le \|F(x_k)-y\| \le \|F(x_0)-y\|$, it follows

$$\begin{aligned}
f(x_k+p_k)-m_k^{TR}(p_k) &\le \frac{1}{2}c^2\kappa_J^2\|p_k\|^4+c\kappa_J\|p_k\|^2\|F(x_k)-y+J(x_k)p_k\| \\
&\le \frac{1}{2}c\kappa_J\|p_k\|^2(c\kappa_J\|p_k\|^2+2\|F(x_0)-y\|) \\
&\le \frac{1}{2}c\kappa_J\Delta_k^2(c\kappa_J\Delta_k^2+2\|F(x_0)-y\|) \\
&\le \frac{1}{2}c\kappa_J\Delta_k^2(c\kappa_J C_{\max}^2\|g_k\|^2+2\|F(x_0)-y\|) \\
&\le \frac{1}{2}c\kappa_J\Delta_k^2(c\kappa_J C_{\max}^2\kappa_J^2\|F(x_0)-y\|^2+2\|F(x_0)-y\|) \\
&\le \frac{1}{2}c\kappa_J\Delta_k^2\|F(x_0)-y\|(c\kappa_J^3 C_{\max}^2\|F(x_0)-y\|+2).
\end{aligned}$$

Theorem 6.3.1 in [22] shows that

$$f(x_k)-m_k^{TR}(p_k) \ge \frac{1}{2}\|g_k\|\min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}.$$

Then,

$$f(x_k)-m_k^{TR}(p_k) \ge \frac{1}{2}\Delta_k\|g_k\|, \tag{4.33}$$

whenever $\Delta_k \le \dfrac{\|g_k\|}{\kappa_J^2}$ and this implies

$$1-\rho_k(p_k) \le \frac{c\kappa_J\Delta_k\|F(x_0)-y\|(c\kappa_J^3 C_{\max}^2\|F(x_0)-y\|+2)}{\|g_k\|}.$$

Namely, termination of the repeat loop occurs with

$$\Delta_k \le \|g_k\|\omega, \qquad \omega = \min\left\{\frac{1}{\kappa_J^2}, \frac{1-\eta}{c\kappa_J\|F(x_0)-y\|(c\kappa_J^3 C_{\max}^2\|F(x_0)-y\|+2)}\right\}. \tag{4.34}$$

Taking into account step 2 and the updating rule at step 3.3, we can conclude that, at termination of step 3, the Trust-Region radius Δ_k satisfies

$$\Delta_k \geq \min\{C_{\min}, \gamma\omega\} \|g_k\|.$$

Finally, by (4.31) λ_k is bounded above, as

$$\lambda_k \leq \frac{\|g_k\|}{\Delta_k} \leq \max\left\{\frac{1}{\gamma\omega}, \frac{1}{C_{\min}}\right\} = \bar{\lambda}. \tag{4.35}$$

□

As problem (4.1) is ill-posed, it may have more than one solution. Even if we have at disposal a starting guess close enough to $x^\dagger$ we will not be sure to converge exactly to that solution, but rather to a solution in its neighbourhood. Then, in the following theorem we employ the previous results to show convergence of the generated sequence to a solution $x^* \in B_r(x^\dagger)$.

Theorem 4.16. *Suppose that Assumptions 4.10 and 4.12 hold and $\delta = 0$. Then, the sequence $\{x_k\}$ generated by Algorithm 4.2 converges to a solution x^* of (4.1) such that $\|x^* - x^\dagger\| \leq r$.*

Proof. Let $\bar{k}$ as in Assumption 4.12 and $k \geq \bar{k}$. From Lemma 4.13 and Lemma 4.14 condition (4.28) holds with θ_k given in (4.30) and monotonically increasing. Then, an adaptation of the proof of Theorem 4.2 in [67] gives that $\{x_k\}$ is convergent. The proof is repeated for sake of clarity. Using (4.25) we obtain

$$\begin{aligned}\|J(x_i)(x_k - x^\dagger)\| &\leq \|F(x_i) - F(x_k) - J(x_i)(x_i - x_k)\| + \|F(x_i) - F(x_k) + J(x_i)(x_i - x^\dagger)\| \\ &\leq c\|x_i - x_k\|\|F(x_i) - F(x_k)\| + \|F(x_k) - y\| + c\|x_i - x^\dagger\|\|F(x_i) - y\|.\end{aligned}$$

From the monotonicity property of Trust-Region methods, $\|F(x_i) - y\| \geq \|F(x_k) - y\|$ as $k \geq i$. Moreover, if we set $\sigma = c\|x_{\bar{k}} - x^\dagger\|$, from Lemma 4.15 we have $\sigma \geq c\|x_i - x^\dagger\|$ for all $i \geq \bar{k}$. Then, for all $k \geq i \geq \bar{k}$ it holds $\|F(x_i) - F(x_k)\| \leq 2\|F(x_i) - y\|$ and $c\|x_i - x_k\| \leq 2\sigma$, so that

$$\|J(x_i)(x_k - x^\dagger)\| \leq (5\sigma + 1)\|F(x_i) - y\|.$$

Letting $e_k = x_k - x^\dagger$, using this result and Lemma 2.18, we obtain that for $k > j \geq \bar{k}$:

$$\begin{aligned}|<e_j - e_k, e_k>| &= \left|\sum_{i=j}^{k-1} <p_i, e_k>\right| = \left|\sum_{i=j}^{k-1} <(J(x_i)J(x_i)^T + \lambda_i I)^{-1})(y - F(x_i)), J(x_i)e_k>\right| \\ &\leq \sum_{i=j}^{k-1} \|(J(x_i)J(x_i)^T + \lambda_i I)^{-1}(y - F(x_i))\|\|J(x_i)e_k\| \\ &\leq (1 + 5\sigma)\sum_{i=j}^{k-1} \frac{1}{\lambda_i}\|F(x_i) - y + J(x_i)(x_{i+1} - x_i)\|\|F(x_i) - y\|.\end{aligned}$$

Thus, (4.8) and (4.28) yield

$$\begin{aligned}|<e_j - e_k, e_k>| &\leq (1 + 5\sigma)\sum_{i=j}^{k-1} \frac{1}{\lambda_i q}\|F(x_i) - y + J(x_i)(x_{i+1} - x_i)\|^2 \\ &\leq \alpha_{\bar{k}}(\|e_j\|^2 - \|e_k\|^2),\end{aligned} \tag{4.36}$$

where $\alpha_{\bar{k}} = \dfrac{(1+5\sigma)\theta_{\bar{k}}}{2q(\theta_{\bar{k}}-1)}$ and we have used $\theta_k/(\theta_k - 1) < \theta_{\bar{k}}/(\theta_{\bar{k}} - 1)$ since the function $\theta/(\theta-1)$ is monotonically decreasing, (cf. Lemma 4.15). Then

$$\|x_k - x_j\|^2 = 2 < e_k - e_j, e_k > + \|e_j\|^2 - \|e_k\|^2 \leq (2\alpha_{\bar{k}} + 1)(\|e_j\|^2 - \|e_k\|^2).$$

Since the sequence $\{\|e_k\|\}$ is bounded from below and monotonically decreasing, hence convergent, it follows that $\{x_k\}$ is a Cauchy sequence, i.e. $\{x_k\}$ converges to a limit point x^*. By $x_k \in B_r(x^\dagger)$ for $k \geq \bar{k}$, it follows $\|x^* - x^\dagger\| \leq r$.

Finally, from Lemma 4.15 we know that $\lambda_k \leq \bar{\lambda}$ and $(\theta_k - 1)/\theta_k > (\theta_{\bar{k}} - 1)/\theta_{\bar{k}}$, for $k \geq \bar{k}$ since the function $(\theta-1)/\theta$ is monotonically increasing. Then, by (4.28) and (4.8)

$$\|x_k - x^\dagger\|^2 - \|x_{k+1} - x^\dagger\|^2 \geq \frac{2(\theta_{\bar{k}} - 1)q^2}{\theta_{\bar{k}}\bar{\lambda}} \|F(x_k) - y\|^2.$$

Thus, we conclude that $\|F(x_k) - y\|$ tends to zero and the limit x^* of x_k solves (4.1).

□

4.3.2 Noisy case

Results similar to those presented in the previous section for the noise free case can be given for the noisy case. All these results will hold for $\bar{k} \leq k < k_*(\delta)$, with $k_*(\delta)$ defined in (4.7).

First we prove that condition (4.15) is satisfied under Assumption 4.12.

Lemma 4.17. *Suppose that* $\delta > 0$ *and Assumptions 4.10 and 4.12 hold. Then,* (4.15) *is satisfied for* $\bar{k} \leq k < k_*(\delta)$.

Proof. By (4.25) and (3.2) we obtain

$$\begin{aligned}
\|y^\delta - F(x^\delta_{\bar{k}}) - J(x^\delta_{\bar{k}})(x^\dagger - x^\delta_{\bar{k}})\| &\leq \delta + \|y - F(x^\delta_{\bar{k}}) - J(x^\delta_{\bar{k}})(x^\dagger - x^\delta_{\bar{k}})\| \\
&\leq \delta + c\|x^\dagger - x^\delta_{\bar{k}}\| \, \|y - F(x^\delta_{\bar{k}})\| \\
&\leq (1 + c\|x^\dagger - x^\delta_{\bar{k}}\|)\delta + c\|x^\dagger - x^\delta_{\bar{k}}\| \, \|y^\delta - F(x^\delta_{\bar{k}})\|.
\end{aligned}$$

Then, at iteration $\bar{k}$, condition (4.7) gives

$$\|y^\delta - F(x^\delta_{\bar{k}}) - J(x^\delta_{\bar{k}})(x^\dagger - x^\delta_{\bar{k}})\| \leq \left(\frac{1 + c\|x^\dagger - x^\delta_{\bar{k}}\|}{\tau} + c\|x^\dagger - x^\delta_{\bar{k}}\| \right) \|y^\delta - F(x^\delta_{\bar{k}})\|,$$

so that (4.15) is satisfied at $k = \bar{k}$ with $\theta_{\bar{k}} = \dfrac{q\tau}{1 + c(1+\tau)\|x^\dagger - x^\delta_{\bar{k}}\|}$, and $\theta_{\bar{k}} > 1$ from (4.27). Further, by Lemma 4.13 condition (4.28) is satisfied with $\theta_k = \theta_{\bar{k}}$, and this implies $\|x^\delta_{\bar{k}+1} - x^\dagger\| < \|x^\delta_{\bar{k}} - x^\dagger\|$. Repeating the above arguments, by induction we can prove that, for $\bar{k} < k < k_*(\delta)$, condition (4.15) holds and (4.28) is satisfied with

$$\theta_k = \frac{q\tau}{1 + c(1+\tau)\|x^\dagger - x^\delta_k\|} > 1. \tag{4.37}$$

□

In the following lemma we prove that for $\bar{k} \leq k < k_*(\delta)$, where $k_*(\delta)$ is defined in (4.7), the error is decreasing and the scalars $\lambda_k > 0$ are bounded above.

Lemma 4.18. *Suppose that $\delta > 0$ and Assumptions 4.10 and 4.12 hold. Then, Algorithm 4.2 generates a sequence x_k^δ such that, for $\bar{k} \leq k < k_*(\delta)$,*

(i) x_k^δ belongs to $B_{2r}(x_{\bar{k}}^\delta) \cap B_r(x^\dagger)$, $\|x_{k+1}^\delta - x^\dagger\| < \|x_k^\delta - x^\dagger\|$, $\theta_{k+1} > \theta_k$.

(ii) There exists a constant $\bar{\lambda} > 0$ such that $\lambda_k \leq \bar{\lambda}$.

Proof. *(i)* By Lemma 4.17 condition (4.15) is satisfied for $\bar{k} \leq k \leq k_*(\delta)$, so (4.28) also holds for $\bar{k} \leq k \leq k_*(\delta)$. Then, $\{\|x_k^\delta - x^\dagger\|\}_{k=\bar{k}}^{k_*(\delta)}$ is monotonically decreasing and consequently x_k^δ belongs to $B_{2r}(x_{\bar{k}}^\delta) \cap B_r(x^\dagger)$ for all $\bar{k} \leq k \leq k_*(\delta)$. Notice also that as a consequence $\theta_{k+1} > \theta_k$ for $\bar{k} \leq k < k_*(\delta)$, as θ_k is defined as in (4.37).

(ii) Proceeding as in the proof of point *(iii)* of Lemma 4.15, just replacing x_k with x_k^δ, we obtain that for $\bar{k} \leq k < k_*(\delta)$, $\lambda_k < \bar{\lambda}$ with $\bar{\lambda}$ defined in (4.35) in which ω is obtained replacing y with y^δ in (4.34). □

In the following theorem we prove the regularizing properties of the method, showing that the error decreases monotonically and the sequence $\{x_{k_*(\delta)}^\delta\}$ converges to a solution of (4.1) as δ tends to zero.

We underline that the iterates generated by Algorithm 4.2 depend continuously on y^δ, as they are obtained by linear operations, if the Trust-Region radius Δ_k, and consequently the scalar λ_k implicitly defined by the Trust-Region problem, depend continuously on δ. Continuous dependence of Δ_k on the noise is related to $\rho_k(x_{k+1} - x_k)$, where $\rho_k(p_k)$ is defined at step 3.2 of Algorithm 4.2. To ensure such continuity we have to assume that $\rho_k(x_{k+1} - x_k) \neq \eta$ for all $k \geq 0$. In fact, this represents an adverse case, as in case there exists an index k such that $\rho_k(x_{k+1} - x_k) = \eta$, we cannot be sure that $\rho_k(x_{k+1}^\delta - x_k^\delta) \geq \eta$, even for small δ, and therefore Δ_k will not depend continuously on δ. On the other hand if $\rho_k(x_{k+1} - x_k) \neq \eta$, $\rho_k(x_{k+1}^\delta - x_k^\delta) - \eta$ will have the same sign of $\rho_k(x_{k+1} - x_k) - \eta$ and Δ_k will depend continuously on δ. This feature is crucial for proving that the sequence $\{x_{k_*(\delta)}^\delta\}$ tends to a solution of (4.1) as δ tends to zero. For this reason $\rho_k(x_{k+1} - x_k) \neq \eta$ is assumed in the following theorem.

Theorem 4.19. *Suppose that Assumptions 4.10 and 4.12 hold and let $\delta \geq 0$. Then,*

(i) the iterates generated by Algorithm 4.2 satisfy the stopping criterion (4.7) after a finite number $k_*(\delta)$ of iterations.

(ii) Suppose moreover that the sequence $\{x_k\}$ generated with the exact data y satisfies $\rho_k(x_{k+1} - x_k) \neq \eta$, for all k. Then, the sequence $\{x_{k_*(\delta)}^\delta\}$ converges to a solution of (4.1) whenever δ tends to zero.

Proof. (i) Summing up from $\bar{k}$ to $k_*(\delta) - 1$, by (4.7), (4.8), (4.28) and Lemma 4.18, it follows

$$(k_*(\delta) - \bar{k})\tau^2\delta^2 \leq \sum_{k=\bar{k}}^{k_*(\delta)-1} \|F(x_k^\delta) - y^\delta\|^2 \leq \frac{\theta_{\bar{k}}\bar{\lambda}}{2(\theta_{\bar{k}} - 1)q^2}\|x_{\bar{k}}^\delta - x^\dagger\|^2.$$

Thus, $k_*(\delta)$ is finite for $\delta > 0$.

(ii) The thesis is obtained by adapting the proof of [50, Theorem 2.3]. Specifically, let x^* be the limit of the sequence $\{x_k\}$ corresponding to the exact data y and let $\{\delta_n\}$ be a sequence of values of δ converging to zero as $n \to \infty$. Denote by y^{δ_n} a corresponding sequence of perturbed data, and by $k_n = k_*(\delta_n)$ the stopping index determined from the discrepancy principle (4.7) applied with $\delta = \delta_n$. Assume first that $\tilde{k}$ is a finite accumulation point of $\{k_n\}$. Without loss of generality, possibly considering a subsequence of $\{\delta_n\}$, we can assume that $k_n = \tilde{k}$ for all $n \in \mathbb{N}$. Thus, from the definition of k_n it follows that

$$\|y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n})\| \leq \tau\delta_n. \tag{4.38}$$

By assumption, $\rho_k(x_{k+1} - x_k) \neq \eta$, for all k, it follows that for the fixed index $\tilde{k}$, the iterate $x_{\tilde{k}}^{\delta}$ depends continuously on δ. Then

$$x_{\tilde{k}}^{\delta_n} \to x_{\tilde{k}}, \qquad F(x_{\tilde{k}}^{\delta_n}) \to F(x_{\tilde{k}}) \qquad \text{as } \delta_n \to 0. \tag{4.39}$$

Therefore, by (4.38), it follows that the $\tilde{k}$-th iterate with exact data y is a solution of $F(x) = y$, i.e. $x^* = x_{\tilde{k}}$, and we can conclude that $x_{k_n}^{\delta_n} \to x^*$ as $\delta_n \to 0$.

It remains to consider the case where $k_n \to \infty$ as $n \to \infty$. As $\{x_k\}$ converges to a solution x^* of (4.1) by Theorem 4.16, there exists $\tilde{k} > 0$ such that

$$\|x_k - x^*\| \leq \frac{1}{2}\bar{r} \qquad \text{for all} \qquad k \geq \tilde{k},$$

where $\bar{r} < \min\left\{\dfrac{q\tau - 1}{c(1+\tau)}, r\right\}$. Then, as x_k^{δ} depends continuously on δ, δ_n tends to zero and $k_*(\delta_n) \to \infty$, there exists δ_n sufficiently small such that $\tilde{k} \leq k_*(\delta_n)$ and

$$\|x_{\tilde{k}}^{\delta_n} - x_{\tilde{k}}\| \leq \frac{1}{2}\bar{r}.$$

Then, for δ_n sufficiently small

$$\|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \|x_{\tilde{k}}^{\delta_n} - x_{\tilde{k}}\| + \|x_{\tilde{k}} - x^*\| \leq \bar{r}. \tag{4.40}$$

Now, from item (i) of Lemma 4.18, it holds $x_{\tilde{k}}^{\delta_n} \in B_{2r}(x_{\tilde{k}}^{\delta_n})$, while from (4.27) and Theorem 4.16 it holds $x^* \in B_{2r}(x_{\tilde{k}}^{\delta_n})$ as

$$\|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \|x_{\tilde{k}}^{\delta_n} - x^\dagger\| + \|x^\dagger - x^*\| \leq 2r.$$

Repeating arguments in Lemma 4.17, we use (4.25), (3.2) and (4.7) and obtain

$$\begin{aligned}
\|y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n}) - J(x_{\tilde{k}}^{\delta_n})(x^* - x_{\tilde{k}}^{\delta_n})\| &\leq \delta_n + \|y - F(x_{\tilde{k}}^{\delta_n}) - J(x_{\tilde{k}}^{\delta_n})(x^* - x_{\tilde{k}}^{\delta_n})\| \\
&\leq \delta_n + c\|x^* - x_{\tilde{k}}^{\delta_n}\|\,\|y - F(x_{\tilde{k}}^{\delta_n})\| \\
&\leq (1 + c\|x^* - x_{\tilde{k}}^{\delta_n}\|)\delta + c\|x^* - x_{\tilde{k}}^{\delta_n}\|\,\|y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n})\| \\
&\leq \left(\frac{1 + c\|x^* - x_{\tilde{k}}^{\delta_n}\|}{\tau} + c\|x^* - x_{\tilde{k}}^{\delta_n}\|\right)\|y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n})\|.
\end{aligned}$$

Thus, by (4.40) and $\bar{r} < \min\left\{\dfrac{q\tau-1}{c(1+\tau)}, r\right\}$, it follows that the following counterpart of (4.15)

$$\|F(x_{\tilde{k}}^{\delta}) - y^{\delta} + J(x_{\tilde{k}}^{\delta})(x^* - x_{\tilde{k}}^{\delta})\| \leq \frac{q}{\theta_k}\|F(x_{\tilde{k}}^{\delta}) - y^{\delta}\|$$

is satisfied at $k = \tilde{k}$ with $\theta_{\tilde{k}} = \dfrac{q\tau}{1+c(1+\tau)\bar{r}} > 1$. Replacing $x^{\dagger}$ with x^*, (4.28) gives $\|x_{\tilde{k}+1}^{\delta_n} - x^*\| < \|x_{\tilde{k}}^{\delta_n} - x^*\|$ and repeating the above arguments, by induction we obtain monotonicity of the error $\|x_k^{\delta_n} - x^*\|$ for $\tilde{k} \leq k \leq k_n$. Then

$$\|x_{k_n}^{\delta_n} - x^*\| < \|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \bar{r}.$$

Finally, since the previous arguments can be repeated for any positive $\epsilon \leq \bar{r}$, provided that δ_n is small enough, we obtain that

$$x_{k_n}^{\delta_n} \to x^* \qquad \text{as} \qquad \delta_n \to 0.$$

□

In the following lemma, we show that, whenever the initial guess x_0 is sufficiently close to a solution of (4.1), it holds $\rho_k(x_{k+1} - x_k) > \eta$ and therefore the thesis of Theorem 4.19 holds. Then, the proposed Trust-Region approach shows the same local regularizing properties of the regularizing Levenberg-Marquardt method.

Lemma 4.20. *Suppose that Assumptions 4.1, 4.10 and 4.4 hold and $\delta = 0$. If x_0 is sufficiently close to a solution of (4.1), then $\rho_k(x_{k+1} - x_k) > \eta$ for $k \geq 0$.*

Proof. Theorem 4.16 implies that $\{x_k\}$ converges to a solution of (4.1). Using (4.32)–(4.33) and $\|p_k\| \leq \Delta_k$, it follows

$$1 - \rho_k(p_k) \leq \frac{\frac{1}{2}c\kappa_J\Delta_k^2(c\kappa_J\Delta_k^2 + \|F(x_k) - y\|)}{\frac{1}{2}\Delta_k\|g_k\|} = \frac{c\kappa_J\Delta_k(c\kappa_J\Delta_k^2 + \|F(x_k) - y\|)}{\|g_k\|},$$

while $\Delta_k \leq C_{\max}\|g_k\| \leq C_{\max}\kappa_J\|F(x_k) - y\|$ implies

$$1 - \rho_k(p_k) \leq c\kappa_J C_{\max}(c\kappa_J\Delta_k^2 + \|F(x_k) - y\|).$$

By the convergence of $\{x_k\}$ to a solution of (4.1) and Assumption 4.10 the right-hand side of the above inequality tends to zero. Hence, if x_0 is close enough to a solution of (4.1) it is ensured $1 - \rho_k(p_k) < 1 - \eta$, for $k \geq 0$. □

4.4 Numerical results

In this section we report on the numerical performance of the proposed approach, that we are going to address as the *regularizing Trust-Region* method. Specifically, it is compared to the regularizing Levenberg-Marquardt method proposed by M. Hanke [50], and we show the improved robustness of the Trust-Region implementation. Moreover, we consider also a standard version of the Trust-Region method. We show that an ad hoc choice of the radius is necessary to gain regularizing properties and that the standard update does not provide them.

4.4.1 Test problems definition

The test problems arise from the discretization of nonlinear Fredholm integral equations of the first kind:

$$\int_0^1 k(t,s,x(s))ds = y(t), \quad t \in [0,1], \tag{4.41}$$

that model inverse problems from groundwater hydrology and geophysics. These problems are known to be ill-posed in the infinite dimensional setting [105, 109]. The discrete counterparts inherit such feature.

We consider four test problems, for different choices of the kernel $k(t,s,x(s))$. Specifically two problems with kernel

$$k(t,s,x(s)) = \log\left(\frac{(t-s)^2+H^2}{(t-s)^2+(H-x(s))^2}\right), \tag{4.42}$$

[104, §3], and two problems with kernel

$$k(t,s,x(s)) = \frac{1}{\sqrt{1+(t-s)^2+x(s)^2}}, \tag{4.43}$$

[66, §6] are considered. The problems are built so that solutions (later denoted as true solutions) are known:

- P1 [104, p. 46]: kernel (4.42) is considered with $H = 0.2$. Problem (4.41) admits as true continuous solutions the functions $x_{true}(s) = c_1 e^{d_1(s+p_1)^2} + c_2 e^{d_2(s-p_2)^2} + c_3 + c_4$ and $x_{true}(s) = 2H - c_1 e^{d_1(s+p_1)^2} - c_2 e^{d_2(s-p_2)^2} - c_3 - c_4$, $s \in [0,1]$, where $c_1 = -0.1, c_2 = -0.075, d_1 = -40, d_2 = -60, p_1 = 0.4, p_2 = 0.67, c_3$ and c_4 are chosen such that $x_{true}(0) = x_{true}(1) = 0$.

- P2 [107, p. 835]: kernel (4.42) is considered with $H = 0.1$. Problem (4.41) has true continuous solutions $x_{true}(s) = 1.3s(1-s)+0.2$ and $x_{true}(s) = 1.3s(s-1)$, $s \in [0,1]$.

- P3 [66, p. 660]: kernel (4.43) is considered and the solutions are $x_{true}(s) = 1$ and $x_{true}(s) = -1$, $s \in [0,1]$.

- P4 [66, p. 662]: kernel (4.43) is considered and the solutions of (4.41) are given by the discontinuous functions

  $$x_{true}(s) = \begin{cases} 1 & \text{if } 0 \le s \le \frac{1}{2} \\ 0 & \text{if } \frac{1}{2} < s \le 1 \end{cases}, \quad x_{true}(s) = \begin{cases} -1 & \text{if } 0 \le s \le \frac{1}{2} \\ 0 & \text{if } \frac{1}{2} < s \le 1 \end{cases}$$

  The case of discontinuous solutions is of interest especially in geophysical applications.

Problem (4.41) is discretized in the following way. The interval $[0,1]$ is discretized with $m = 100$ equidistant grid points $t_i = (i-1)h$, $h = 1/(m-1)$, $i = 1,\dots,m$. Function $x(s)$ is approximated from the n-dimensional subspace of $H_0^1(0,1) = \{u \in$

$L^2(0,1) | D^\alpha u \in L^2(0,1)\ \forall |\alpha| \le 1$ and $Tu = 0\}$, where T is the trace operator, spanned by standard piecewise linear functions. Specifically, we choose $n = 64$ and let $s_j = (j-1)h$, $h = 1/(n-1)$, $j = 1,\dots,n$, and look for an approximation $\hat{x}(s) = \sum_{j=1}^n \hat{x}_j \phi_j(s)$ where

$$\phi_1(s) = \begin{cases} \dfrac{s_2 - s}{h} & \text{if} \quad s_1 \le s \le s_2 \\ 0 & \text{otherwise} \end{cases}, \qquad \phi_n(s) = \begin{cases} \dfrac{s - s_{n-1}}{h} & \text{if} \quad s_{n-1} \le s \le s_n \\ 0 & \text{otherwise} \end{cases},$$

and

$$\phi_j(s) = \begin{cases} \dfrac{s - s_{j-1}}{h} & \text{if} \quad s_{j-1} \le s \le s_j, \\ \dfrac{s_{j+1} - s}{h} & \text{if} \quad s_j \le s \le s_{j+1}, \\ 0 & \text{otherwise} \end{cases} \qquad j = 2, \dots n-1. \tag{4.44}$$

Finally, the integrals $\int_0^1 k(t_i, s, \hat{x}(s))ds$, $1 \le i \le m$, are approximated by the composite rectangular rule on the points s_j, $1 \le j \le n$, i.e.

$$\int_0^1 K(t_i, s, \hat{x}(s))ds \sim h \sum_{j=1}^n K(t_i, s_j, \hat{x}(s_j)) \qquad i = 1, \dots, m.$$

The resulting discrete problems are nonlinear systems (4.1) of m equations with unknown $x = (\hat{x}_1, \dots, \hat{x}_n)^T$. We observe that $\hat{x}(s_j) = \hat{x}_j$; thus, the j-th component of x approximates a solution of (4.41) at s_j.

In the following, with P1, P2, P3 and P4 we will denote the nonlinear systems arising from the discretization of the above presented test problems. We will denote with $x^\dagger \in \mathbb{R}^n$ a solution of the discretized problems, computed with noise level $\delta = 0$.

4.4.2 Implementation of Algorithm 4.2

Let's now discuss the implementation of Algorithm 4.2. All procedures are implemented in MATLAB and run using MATLAB 2014b on an Intel Core(TM) i7-4510U 2.6 GHz, 8 GB RAM; the machine precision is $\epsilon_m \approx 2 \cdot 10^{-16}$.

Regarding the parameters setting, in Algorithm 4.2 it is set $\eta = \dfrac{1}{4}$, $\gamma = \dfrac{1}{6}$, $q = 1.1/\tau$, where $\tau = 1.5$ is the parameter in the discrepancy principle (4.7), that is used as the stopping criterion. Noisy data y^δ are considered. Given the noise level δ, they are obtained perturbing the exact data y by normally distributed values with mean 0 and variance δ^2, using the MATLAB function randn. The Algorithm is run setting $\Delta_k = \frac{1-q}{\|B_k\|}\|g_k\|$.

The exact Jacobian of the nonlinear function F obtained by the discretization of (4.41) is used. A maximum number of 300 iterations is allowed and a failure is declared if this limit is exceeded.

Numerically it emerged that the updating rule at step 2 of Algorithm 4.2, in accordance with the theory, provides the desired regularizing properties. As an example, in the upper part of Figure 4.3 we consider problem P1 with $\delta = 10^{-2}$. On the left we can see that the computed solution (dotted line) is a good approximation of the true solution approached (solid line). On the right the monotonic

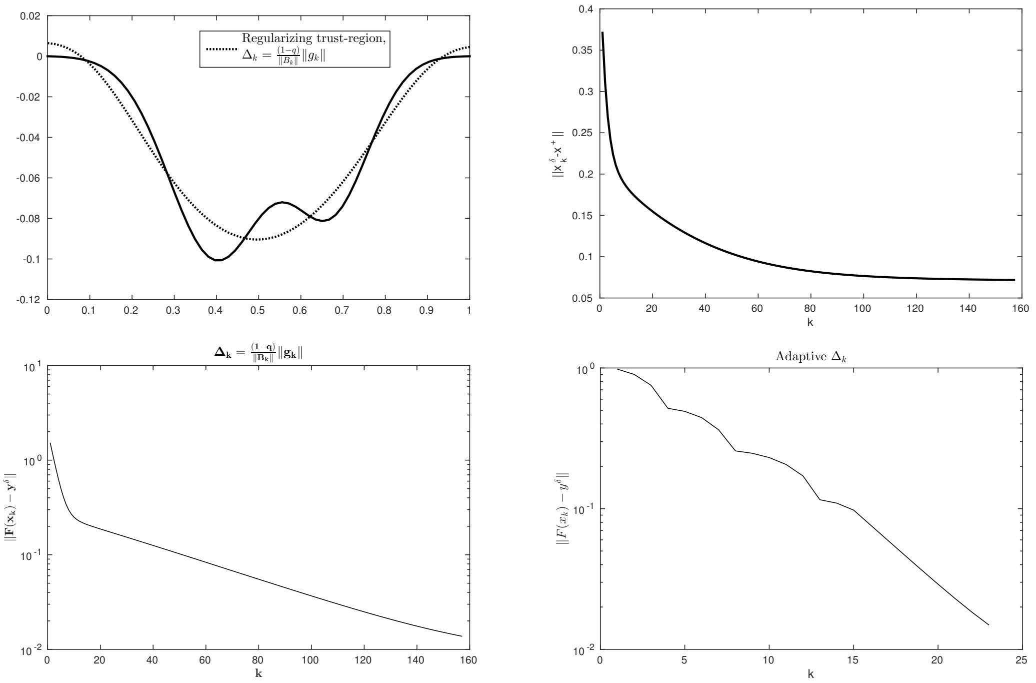


Figure 4.3: *Problem* `P1`, $x_0 = 0e$, $\delta = 10^{-2}$. *Upper part: Solution approximation computed with* $\Delta_k = \frac{1-q}{\|B_k\|}\|g_k\|$ *(dotted line) and true solution (solid line) (left) and monotonic decrease of the error* $\|x_k^\delta - x^\dagger\|$ *(right). Bottom part: Decrease of the residual* $\|F(x_k) - y^\delta\|$, *for* $\Delta_k = \frac{1-q}{\|B_k\|}\|g_k\|$ *(left) and adaptive choice of the Trust-Region radius* (4.45) *(right).*

reduction of the error is reported. However, the method results to be slow in practice. Then, for the numerical tests we employed an adaptive Trust-Region radius update. Namely we choose the starting Trust-Region radius as

$$\Delta_0 = \mu_0\|F(x_0) - y^\delta\|, \qquad\qquad \mu_0 = 10^{-1}.$$

Then, at step 2 of Algorithm 4.2 the radius is updated for the next iteration as follows:

$$\Delta_{k+1} = \mu_{k+1}\|F(x_{k+1}^\delta) - y^\delta\|, \qquad \mu_{k+1} = \begin{cases} \frac{1}{6}\mu_k & \text{if } q_k < q \\ 2\mu_k & \text{if } q_k > \nu q \\ \mu_k & \text{otherwise} \end{cases} \tag{4.45}$$

with $q_k = \dfrac{\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|}{||F(x_k^\delta) - y^\delta||}$, and $\nu = 1.1$. Maximum and minimum values for Δ_k are considered, and set to $\Delta_{\max} = 10^4$ and $\Delta_{\min} = 10^{-12}$. This updating strategy mimics the adaptive strategy used for well-posed problems (cf. Section 2.2), but the quantity that is adaptively updated is parameter μ_k rather than the radius Δ_k itself. Also, the decision of enlarging or shrinking the Trust Region is based on the fulfilment of the q-condition, rather than on the ratio between the actual and the predicted reduction. Indeed, with update (4.45) it is not guaranteed to have the q-condition satisfied, unlike in the case where Δ_k is chosen as in step 2 of Algorithm 4.2. Then, Δ_k is adjusted taking into account the q-condition by monitoring the value q_k. If the q-condition was not satisfied at the last computed iterate x_k^δ (i.e. if $q_k < q$) parameter μ_k is decreased to take a smaller radius, in the case where the q-condition was fulfilled (i.e. if $q_k \geq q$) parameter μ_k is increased or kept fixed. Moreover, the radius definition relates it to the norm of the residual, so that convergence to zero as the residual tends to zero is preserved. Notice also that Δ_k is cheaper to compute than the upper bound in (4.22), as one can spare the computation of $\|B_k\|$ and the estimation of the bound κ_J.

We compare as an example in the bottom part of Figure 4.3 the decrease of the residual $\|F(x_k) - y^\delta\|$ for the two choices of Trust-Radius radius strategy. We notice that in both cases the stopping criterion is satisfied, but with the adaptive choice the decrease is much faster. Then in all the tests presented in the following sections this updating strategy is used. We will see that it turns out to be efficient in practice.

At step 3 the KKT conditions (2.7) of the Trust-Region subproblem (4.21) are solved. Then, a couple $(\lambda_k, p(\lambda_k))$ satisfying (2.7) is looked for. As from the theory we know that $\lambda_k > 0$, we don't need to try the Newton step and we can directly solve $\|p(\lambda)\| = \Delta_k$. Then, as illustrated in Section 2.2.1, we apply Newton's method to equation (2.10). Each Newton's iteration requires the Cholesky factorization of a shifted matrix of the form $B_k + \lambda I$, see Section 2.2.1.1 and Algorithm 2.2. As we know that $\lambda > 0$, we are sure that the Cholesky factorizations will be reliable.

Typically high accuracy in the solution of the above scalar equations is not needed [9, 22] and this fact was experimentally verified also for our test problems. Hence, after extensive numerical experience, we decided to terminate the Newton's process as soon as $|\Delta_k - \|p(\lambda)\|| \leq 10^{-2}\Delta_k$ [22, §7.3.10].

4.4.3 Validation of the regularizing properties of the Trust-Region method

Our experiments are made varying the noise level δ on the data y^δ. Tables 4.1 and 4.2 display the results obtained by the regularizing Trust-Region algorithm with noise $\delta = 10^{-4}$ and $\delta = 10^{-2}$ respectively. Runs for four different initial guesses x_0 are reported in the tables. For problems `P1` and `P2` the initial guesses are $x_0 = 0e, -0.5e, -e, -2e$ and $x_0 = 0e, 0.5e, e, 2e$ respectively, where e denotes the vector $e = (1, \dots, 1)^T$. For problem `P3` the initial guess was chosen as the vector $x_0(\alpha)$ with j-th component given by $(x_0(\alpha))_j = g_\alpha(s_j)$ for $j = 1, \dots, n$, where $g_\alpha(s) = (-4\alpha + 4)s^2 + (4\alpha - 4)s + 1$, and s_j being the grid points in $[0, 1]$. We have chosen $\alpha = 1.25, 1.5, 1.75, 2$. For problem `P4` the initial guess $x_0(\beta, \chi)$ has components $(x_0(\beta, \chi))_j = g_{\beta,\chi}(s_j)$ for $j = 1, \dots, n$ with $g_{\beta,\chi} = \beta - \chi s$ and $(\beta, \chi) = (1, 1), (0.5, 0), (1.5, 1), (1.5, 0)$. In the tables we report: the initial guesses (for increasing distance from the true solutions) the number of outer iterations `it` performed by the Trust-Region method; the residual at the solution $\|F(x^\delta_{k_*(\delta)}) - y^\delta\|$; the number of function evaluations `nf` performed; the average number `cf` of Cholesky factorizations per nonlinear iteration. To assess the quality of the results obtained, we measured the distance between the final iterate $x^\delta_{k_*(\delta)}$ and the

Problem		RTR					RLM
	x_0	it	$\|F(x^\delta_{k_*(\delta)})-y^\delta\|$	nf	cf	e_T	e_T
P1	$0e$	47	1.5e−4	48	4.7	5.2e−3	4.3e−3
	$-0.5e$	55	1.2e−4	56	4.9	6.1e−2	6.3e−3
	$-1e$	58	1.4e−4	59	4.8	1.0e−2	1.1e−2
	$-2e$	63	1.5e−4	64	4.7	1.8e−2	1.5e−2
P2	$0e$	54	1.5e−4	55	5.1	1.4e−3	*
	$0.5e$	48	1.2e−4	49	5.1	3.2e−3	*
	$1e$	53	1.3e−4	54	4.9	6.3e−3	8.3e−3
	$2e$	59	1.2e−4	60	4.7	8.9e−3	4.8e−3
P3	$x_0(1.25)$	44	1.4e−4	45	3.4	9.1e−3	3.1e−3
	$x_0(1.5)$	47	1.4e−4	48	3.3	5.1e−2	6.2e−2
	$x_0(1.75)$	48	1.4e−4	49	3.3	3.2e−1	3.1e−1
	$x_0(2)$	66	1.4e−4	75	3.4	4.3e−1	3.8e−1
P4	$x_0(1,1)$	74	1.5e−4	86	3.2	4.6e−1	*
	$x_0(0.5,0)$	70	1.5e−4	84	3.3	4.8e−1	4.7e−1
	$x_0(1.5,1)$	78	1.4e−4	93	3.5	4.9e−1	4.8e−1
	$x_0(1.5,0)$	81	1.5e−4	93	3.6	6.6e−1	6.3e−1

Table 4.1: *Results obtained by the regularizing Trust-Region (RTR) method and the regularizing Levenberg-Marquardt (RLM) method with noise $\delta = 10^{-4}$ and different initial guesses.*

true solution approached. In particular e_T=$\max_{1\le j\le n}|x_{true}(s_j)-(x^\delta_{k_*(\delta)})_j|$ is the maximum absolute value of the difference between the components associated to points s_j.

Tables 4.1 and 4.2 show that the regularizing Trust-Region method solves all the tests. By step 3 of Algorithm 4.2, the difference between the number of function evaluations and the number of Trust-Region iterations, if greater than one, indicates the number of trial iterates that were rejected because a sufficient reduction on f_δ was not achieved. We observe that in 27 out of 32 runs, all the iterates generated were accepted. This occurrence seems to indicate that the Trust-Region updating rule works well in practice.

Problem		RTR					RLM
	x_0	it	$\Vert F(x^\delta_{k_*(\delta)}) - y^\delta \Vert$	nf	cf	e_T	e_T
P1	$0e$	23	1.5e−2	24	5.7	1.8e−2	1.8e−2
	$-0.5e$	33	1.5e−2	34	5.4	3.9e−2	3.6e−2
	$-1e$	34	1.2e−2	35	5.4	5.5e−2	5.5e−2
	$-2e$	37	1.3e−2	38	5.3	8.4e−2	6.7e−2
P2	$0e$	28	1.4e−2	29	5.5	7.1e−3	*
	$0.5e$	25	1.4e−2	26	5.6	3.1e−2	*
	$1e$	31	1.5e−2	32	5.6	6.7e−2	4.6e−2
	$2e$	36	1.4e−2	37	5.4	8.9e−2	6.7e−2
P3	$x_0(1.25)$	19	1.0e−2	20	4.5	1.5e−1	1.5e−1
	$x_0(1.5)$	22	1.1e−2	23	4.0	3.2e−1	3.2e−1
	$x_0(1.75)$	20	1.4e−2	21	4.5	5.0e−1	5.1e−1
	$x_0(2)$	22	1.3e−2	23	4.4	6.9e−1	7.0e−1
P4	$x_0(1,1)$	17	1.5e−2	18	4.5	5.6e−1	5.4e−1
	$x_0(0.5,0)$	18	1.4e−2	19	4.2	5.5e−1	*
	$x_0(1.5,1)$	24	1.2e−2	25	4.6	5.0e−1	5.0e−1
	$x_0(1.5,0)$	31	1.3e−2	32	4.4	8.4e−1	*

Table 4.2: *Results obtained by the regularizing Trust-Region (RTR) method and the regularizing Levenberg-Marquardt (RLM) method with noise $\delta = 10^{-2}$ and different initial guesses.*

Further insight on the Trust-Region updating rule (4.45) can be gained analyzing the regularizing properties of the implemented Trust-Region strategy. First, we verified numerically that, though not explicitly enforced, the q-condition is satisfied in most of the iterations. As an illustrative example, we consider problem P2 with $\delta = 10^{-4}$ and $x_0 = 0e$. In the left plot in Figure 4.4, we display the values $q_k = \dfrac{||F(x^\delta_k) - y^\delta + J(x^\delta_k)p_k||}{||F(x^\delta_k) - y^\delta||}$ versus the Trust-Region iterations, marked by an asterisk, and the fixed value $q = 1.1/\tau \approx 0.733$, depicted by a solid line. To have the q-condition satisfied the points q_k should stay above the solid line. We observe that this holds at most of the iterations.

The plot on the right of Figure 4.4 shows that the error between x^δ_k and the solution approached with exact data $x^\dagger$ decays monotonically through the iterations, which results to be in accordance with the theoretical results in Lemma 4.18. The regularizing properties of the implemented Trust-Region scheme are also shown in Figure 4.5 where, for each test problem we plot the error $||x^\delta_{k_*(\delta)} - x^\dagger||$ for decreasing noise levels. It is evident that, in accordance with the theory, the error decays as the noise level decreases.

4.4.4 Comparison with Levenberg-Marquardt method

Let us now compare the regularizing Trust-Region and Levenberg-Marquardt procedures. The Levenberg-Marquardt approach was implemented imposing condition (4.6) and solving (4.19) by Newton's method. The process is stopped as soon as $|\Vert F(x^\delta_k) - y^\delta + J(x^\delta_k)p(\lambda)\Vert - q\Vert F(x^\delta_k) - y^\delta\Vert| \le 10^{-5}$. If a solution of (4.6) does not exist, it is not clear in this approach how to choose λ_k. Then, we decided to take

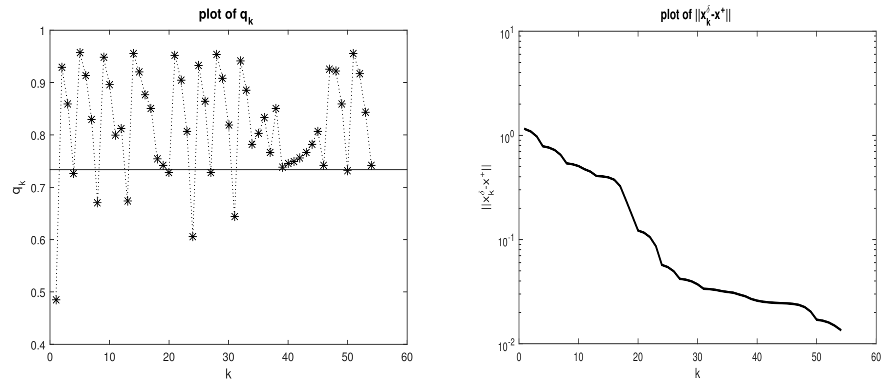


Figure 4.4: *Regularizing Trust-Region applied to* P2, $x_0 = 0e$, $\delta = 10^{-4}$. *Left: values* $q_k = \frac{||F(x_k^\delta)-y^\delta+J(x_k^\delta)p_k||}{||F(x_k^\delta)-y^\delta||}$ *(marked by an asterisk) and value of* $q = 1.1/\tau$ *(solid line) versus the iterations. Right: semilog plot of the error* $||x_k^\delta - x^\dagger||$ *versus the iterations (on the right).*

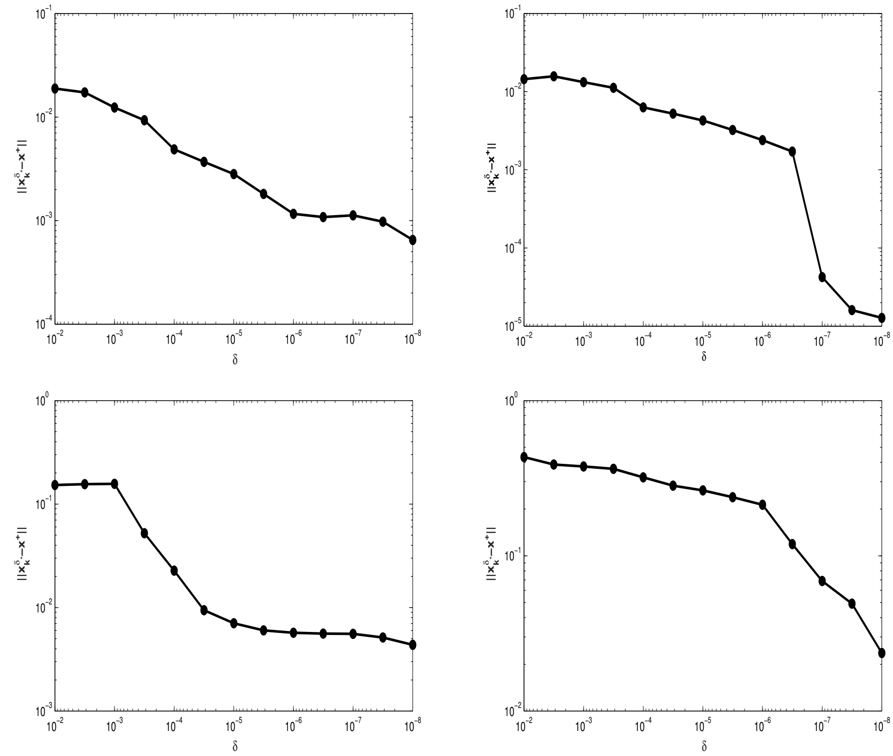


Figure 4.5: *Regularizing Trust-Region applied to* P1, $x_0 = 0e$ *(top left),* P2, $x_0 = 0e$ *(top right),* P3, $x_0 = x_0(\alpha) = x_0(1.25)$ *(lower left) and to* P4, $x_0 = x_0(\beta,\chi) = x_0(0.5,0)$ *(lower right). Log plot of the error* $||x_{k_*(\delta)}^\delta - x^\dagger||$ *versus the noise level* δ.

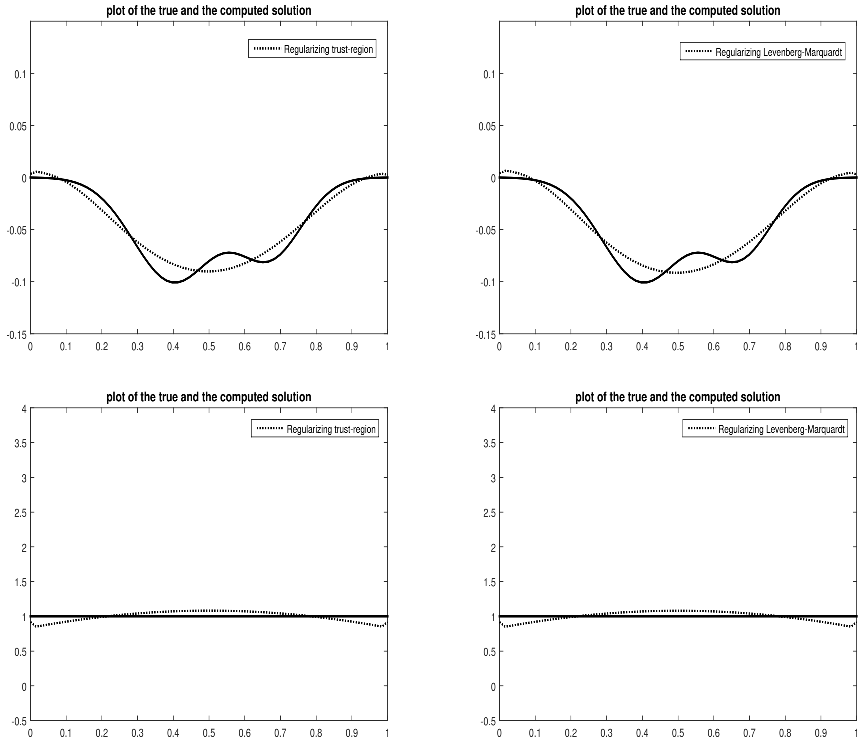


Figure 4.6: *Regularizing Trust-Region (left) and regularizing Levenberg-Marquardt (right), true solution (solid line) and approximate solutions (dotted line). Upper part:* `P1`*, $\delta = 10^{-2}$, $x_0 = 0e$. Lower part:* `P3`*, $\delta = 10^{-2}$, $x_0 = x_0(\alpha) = x_0(1.25)$.*

the last approximation computed by the Newton's method before a failure of the root-finding method is declared.

On runs that are successful for both methods, the two approaches show quite similar performance. The accuracy in the solution approximation increases when initial guesses close to the true solution are chosen. For all runs, the resulting error e_T for Levenberg-Marquardt method is reported in the last column of Tables 4.1, 4.2. As an example, Figure 4.6 compares the solutions computed by the two methods for problems `P1` and `P3` for $\delta = 10^{-2}$. It is evident that the two approaches provide solutions of similar accuracy.

On the other hand, symbols " $*$ " in the Tables, denote that in 7 runs out of 32 the Levenberg-Marquardt algorithm does not act as a regularizing method, as it is not able to provide a parameter ensuring regularizing properties. The implemented version of the method generates a sequence that approaches a solution of the noisy problem. In Figure 4.7 we show two unsuccessful runs of the Levenberg-Marquardt method. Approximated solutions computed by the regularizing Trust-Region and Levenberg-Marquardt procedures are compared for runs on problems `P2` and `P4`. While the Trust-Region method approximates solutions of the original problems in a stable way, the Levenberg-Marquardt converges to a noisy solution, that indeed presents the typical highly oscillatory behaviour.

The overall experience on the Levenberg-Marquardt algorithm seems to indicate that a method based on the q-condition is more robust than one based on

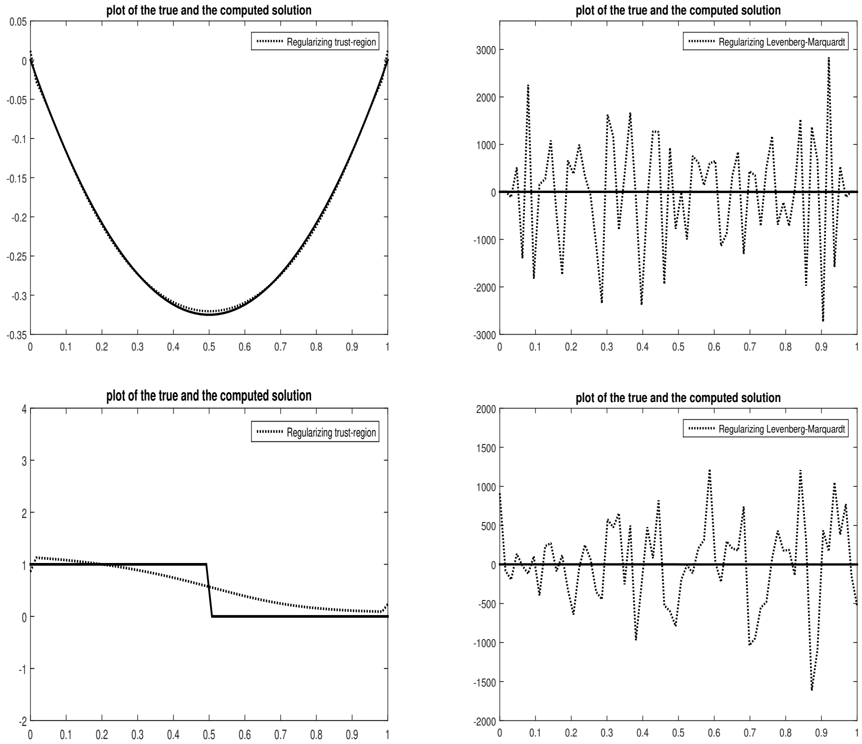


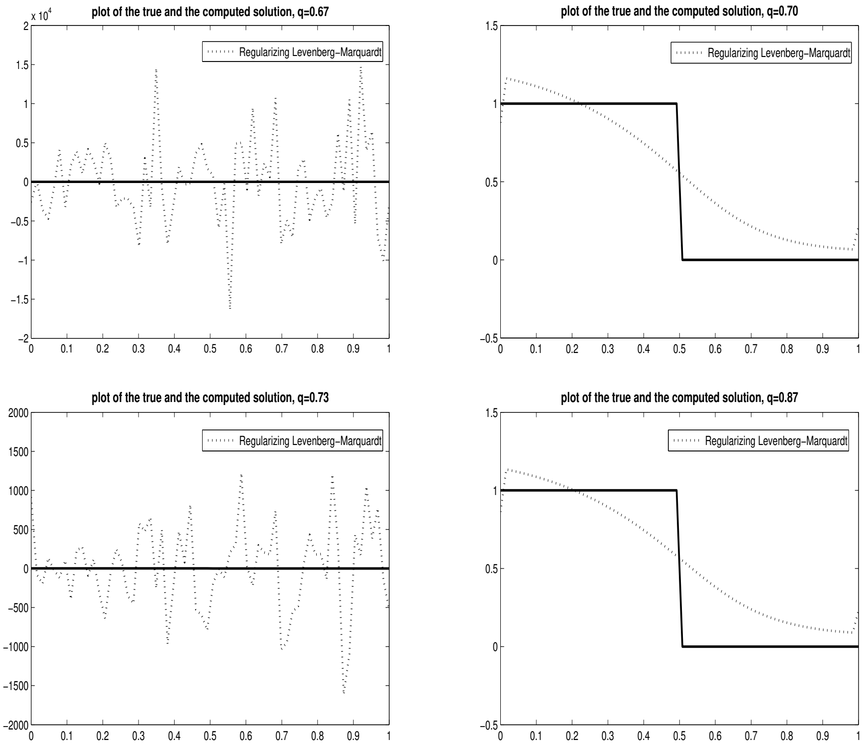
Figure 4.7: *True solution (solid line) and approximate solutions (dotted line) computed by the regularizing Trust-Region method (on the left) and the regularizing Levenberg-Marquardt method (on the right). Upper part: problem* `P2`*,* $\delta = 10^{-2}, x_0 = 0e$*; lower part: problem* `P4`*,* $\delta = 10^{-2}, x_0 = x_0(\beta,\chi) = x_0(0.5,0)$.

condition (4.6). The use of (4.8) also makes the method less dependent on parameter q, making its choice less important. In order to support this claim, in Figure 4.8 we report for problem `P4` and $\delta = 10^{-2}$ four solution approximations computed by the Levenberg-Marquardt algorithm for varying values of q, i.e. $q = 0.67, 0.70, 0.73, 0.87$. It is evident that the method is highly sensitive to the choice of the parameter q and the quality of the solution approximation does not steadily improve as q increases.

4.4.5 Comparison with standard Trust-Region method

We conclude this section considering the standard Trust-Region strategy. It is well-known that the standard updating rule promotes the use of inactive Trust Regions, at least in the latest stages of the procedure, cf. Remark 2.8. Clearly, this can adversely affect the solution of ill-posed problems as the fast convergence of Newton's method pushes the sequence to the noisy solution. Our experiments confirmed this fact.

In our implementation of the standard Trust-Region method, we chose the Trust-Region radius accordingly to technicalities well-known in the literature, see



plot of the true and the computed solution, q=0.67

plot of the true and the computed solution, q=0.70

plot of the true and the computed solution, q=0.73

plot of the true and the computed solution, q=0.87

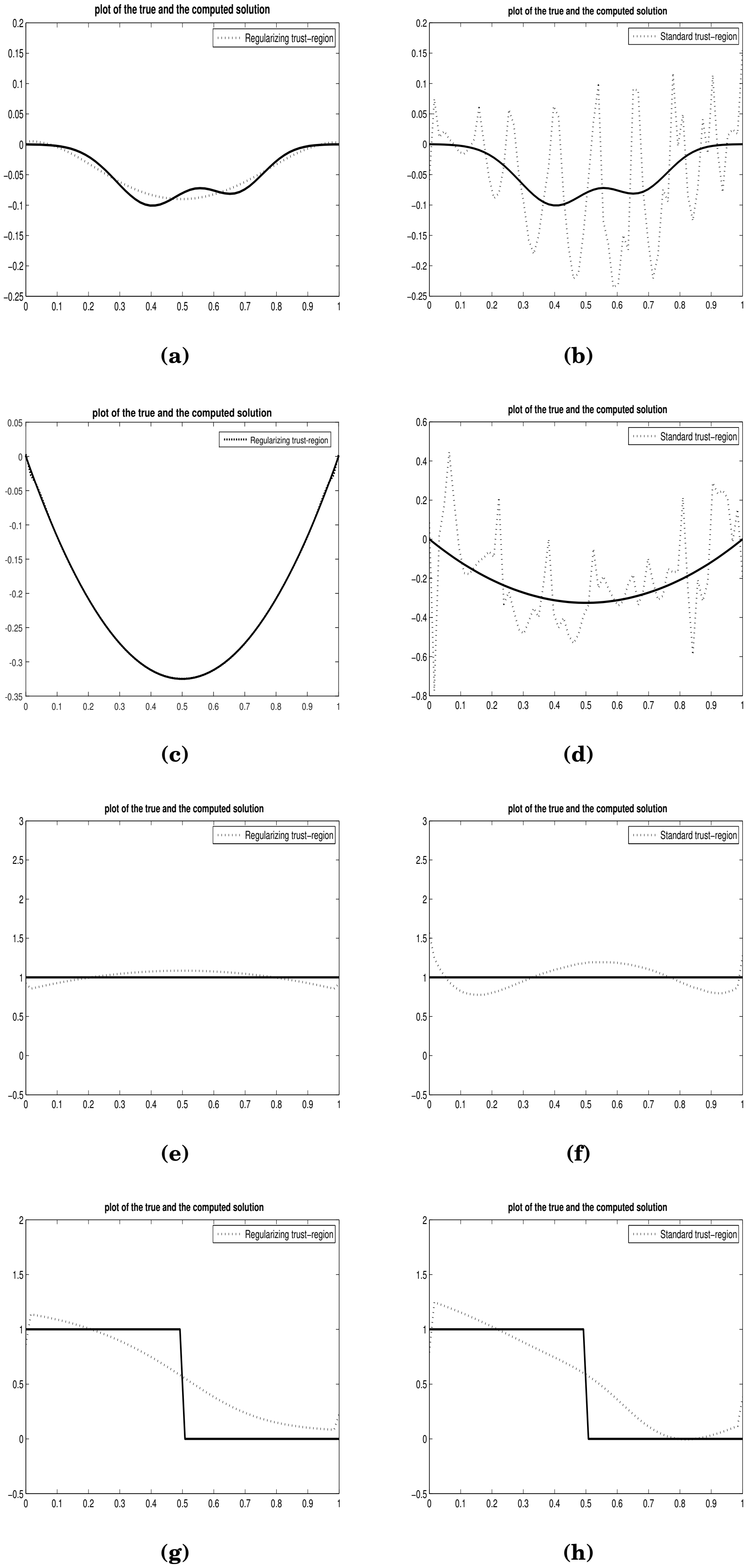
Figure 4.8: *Problem* P4*,* $\delta = 10^{-2}$*,* $x_0 = x_0(\beta,\chi) = x_0(1.5,0)$*. True solution (solid line) and approximate solution (dotted line) computed by the regularizing Levenberg-Marquardt method for values of* $q = 0.67, 0.70, 0.73, 0.87$.

Section 2.2 or [22, §6.1] and [85, Chapter 4]. In particular, we set $\Delta_0 = 1$,

$$\Delta_{k+1} = \begin{cases} \min\{2\Delta_k, \Delta_{\max}\} & \text{if } \rho_k(p_k) > \dfrac{3}{4}, \\ \Delta_k & \text{if } \dfrac{1}{4} \le \rho_k(p_k) \le \dfrac{3}{4}, \\ \dfrac{\|p_k\|}{4} & \text{if } \rho_k(p_k) < \dfrac{1}{4}, \end{cases} \tag{4.46}$$

with $\Delta_{\max} = 10^4$ and we chose $\Delta_{\min} = 10^{-12}$ as the minimum values for Δ_k.

For $\delta = 10^{-2}$ and problems P1 and P2, the sequences computed by the standard Trust-Region method approach solutions of the noisy problem. The same behaviour occurs in most of the runs with P1 and P2 and noise level $\delta = 10^{-4}$. Conversely, the approximations provided by the regularizing Trust-Region procedure are accurate approximations of true solutions in all the tests. The approximations computed by the standard Trust-Region method applied to problems P3 and P4 are less accurate than those computed by the regularizing Trust-Region method although they do not show the strong oscillatory behaviour arising in problems P1 and P2. In problem P4, this behaviour is evident when the second, third and fourth starting guesses are used, while the approximation computed starting from the first initial guess is as accurate as the one computed by the regularizing Trust-Region method. This good result of the standard Trust-Region approach on problem P4 with $x_0 = x_0(1,1)$ is due to the fact that the Trust Region is active in all iterations and therefore a regularizing behaviour is implicitly provided. As



(a) **(b)**

(c) **(d)**

(e) **(f)**

(g) **(h)**

Figure 4.9: *True solution (solid line) and approximate solutions (dotted line) computed by the regularizing Trust-Region method (on the left) and the standard Trust-Region method (on the right). (a)-(b) problem* P1*, $\delta = 10^{-2}$,$x_0 = 0e$; (c)-(d) problem* P2*, $\delta = 10^{-2}$,$x_0 = 0e$; (e)-(f) problem* P3*, $\delta = 10^{-2}$, $x_0 = x_0(1.25)$; (g)-(h) problem* P4*, $\delta = 10^{-2}$, $x_0 = x_0(0.5,0)$.*

an example in Figure 4.9 we compare some solution approximations computed by the regularizing Trust-Region (left) and by the standard Trust-Region (right) approaches with $\delta = 10^{-2}$ applied to problem P1 (figures (a)-(b)), P2 (figures (c)-(d)), P3 (figures (e)-(f)) and P4 (figures (g)-(h)).

4.4.6 Choice of the quadrature formula

In this section we motivate the choice we made for the quadrature formula to approximate the integrals. In the numerical experimentation we noticed that choosing a quadrature formula different from the rectangular rule, both the regularizing Trust-Region and Levenberg-Marquardt methods compute solutions that are good approximations of true solutions inside the interval, but they are misinterpreted at the end-points of the interval. The computed solution approximations present peaks at the end-points, that become higher and higher as the starting point moves away from the desired solution. In [77] the authors ascribe this behaviour to the quadrature formula, and assert that "it can be shown that with quadrature methods some of the singular vectors reflect the patterns in the weights, explaining why the solution has half its true value at the end points". In the numerical tests there are evidences of this behaviour. Using composite trapezoidal rule the first and the last components of the singular vectors of the Jacobian matrix associated to the largest singular values are about half the others, that have instead more or less the same magnitude. The same behaviour is reflected in the components of the resulting step. On the other hand, this does not happen if the rectangular rule is used.

With noisy data this behaviour is evident in all the iterations and causes higher and higher peaks as the noise increases or the starting guess moves farther from the sought solution. In the noise free case it is evident only in the firsts iterations, so that the solution approximations computed at the beginning of the process show peaks at the and points, which are then softened toward the end of the process.

In Figure 4.10 we report numerical evidence of this behaviour. We consider the noise free case in the upper part and $\delta = 10^{-2}$ in the lower part. Plots on the left refer to the rectangular rule and plots on the right to the trapezoidal rule. We report $e_I = \max_{2\leq j\leq n-1}|x_{true}(s_j) - (x^{\delta}_{k_*(\delta)})_j|$ the error corresponding to points inside the interval and $e_E = \max\{|x_{true}(s_1) - (x^{\delta}_{k_*(\delta)})_1|, |x_{true}(s_n) - (x^{\delta}_{k_*(\delta)})_n, |\}$ the error corresponding to the end-points of the interval. Let's consider first the trapezoidal rule. When noise is not present, at the beginning of the process the two error are different and there are peaks at the end points of the interval, but after few iterations e_E starts to decrease and toward the end of the process the solution approximation is good also at the end points of the interval. With noisy data it remains bigger than e_I for all the optimization process and the solution approximation shows peaks. With the rectangular rule both in case of noisy and exact data e_I and e_E have the same magnitude along all the optimization process.

Then, a quadrature formula with the same weights for all the nodes should be used. However, with the mid-point rule the resulting behaviour is the same as that of the trapezoidal rule, even if the weights satisfy the desired condition. This can be explained by the following analysis, that is done in the linear case for simplicity. We consider problem

$$\int_0^1 K(t,s)x(s)ds.$$

We consider the same setting as in Section 4.4.1, i.e for fixed n we define $s_j = (j-1)h$, $h = 1/(n-1)$, $j = 1,\dots,n$ and we approximate $x(s)$ with a piecewise linear function with nodes s_j: $\hat{x}(s) = \sum_{l=1}^{n} \hat{x}_l \phi_l(s)$. If, for fixed t, we approximate the resulting integral with the mid-point rule we obtain:

$$\int_0^1 K(t,s)\hat{x}(s)ds \simeq h\sum_{j=1}^{n-1} K\left(t, \frac{s_{j+1}+s_j}{2}\right)\hat{x}\left(\frac{s_{j+1}+s_j}{2}\right).$$

Taking into account that for all j, $\Phi_j(s) \neq 0$ in (s_{j-1}, s_{j+1}), we obtain

$$\hat{x}\left(\frac{s_{j+1}+s_j}{2}\right) = \sum_{l=1}^{n} \hat{x}_l \phi_l\left(\frac{s_{j+1}+s_j}{2}\right) = \frac{s_{j+1}-s_j}{2h}(\hat{x}_j + \hat{x}_{j+1}) = \frac{(\hat{x}_j + \hat{x}_{j+1})}{2}.$$

Then,

$$\int_0^1 K(t,s)\hat{x}(s)ds \simeq h\sum_{j=1}^{n-1} K\left(t, \frac{s_{j+1}+s_j}{2}\right)\frac{(\hat{x}_j + \hat{x}_{j+1})}{2}.$$

This implies that even if the weights are all the same, the coefficient of $\hat{x}_j$ varies with j and in particular the first and the last components of x have coefficients that are half the coefficients of the other components. This generates the same problem observed with the trapezoidal rule. This is due to the fact that nodes of the piecewise linear function do not coincide with the nodes of the quadrature formula. To avoid this, the rectangle rule should be used:

$$\int_0^1 K(t,s)x(s)ds \simeq h\sum_{j=1}^{n} K(t,s_j)\hat{x}_j, \tag{4.47}$$

so that all the components of x have the same weight. As we have seen in previous sections, with this rule the computed solution has indeed no peaks. We compare the solution approximations computed with the rectangle rule with those computed with the trapezoidal rule in Figure 4.11 for problems `P2` (upper part) and `P4` (lower part). In all the plots $\delta = 10^{-2}$. As expected, left plots (that correspond to the rectangular rule) do not show peaks, while evident peaks are present in solution approximations on the right (corresponding to the trapezoidal rule).

4.5 Chapter conclusion

In this section we have presented a Trust-Region method for nonlinear ill-posed systems with noisy data. A suitable Trust-Region radius choice is designed to

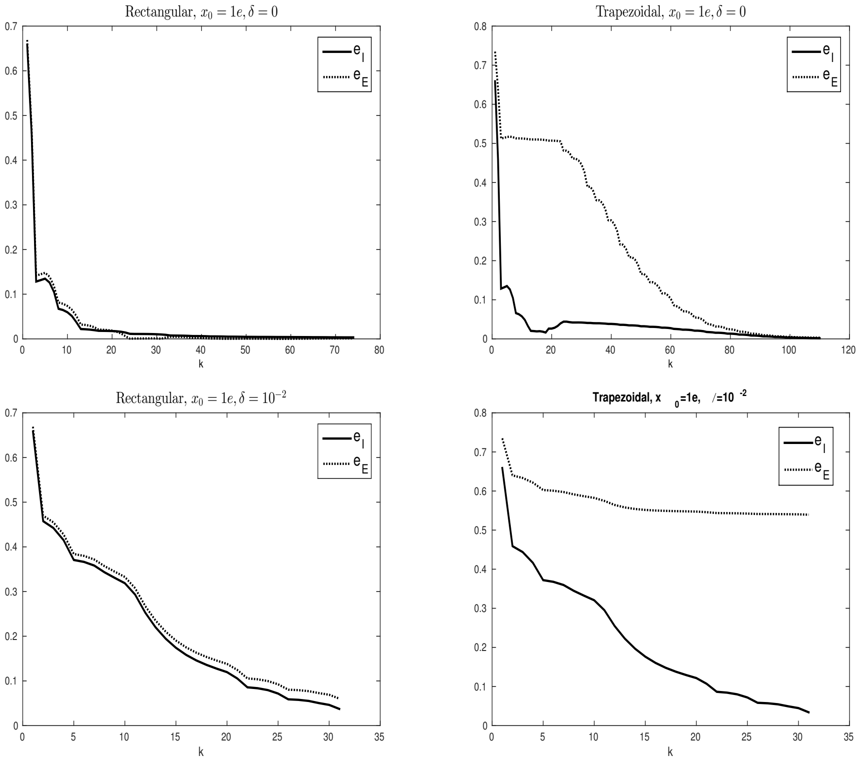


Figure 4.10: *Problem* P2*, $x_0 = 1e$, e_I error inside the interval, e_E error in the end-points. Upper part: $\delta = 0$; Lower part: $\delta = 10^{-2}$. Rectangular rule (left) and trapezoidal rule (right).*

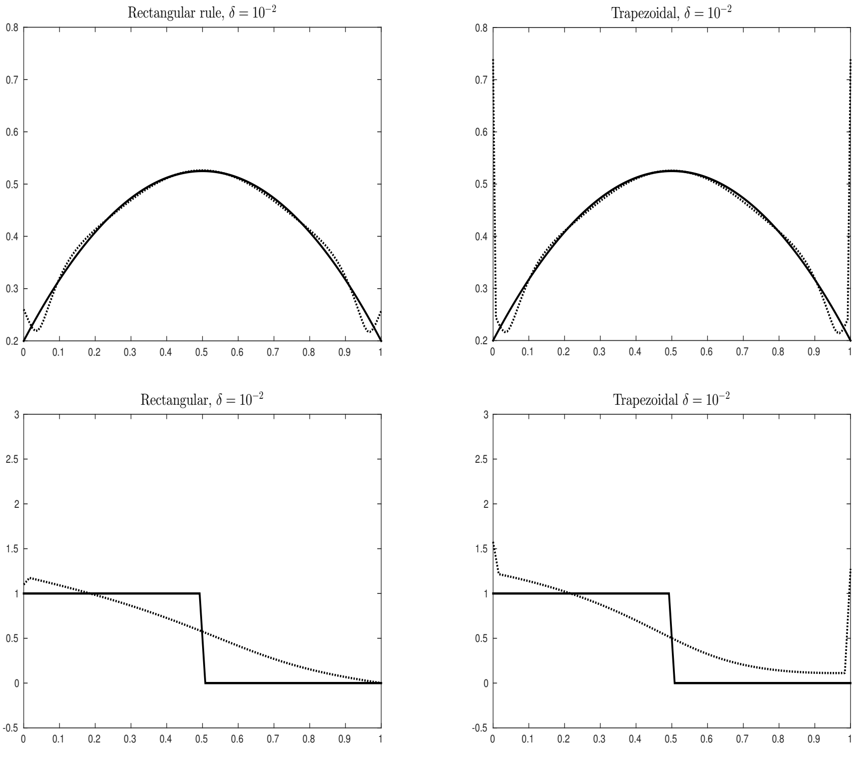


Figure 4.11: *Solution approximation and true solution, $\delta = 10^{-2}$. Upper part: problem* P2*; Lower part: problem* P4*. Rectangular rule (left) and trapezoidal rule (right).*

provide a regularizing behaviour to the method. The proposed approach shares the same local convergence properties as the regularizing Levenberg-Marquardt method proposed in [50]. Convergence properties are enhanced with respect to the regularizing Levenberg-Marquardt procedure in the following respects. With exact data, if there exists an accumulation point of the iterates which solves (4.1), then any accumulation point of the sequence solves (4.1). With noisy data it is more likely to satisfy the discrepancy principle irrespective of the closeness of the initial guess to a solution of (4.1). Indeed, condition (4.8) is well-defined even for initial guesses not close to a solution. Moreover, remarkably the new approach is shown to be less sensitive than the regularizing Levenberg-Marquardt method to the choice of the parameter q involved in the regularizations (4.6) and (4.8).

Our contribution covers theoretical and practical aspects of the method proposed. From a theoretical point of view, we propose the use of a Trust-Region radius converging to zero as k tends to zero. Trust-region methods with this distinguishing feature have been proposed in several papers, see [34, 36, 38, 113], but none of such works was either devised for ill-posed problems or applied to them. Thus, our study offers new insights on the potential of this choice. Finally, local convergence analysis has been carried out without assuming the invertibility of the Jacobian J of F or the boundedness of the inverse, which will commonly not hold in the presence of ill-posedness, but rather under weaker conditions, different form the local error-bound condition. Therefore, our results may represent a progress in the theoretical investigation of convergence of Trust-Region methods.

Concerning numerical aspects, we discussed an implementation of the regularizing Trust-Region method, and tested its ability to approximate a solution of (4.1) in presence of noise on four problems arising from the discretization of Fredholm integral equations of the first kind. Comparison with a standard Trust-Region scheme highlights the impact of the proposed Trust-Region radius choice on regularization, and confirms that the solution of noisy problems may be misinterpreted by the standard Trust-Region method. The numerical experience presented confirms the effectiveness of the Trust-Region radius adopted and the regularizing properties of the resulting Trust-Region method.

CHAPTER

5 Non-zero residual problems

In this chapter we consider nonlinear least squares problems of the form (3.1). Particularly, we focus on the general case in which it is not possible to assume the existence of x such that $F(x) = y$, so that the problem residual will be strictly positive, but we assume that a local minimum $x^\dagger$ exists. As in the previous chapter, we suppose to have only noisy data y^δ at disposal, such that (3.2) holds. Then, we have to deal with the noisy problem (3.3).

The aim of this chapter is to present a method for small residual problems that has the same regularizing properties as the methods we described in Chapter 4. More precisely, we look for a method that guarantees the following. In case of exact data, the sequence of gradients $\{\|J(x_k)^T(F(x_k)-y)\|\}$ should go to zero and the sequence of generated solution approximations should converge to a solution of (3.1). In case of noisy data, if an initial guess close to $x^\dagger$ is given, the method should have the potential to approach a solution of the unperturbed problem.

We are not aware of other methods, specially designed for nonzero residual ill-posed nonlinear least squares problems. We are aware only of [3], where convergence rates of Tikhonov methods are considered allowing also the case of nonzero residual. However the focus of that paper is on the study of the role of weak closedness of the operator in regularization theory. The method is studied only theoretically and a practical implementation is not discussed. Conditions on the regularization parameters are given to achieve regularizing properties, but a practical rule for selecting them is not provided. In [21] the authors study the properties of parameter identification problems formulated as least squares problems, like stability and the characterization of solutions, but they are not concerned with their numerical solution.

We consider a non-stationary iterated Tikhonov procedure [31, Chapter 10], [28], cf. also Section 2.4.4. At each iteration, given a positive parameter λ_k and a symmetric and positive definite regularization matrix $L_k \in \mathbb{R}^{n\times n}$ the step is computed solving the following regularized subproblem, given x_k^δ the current iterate:

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}\|F(x_k^\delta)-y^\delta + J(x_k^\delta)p\|^2 + \frac{\lambda_k}{2}\|L_k^{\frac{1}{2}}p\|^2. \tag{5.1}$$

As already explained in Section 2.4.4, the addition of matrix L_k in the Levenberg-

Marquardt model helps to improve the solution approximation [17, 29]. Here, we need it to provide regularizing properties also in presence of strictly positive residual. Then, with respect to the Levenberg-Marquardt described in the previous section, here we have to set the matrix along with the parameter λ_k.

Regularizing properties are ensured by the procedures described in the previous chapter thanks to two key ingredients:

1. the employment of a proper stopping criterion to avoid semiconvergence phenomenon, i.e. the method must be stopped before convergence is reached, to be sure that the generated sequence is not approaching a solution of the noisy problem,

2. a mechanism to control the step length, that needs to be not too large.

Regarding the first item, in the context of nonzero residual problems it not possible to use the discrepancy principle given in (4.7) and employed for the zero residual case, as $\|F(x_k^\delta)-y\|$ is not converging to zero anymore. This is not the quantity that we should look at to decide when we are close enough to a solution of (3.3). Then, inspiring by methods used for well-posed problems, we will rather control the gradient ∇f_δ of function f_δ as, if ∇f is the gradient of f, from (3.2) it holds:

$$\begin{aligned}\|\nabla f(x)\| =& \|J(x)^T(F(x)-y)\| = \|J(x)^T(F(x)-y\pm y^\delta)\| \leq \\ & \|J(x)^T(y-y^\delta)\| + \|\nabla f_\delta(x)\| \leq \|J(x)\|\delta + \|\nabla f_\delta(x)\|.\end{aligned}$$

Assuming the norm of the Jacobian to be bounded, if we check that the norm of the noisy gradient is less than the noise level, we can reasonably assume to have the same accuracy on the exact gradient. Then, in case of noisy data the process is stopped at iteration $k_*(\delta)$ satisfying the following discrepancy principle:

$$\|g_{k_*(\delta)}\| \leq \tau\delta < \|g_k\|, \quad 0 \leq k < k_*(\delta), \tag{5.2}$$

where we have used notation (3.6) and $\tau > 0$ is appropriately chosen.

Regarding item 2, in the methods described in the previous chapter, condition (4.6) or its improvement (4.8) were used to set the free regularization parameter λ, that determines the length of the step. Condition (4.8) relates the model for $F(x)-y$ at the current iterate to the norm of the residual. Here, we use an analogous condition, obtained considering the gradient and its model. Note that in case f is twice continuously differentiable, its Hessian matrix is given by:

$$\nabla^2 f(x) = J(x)^T J(x) + S(x) = J(x)^T J(x) + \sum_{j=1}^{m} (F(x)-y)_j \nabla^2 F_j(x). \tag{5.3}$$

Then, a first order model for the gradient in x would be given by

$$\nabla f(x) + \nabla^2 f(x) p = \nabla f(x) + (J(x)^T J(x) + S(x))p.$$

As we are employing a Gauss-Newton model for f we are actually omitting the second derivatives of F, as we do not want to compute them. Then, we will omit their contribution also in the gradient model and given the current iterate x_k^δ we will define

$$M_k^g(p) = J(x_k^\delta)^T(F(x_k^\delta) - y^\delta) + J(x_k^\delta)^T J(x_k^\delta)p = g_k + B_k p. \tag{5.4}$$

Notice also that $M_k^g(p)$ is the gradient of

$$\frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p\|^2,$$

i.e. of the approximation of the function f_δ around the current iterate x_k^δ adopted in (5.1). Then, condition (4.8) will be replaced by

$$\|M_k^g(p(\lambda))\| \geq q\|g_k\|, \qquad\qquad q \in (0,1), \tag{5.5}$$

which we will refer to as the generalized q-condition.

The employment of Gauss-newton model has several implications on our method. First, it is well known that if $x^\dagger$ is a solution of (3.1) and $\|S(x^\dagger)\|$ is too large, the Gauss-Newton method may not be locally convergent (see Section 2.3 or [27, §10.2]). $\|S(x^\dagger)\|$ is a combined measure of the nonlinearity and residual size of the problem. Then, with the proposed method we can handle only small residual or mildly nonlinear problems. Moreover, the fact that we are omitting the second order derivatives of F affects also the Assumptions we have to make to analyze local convergence of the method. Assumption 4.12 that is made in the zero residual case will be replaced by the following.

Assumption 5.1. *Given a solution $x^\dagger$ of problem (3.1), there exist $r > 0$, $c > 0$ and $\sigma \in (0,q)$ such that*

$$\|\nabla f(\tilde{x}) - \nabla f(x) - J(x)^T J(x)(\tilde{x} - x)\| \leq (c\|\tilde{x} - x\| + \sigma)\|\nabla f(x) - \nabla f(\tilde{x})\|, \tag{5.6}$$

for all $x, \tilde{x} \in B_r(x^\dagger)$.

This inequality is motivated by the following observations. If ∇f is continuously differentiable, it follows from (5.3)

$$\nabla f(x+p) - \nabla f(x) - J(x)^T J(x)p =$$

$$= \int_0^1 \left[J(x+tp)^T J(x+tp) - J(x)^T J(x)\right] p\, dt + \int_0^1 S(x+tp)p\, dt$$

$$= \int_0^1 J(x+tp)^T [J(x+tp) - J(x)]p\, dt + \int_0^1 [J(x+tp) - J(x)]^T J(x)p\, dt + \int_0^1 S(x+tp)p\, dt.$$

Then, letting σ satisfying

$$\|S(x)\| \leq \sigma \tag{5.7}$$

for x in the considered neighbourhood of $x^\dagger$, if J is Lipschitz continuous, with Lipschitz constant L, and it holds $\|J(x)\| \leq K$, we obtain

$$\|\nabla f(x+p) - \nabla f(x) - J(x)^T J(x)p\| \leq KL\|p\|^2 + \sigma\|p\|.$$

Setting $\tilde{x} = x + p$ we obtain that the following inequality holds with $c = KL$ and σ satisfying (5.7):

$$\|\nabla f(\tilde{x}) - \nabla f(x) - J(x)^T J(x)(\tilde{x} - x)\| \leq c\|\tilde{x} - x\|^2 + \sigma\|\tilde{x} - x\|. \tag{5.8}$$

This condition is analogous to (4.4), that holds for zero residual problems in case J is Lipschitz continuous. We have seen that (4.4) is not strong enough to prove regularizing properties for zero residual problems and has then to be replaced by the tangential cone condition (4.3) (cf. Chapter 4). Here, in the same way, we need a stronger condition than (5.8). In case of rank-deficient Jacobian indeed, it may happen that $\tilde{x} - x$ belongs to the null space of $J(x)^T J(x)$ and $\nabla f(\tilde{x}) = \nabla f(x)$. In this case, the bound on the right of (5.8) is too rough. Writing down the tangential cone condition for $\nabla f(x) = 0$ yields:

$$\|\nabla f(\tilde{x}) - \nabla f(x) - \nabla^2 f(x)(\tilde{x} - x)\| \leq c\|\tilde{x} - x\|\|\nabla f(\tilde{x}) - \nabla f(x)\|.$$

Then condition (5.6) can be seen as a tangential cone condition for $\nabla f(x)$ where the term $S(x)$ containing the second derivatives of F has been dropped from $\nabla^2 f(x)$ and the term $\sigma\|\nabla f(x) - \nabla f(\tilde{x})\|$ appears on the right to take into account this omitted contribution, as well as $\sigma\|\tilde{x} - x\|$ appears in (5.8). As a consequence of the fact that the second derivatives are approximated, the right hand side in (5.6) is $O(\|\nabla f(x) - \nabla f(\tilde{x})\|)$ rather than $O(\|x - \tilde{x}\|\|\nabla f(x) - \nabla f(\tilde{x})\|)$, as it would be if the tangential cone condition for $\nabla f(x)$ was used. Notice also that in Assumption 5.1, σ is assumed to be in $(0, q)$. As it represents a bound for $\|S(x)\|$, this restriction implies that our analysis is focused on small residual problems.

In order to support our Assumption 5.1, we will present in Section 6.3 a model problem for which it is shown to hold and we will provide numerical evidence for it in Section 5.5 for the test problems we consider.

The rest of this chapter is organized as follows. For the zero residual case, it was crucial to have monotonic decrease of the norm of the error to prove the regularizing properties of the method. We need the same property here, and therefore Section 5.1 is devoted to the description of the conditions that allows us to prove the desired monotonicity also in the nonzero residual case. This section is needed to motivate our subsequent choices. In Section 5.2 the method proposed is presented and it is described how to enforce the conditions previously introduced. Section 5.3 is devoted to the theoretical analysis of the method, both in the noise free and in the noisy case. In Section 5.4 we adapt the proposed approach to constrained problems. The numerical performance of the method is studied in Section 5.5.

The method and the theory presented in this section, parallel those presented in [S2]. In [S2] a generic Hilbert space $\mathscr{H}$ is considered, while here we focus on the finite dimensional case $\mathscr{H} = \mathbb{R}^n$, for sake of homogeneity with the rest of the thesis. All the results presented in this chapter indeed are valid in a generic Hilbert space setting, as we briefly remark in Chapter 6.

5.1 Monotonic error decrease

In order to properly choose parameter λ_k and matrix L_k in (5.1) we have to take into account that non-stationary iterated Tikhonov procedures for zero residual problems [28, 50] provide regularizing properties thanks to the fact that the method achieves monotone decrease of the norm of the error $e_k = x^\dagger - x_k^\delta$ between the true solution and the current iterate, even when noisy problems are solved.

Given the step $p_k = p(\lambda_k)$ solution of (5.1), the desired property can be gained thanks to the following two key relations:

$$\|M_k^g(e_k)\| \le \frac{1}{\theta_k}\|M_k^g(p_k)\|, \;\; \theta_k > 1, \tag{5.9}$$

$$B_k^+ p_k = -\frac{1}{\lambda_k} M_k^g(p_k), \tag{5.10}$$

where $M_k^g(p)$ is defined in (5.4). The first relation parallels that established in the zero residual case (4.29) between the zero residual model $M_k(p) = F(x_k^\delta) + J(x_k^\delta)p$ in the error and that in the step, that is enforced through the q-condition (4.8). For nonzero residual problems we use the analogous condition (5.9), employing the nonzero residual model (5.4). The second condition can be enforced by a suitable choice of the matrix L_k. We will discuss this in next section.

In the following lemma we show that if the two relations (5.9) and (5.10) hold and $p_k = p(\lambda_k) \in \mathscr{R}(B_k)$, we can obtain the same results as in Lemma 4.13, that holds in the zero residual case, and prove the monotonic decrease of the error.

Lemma 5.2. *Assume that $x^\dagger$ is a solution of (3.1). Let $e_k = x^\dagger - x_k^\delta$, $M_k^g(p)$ defined in (5.4). Assume that (5.10) is satisfied and there exists $\theta_k > 1$ such that condition (5.9) holds. Suppose furthermore that $J(x_k^\delta)$ is of rank $\ell \le n$ and let $x_{k+1}^\delta = x_k^\delta + p_k$ with $p_k = p(\lambda_k) \in \mathscr{R}(B_k)$. Then it holds*

$$\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 \le \frac{2}{\lambda_k}\left(\frac{1}{\theta_k} - 1\right)\|M_k^g(p_k)\|^2. \tag{5.11}$$

Proof. Since p_k belongs to the range space of B_k it follows that $(p_k)_j = 0$ for $j = \ell+1, \ldots, n$, where $(p_k)_j$ is the j-th component of p_k. Then,

$$< B_k^+ p_k, B_k p_k >= \|p_k\|^2, \qquad\qquad < B_k^+ p_k, B_k e_k >=< p_k, e_k > . \tag{5.12}$$

Then, taking into account (5.10)

$$\begin{aligned}
\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 &= 2 < x_{k+1}^\delta - x_k^\delta, x_k^\delta - x^\dagger > + \|x_{k+1}^\delta - x_k^\delta\|^2 \\
&= 2 < p_k, -e_k > + \|p_k\|^2 = 2 < B_k^+ p_k, -B_k e_k > + \|p_k\|^2 \\
&= 2 < B_k^+ p_k, -g_k - B_k e_k > -2 < B_k^+ p_k, -g_k - B_k p_k > \\
&-2 < B_k^+ p_k, B_k p_k > + \|p_k\|^2 = \\
&= \frac{2}{\lambda_k} < -M_k^g(p_k), -M_k^g(e_k) > - \frac{2}{\lambda_k} < -M_k^g(p_k), -M_k^g(p_k) > - \|p_k\|^2 \\
&\le \frac{2}{\lambda_k}\|M_k^g(p_k)\|\|M_k^g(e_k)\| - \frac{2}{\lambda_k}\|M_k^g(p_k)\|^2 - \|p_k\|^2.
\end{aligned}$$

From condition (5.9)

$$\begin{aligned}\|x_{k+1}^{\delta}-x^{\dagger}\|^2-\|x_k^{\delta}-x^{\dagger}\|^2 &\le \frac{2}{\lambda_k}\frac{1}{\theta_k}\|M_k^g(p_k)\|^2-\frac{2}{\lambda_k}\|M_k^g(p_k)\|^2-\|p_k\|^2\\ &\le \frac{2}{\lambda_k}\left(\frac{1}{\theta_k}-1\right)\|M_k^g(p_k)\|^2.\end{aligned}$$

□

Then, in order to obtain the desired monotone decrease of the error, we need to ensure (5.9) and (5.10) to hold and the step to be in the range of B_k.

5.2 The method

Motivated by the previous considerations, we present a non-stationary iterated Tikhonov procedure for least squares problems employing a step satisfying both conditions (5.9) and (5.10). Let us first focus on the case in which $J(x_k^{\delta})$ is full rank.

A step p_k is the solution of (5.1) if and only if it satisfies the following linear system:

$$(J(x_k^{\delta})^T J(x_k^{\delta})+\lambda_k L_k)p=-J(x_k^{\delta})^T(F(x_k^{\delta})-y^{\delta}). \tag{5.13}$$

Relation (5.10) can be obtained with a suitable choice of matrix L_k. From (5.4) and (5.13) indeed, it follows that $-\frac{1}{\lambda_k}M_k^g(p_k)=L_kp_k$. This suggests to choose $L_k=B_k^{-1}$.

As in Chapter 4, we look for a method in which the parameter λ_k is set in an automatic way and such that (5.9) holds. Then, we adopt a reformulation of problem (5.1) and we consider the following elliptical Trust-Region problem:

$$\begin{aligned}&\min_{p\in\mathbb{R}^n}\frac{1}{2}\|F(x_k^{\delta})-y^{\delta}+J(x_k^{\delta})p\|^2\\ &\text{s.t. } \|B_k^{-1/2}p\|\le\Delta_k.\end{aligned} \tag{5.14}$$

Its solution [22] is a step $p=p(\lambda)$ satisfying

$$(B_k+\lambda B_k^{-1})p=g_k, \tag{5.15a}$$

$$\lambda\ge 0 \tag{5.15b}$$

$$\lambda(\|B_k^{-1/2}p\|-\Delta_k)=0, \tag{5.15c}$$

$$\|B_k^{-1/2}p\|\le\Delta_k. \tag{5.15d}$$

Then, given the couple $(\lambda_k,p(\lambda_k))\in\mathbb{R}^+\times\mathbb{R}^n$ solution of (5.15), if $\lambda_k>0$, the step $p_k=p(\lambda_k)$ solves (5.1) with $L_k=B_k^{-1}$. With this reformulation of the problem, instead of choosing λ_k in order to obtain a step $p(\lambda_k)$ satisfying (5.9), we need to properly select the Trust-Region radius Δ_k to obtain a couple $(\lambda_k,p(\lambda_k))$ with $\lambda_k>0$ and $p(\lambda_k)$ satisfying (5.9). We will show in Lemma 5.6 how to make such a suitable choice of Δ_k.

Let us focus on how to compute $p(\lambda_k)$. As usual, we reformulate the problem as a spherical Trust-Region problem. Letting $z=B_k^{-1/2}p$, problem (5.14) reduces to

$$\begin{aligned} &\min_{z\in\mathbb{R}^n} \frac{1}{2} z^T B_k^2 z + (B_k^{1/2} g_k)^T z + f_\delta(x_k^\delta), \\ &\text{s.t. } \|z\| \le \Delta_k. \end{aligned} \tag{5.16}$$

The solution to (5.16) is given by $z_k = z(\lambda_k)$ where the couple $(\lambda_k, z(\lambda_k))$ is the solution of

$$(B_k^2 + \lambda I) z(\lambda) = -B_k^{1/2} g_k, \tag{5.17a}$$

$$\lambda \ge 0, \tag{5.17b}$$

$$\lambda(\|z(\lambda)\| - \Delta_k) = 0, \tag{5.17c}$$

$$\|z(\lambda)\| \le \Delta_k. \tag{5.17d}$$

If we let

$$p(\lambda) = B_k^{1/2} z(\lambda), \tag{5.18}$$

$p_k = p(\lambda_k)$ solves (5.14). Then, the step can be computed solving (5.17) and employing (5.18). Note that, in order to compute $z(\lambda_k)$ the linear system (5.17a) has to be solved in which matrix B_k^2 appears. In the applications we consider, the ill-conditioning of matrix B_k arises from the smallest singular values close to zero. We will prove that in our approach λ_k is ensured to be strictly positive. Then, the conditioning of the linear system is not deteriorated by the fact that B_k is squared. Moreover, we assume to be able to compute the SVD decomposition of B_k to evaluate the right-hand side in (5.17a).

In case $J(x_k^\delta)$ is not full rank, problem (5.14) is not well-defined. However we can still compute the vector $z(\lambda_k)$ solution of the Trust-Region subproblem (5.16) and define the step through (5.18). In this way, we still have a step $p(\lambda)$ satisfying (5.10), as we show in the following lemma. Notice also that $p_k \in \mathscr{R}(B_k)$, then Lemma 5.2 holds.

Lemma 5.3. *Suppose $\|g_k\| \ne 0$ and that $J(x_k^\delta)$ is of rank $\ell \le n$. Let $z(\lambda)$ be the minimum norm solution of (5.17a) with $\lambda \ge 0$ and $p(\lambda)$ given in (5.18). Then it holds*

$$B_k^+ p(\lambda) = -\frac{1}{\lambda} M_k^g(p(\lambda)). \tag{5.19}$$

Proof. Let $\bar{\Sigma}_k \in \mathbb{R}^{n\times n}$ be the diagonal matrix with entries $\varsigma_1, \ldots, \varsigma_\ell, 0, \ldots, 0$ on the diagonal and $r = U_k^T(F(x_k^\delta) - y^\delta)$. Then, $B_k = V_k \Sigma_k^T \Sigma_k V_k^T = V_k \bar{\Sigma}_k^2 V_k^T$ and from (5.17a) it follows

$$z(\lambda) = -V_k \bar{\Sigma}_k (\bar{\Sigma}_k^4 + \lambda I)^+ \Sigma_k^T r \tag{5.20}$$

and (5.18) yields

$$p(\lambda) = -V_k \bar{\Sigma}_k^2 (\bar{\Sigma}_k^4 + \lambda I)^+ \Sigma_k^T r. \tag{5.21}$$

Then, from (5.4)

$$M_k^g(p(\lambda)) = -V_k \bar{\Sigma}_k^4 (\bar{\Sigma}_k^4 + \lambda I)^+ \Sigma_k^T r + V_k \Sigma_k^T r = V_k \lambda (\bar{\Sigma}_k^4 + \lambda I)^+ \Sigma_k^T r. \tag{5.22}$$

Taking into account that

$$B_k^+ = V_k {\bar{\Sigma}_k^2}^+ V_k^T,$$

and comparing the expression of $M_k^g(p(\lambda))$ with (5.21) we obtain the thesis. □

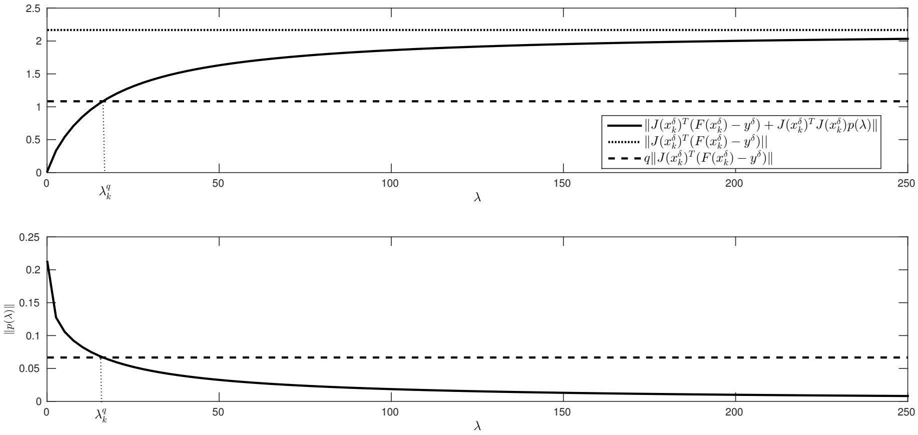


Figure 5.1: *Effect of the generalized q-condition 5.5 on the step length.*

Then, both in case the Jacobian is full-rank or rank deficient we can obtain (5.10).

Regarding condition (5.9), we will see in the next section (cf. Lemmas 5.9 and 5.14) that this relation between the model evaluated in the step and that evaluated in the error can be granted, provided that the step $p_k = p(\lambda_k)$ satisfies the generalized q-condition (5.5). Here, we show how (5.5) can be enforced by a suitable Trust-Region radius choice.

5.2.1 Choice of the Trust-Region radius

Condition (5.5) controls the value of the norm of the gradient model that has to be greater than a fixed fraction of the norm of the gradient and it provides a criterion to choose the free parameter λ_k in (5.1). As (4.8), the generalized q-condition is a constraint on the length of the step, its effect on it is illustrated in Figure 5.1 where we plot $\|M_k^g(p(\lambda))\|$ (top) and $\|p(\lambda)\|$ (bottom) varying λ. By imposing (5.5) the regularization parameter λ is forced to be greater then the value λ_k^q satisfying

$$\|M_k^g(p(\lambda_k^q))\| = q\|g_k\|, \tag{5.23}$$

avoiding too small values that correspond to large steps, as it is shown at the bottom of Figure 5.1.

We are going to show that differently from (4.6), a λ_k^q satisfying (5.23) always exists and that a step p_k of the form (5.18) satisfying (5.5) can be provided by an appropriate Trust-Region radius choice. To this end we need the following preliminary results.

Lemma 5.4. *Suppose $\|g_k\| \neq 0$ and that $J(x_k^\delta)$ is of rank $\ell \leq n$. Let $z(\lambda)$ be the minimum norm solution of (5.17a) with $\lambda \geq 0$ and $p(\lambda)$ given in (5.18). Then, denoting $r = U_k^T(F(x_k^\delta) - y^\delta) = [r_1, \dots, r_m]$, we have that*

$$\|z(\lambda)\|^2 = \sum_{i=1}^{\ell} \left(\frac{\varsigma_i^2 r_i}{\varsigma_i^4 + \lambda} \right)^2. \tag{5.24}$$

Moreover, $\|M_k^g(p(\lambda))\|$ *is a monotone increasing function for* $\lambda \geq 0$ *and*

$$\lim_{\lambda\to 0} \|M_k^g(p(\lambda))\| = 0,$$
$$\lim_{\lambda\to\infty} \|M_k^g(p(\lambda))\| = \|J(x_k^\delta)^T(F(x_k^\delta)-y^\delta)\|.$$

Proof. Note that (5.20) yields (5.24). Moreover, from (5.22) it follows

$$\|M_k^g(p(\lambda))\|^2 = \sum_{i=1}^{\ell}\left(\frac{\varsigma_i\lambda r_i}{\varsigma_i^4+\lambda}\right)^2. \tag{5.25}$$

Taking derivatives it is possible to show that the function $\|M_k^g(p(\lambda))\|$ is monotonic increasing. Then, taking into account that

$$\|J(x_k^\delta)^T(F(x_k^\delta)-y^\delta)\|^2 = \sum_{i=1}^{\ell}(\varsigma_i r_i)^2, \tag{5.26}$$

the thesis easily follows. □

Now, we are in the position of proving that condition (5.5) can be satisfied.

Lemma 5.5. *Let* $z(\lambda)$ *be the minimum norm solution of* (5.17a) *with* $\lambda \geq 0$ *and* $p(\lambda)$ *be given in (5.18). It exists* $\lambda_k^q > 0$ *such that if* $\lambda_k \geq \lambda_k^q$ *then* $p_k = p(\lambda_k)$ *satisfies condition* (5.5)*.*

Proof. From Lemma 5.4 if follows that $\|M_k^g(p(\lambda))\|$ is a monotonic increasing function for $\lambda \geq 0$. As $0 \leq \|M_k^g(p(\lambda))\| \leq \|g_k\|$ for all $\lambda \geq 0$, there exists λ_k^q such that (5.23) holds. Then, condition (5.5) is satisfied for any $\lambda_k \geq \lambda_k^q$ and $\lambda_k^q = 0$ if and only if $\|J(x_k^\delta)^T(y^\delta - F(x_k^\delta))\| = 0$. □

We now provide a suitable choice of the Trust-Region radius that guarantees that the resulting regularization parameter λ_k is big enough to ensure that the step $p_k = p(\lambda_k)$ satisfies condition (5.5).

Lemma 5.6. *Let* $z_k = z(\lambda_k)$ *be the minimum norm solution of* (5.17a) *with* $\lambda_k \geq 0$ *and* $p_k = p(\lambda_k)$ *be given in (5.18). If*

$$\Delta_k \leq \frac{1-q}{\|B_k\|^2}\|B_k^{1/2}g_k\| \tag{5.27}$$

the step p_k *satisfies* (5.5)*.*

Proof. From (5.25) and (5.26) it follows

$$\begin{aligned}\|M_k^g(p(\lambda))\|^2 &= \lambda^2\sum_{i=1}^{\ell}\left(\frac{\varsigma_i r_i}{\varsigma_i^4+\lambda}\right)^2 \geq \lambda^2\frac{\sum_{i=1}^{\ell}(\varsigma_i r_i)^2}{(\|B_k\|^2+\lambda)^2}\\ &= \lambda^2\frac{1}{(\|B_k\|^2+\lambda)^2}\|J(x_k^\delta)^T(F(x_k^\delta)-y^\delta)\|^2.\end{aligned}$$

Then, we can obtain an upper bound for λ_k^q defined in (5.23) proceeding as follows:

$$\begin{aligned} q\|J(x_k^\delta)^T(F(x_k^\delta)-y^\delta)\| &= \|M_k^g(p(\lambda_k^q))\| \\ &\geq \frac{\lambda_k^q}{\|B_k\|^2+\lambda_k^q}\|J(x_k^\delta)^T(F(x_k^\delta)-y^\delta)\|, \end{aligned}$$

so

$$\lambda_k^q \leq \frac{q\|B_k\|^2}{1-q}. \tag{5.28}$$

By (5.17a) one has

$$\|z(\lambda_k^q)\| \geq \frac{\|B_k^{1/2}g_k\|}{\|B_k^2+\lambda_k^q I\|}, \tag{5.29}$$

and by (5.28) it holds

$$\|B_k^2+\lambda_k^q I\| \leq \|B_k\|^2 + \frac{q}{1-q}\|B_k\|^2 = \frac{1}{1-q}\|B_k\|^2.$$

By construction $\|z_k\| \leq \Delta_k$. If (5.27) holds, using (5.29), we obtain

$$\|z_k\| = \|z(\lambda_k)\| \leq \frac{1-q}{\|B_k\|^2}\|B_k^{1/2}g_k\| \leq \frac{\|B_k^{1/2}g_k\|}{\|B_k^2+\lambda_k^q I\|} \leq \|z(\lambda_k^q)\|.$$

Since by (5.24) it follows that $\|z(\lambda)\|$ is monotonically decreasing, the previous inequality yields $\lambda_k \geq \lambda_k^q$ and by Lemma 5.5 the thesis holds.

□

With this choice of Δ_k it is not necessary to check if condition (5.5) is satisfied. Notice also that the Trust-Region radius goes to zero whenever $\|g_k\|$ converges to zero.

From Lemma 5.3 and Lemma 5.6 we can conclude that a step p_k of the form (5.18), satisfying both (5.10) and (5.5) exists. We will show in the next section that if condition (5.5) is met, then also condition (5.9) holds, so our method provides the desired monotone decrease of the norm of the error, as stated in Lemma 5.2.

These results suggest the Trust-Region iteration described in Algorithm 5.1. Once p_k has been computed, the classical ratio between the actual and the predicted reduction is computed. As in classical Trust-Region approaches, if there is a good agreement between the function and the model the step is accepted, otherwise the step is rejected and the Trust Region is reduced.

Regarding the well-definiteness and the choice of constants in Algorithm 5.1, the same reasoning as for Algorithm 4.2 applies, cf. Section 4.2. In case of noisy data the process is stopped whenever the norm of the gradient goes under the noise level, i.e. at iteration $k_*(\delta)$ satisfying the discrepancy principle (5.2).

5.3 Convergence analysis

The convergence analysis is carried out under Assumption 5.1 and under the following additional assumption.

Algorithm 5.1 k-th iteration of the elliptical regularizing Trust-Region method for problem (3.1)

```
**Input:** $x_k^\delta$, $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < C_{\min} < C_{\max}$, $q \in (0,1)$, $y^\delta$.
1. Choose $\Delta_k \in \left[C_{\min}\|B_k^{1/2}g_k\|, \min\left\{C_{\max}, \frac{1-q}{\|B_k\|^2}\right\}\|B_k^{1/2}g_k\|\right]$.
2. Repeat
   2.1 Compute the solution $z_k$ of Trust-Region subproblem (5.16).
   2.2 Set $p_k = B_k^{1/2}z_k$.
   2.3 Compute
       $$\rho_k(p_k) = \frac{f_\delta(x_k^\delta) - f_\delta(x_k^\delta + p_k)}{f_\delta(x_k^\delta) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2}.$$
   2.4 If $\rho_k(p_k) < \eta$,set $\Delta_k = \gamma\Delta_k$.
Until $\rho_k(p_k) \ge \eta$.
3. Set $x_{k+1}^\delta = x_k^\delta + p_k$.
```

Assumption 5.7. *$J(x)$ is Lipschitz continuous in a neighbourhood of the level set $\mathscr{L} = \{x \in \mathbb{R}^n \ s.t. \ f_\delta(x) \le f_\delta(x_0)\}$ with Lipschitz constant L.*

Notice that this assumption is commonly made in ill-posed problem context, cf. [67]. We will consider first the noise free case and then the noisy case.

5.3.1 Noise free case

In this section we consider noise free problems, we assume that $\delta = 0$ and we drop the symbol δ from the generated sequence, the data y and the function.

We prove the local convergence properties of the method. As in the zero residual case, we assume that there exists a specific iterate $\bar{k}$ such that $x_{\bar{k}}$ is sufficiently close to a solution of (3.1) and inequality (5.6) holds in a neighbourhood of such iterate.

Assumption 5.8. *Let $\delta = 0$ and $x^\dagger$ be a solution of (3.1). Suppose that for some iteration index $\bar{k}$ there exist $r > 0$, $c > 0$ and $\sigma \in (0,q)$ such that inequality (5.6) holds for any $x, \tilde{x} \in B_{2r}(x_{\bar{k}})$ and*

$$\|x_{\bar{k}} - x^\dagger\| < \min\left\{\frac{q-\sigma}{c}, r\right\}. \tag{5.30}$$

In the following Lemma we show that, under the previous assumptions, (5.9) holds for all $k \ge \bar{k}$ and therefore by Lemma 5.2 the error decreases monotonically for $k \ge \bar{k}$.

Lemma 5.9. *Assume that Assumption 5.8 holds. Let $e_k = x^\dagger - x_k$ and p_k computed at step 2.2 of Algorithm 5.1. Then, it exists $\theta_k > 1$ such that condition (5.9) holds for all $k \ge \bar{k}$.*

Proof. From the choice of Δ_k at step 1 of Algorithm 5.1 and Lemma 5.6 it follows that the step p_k satisfies condition (5.5). From (5.5) and (5.6) we obtain

$$\|M^g_{\bar{k}}(e_{\bar{k}})\| \le \left(c\|e_{\bar{k}}\| + \sigma\right)\|J(x_{\bar{k}})^T(F(x_{\bar{k}}) - y)\| \le$$
$$\le \left(\frac{c\|e_{\bar{k}}\| + \sigma}{q}\right)\|M^g_{\bar{k}}(p_{\bar{k}})\|,$$

so that condition (5.9) is satisfied for $k = \bar{k}$ with $\theta_{\bar{k}} = \frac{q}{c\|e_{\bar{k}}\|+\sigma} > 1$ from Assumption 5.8. From Lemma 5.2 it follows $\|e_{\bar{k}+1}\| < \|e_{\bar{k}}\|$ so that $x_{\bar{k}+1} \in B_{2r}(x_{\bar{k}}) \cap B_r(x^\dagger)$ and Assumption 5.8 holds also for $k = \bar{k}+1$. Repeating the above arguments, by induction it is possible to prove that condition (5.9) holds for all $k \ge \bar{k}$, with

$$\theta_k = \frac{q}{c\|e_k\| + \sigma} > 1. \tag{5.31}$$

□

In the next Lemma some important features of the procedure are shown.

Lemma 5.10. *Suppose that Assumption 5.8 holds and $g(x_k) \ne 0$. Then, Algorithm 5.1 generates a sequence $\{x_k\}$ such that, for $k \ge \bar{k}$,*

(i) $\lambda_k > 0$, and x_k belongs to $B_{2r}(x_{\bar{k}}) \cap B_r(x^\dagger)$;

(ii) $\|x_{k+1} - x^\dagger\| < \|x_k - x^\dagger\|$, $\theta_{k+1} > \theta_k$.

Proof. From the choice of Δ_k at step 1 of Algorithm 5.1 and Lemma 5.6 it follows that the step p_k computed at step 2.2 satisfies condition (5.5), so $\lambda_k \ge \lambda_k^q > 0$.

From Lemma 5.9, Lemma 5.2 holds for all $k \ge \bar{k}$ and (5.11) implies that the sequence $\{\|x_k - x^\dagger\|\}_{k=\bar{k}}^{\infty}$ is monotonic decreasing. As a consequence, x_k belongs to $B_{2r}(x_{\bar{k}}) \cap B_r(x^\dagger)$ for all $k \ge \bar{k}$ and from (5.31), $\theta_{k+1} > \theta_k$ for all $k \ge \bar{k}$.

□

Remark 5.11. *Lemma 5.10 shows that λ_k is strictly positive, then from (5.17c) it follows that $\|z_k\| = \Delta_k$, i.e. the Trust Region is active, to produce a regularizing effect. This feature is shared also by the Trust-Region method proposed for the zero residual case.*

In the next theorem we prove convergence of the sequence $\{x_k\}$ to a point belonging to $\mathscr{S} \cap \bar{B}_r(x^\dagger)$ where

$$\mathscr{S} = \{x \mid J(x)^T(y - F(x)) = 0\}. \tag{5.32}$$

Theorem 5.12. *Suppose that Assumption 5.8 holds. Then, the sequence $\{x_k\}$ generated by Algorithm 5.1 converges to a solution x^* of (3.1) such that $\|x^* - x^\dagger\| \le r$.*

Proof. Let $\bar{k}$ as in Assumption 5.8. Lemma 5.9 and 5.10 show that (5.11) holds for all $k \ge \bar{k}$ with θ_k given in (5.31). Let $e_k = x^\dagger - x_k$, $\bar{k} \le j < k$ and l between j and k such that

$$\|J(x_l)^T(y - F(x_l))\| = \min_{j \le i < k} \|J(x_i)^T(y - F(x_i))\|.$$

Let $\bar{\gamma} = c\|x^\dagger - x_{\bar{k}}\| \geq c\|x^\dagger - x_i\|$ for all $i \geq \bar{k}$. Using (5.6) and the definition of l we obtain for all $j \leq i < k$

$$\begin{aligned}\|B_i e_l\| \leq & \| - g_i - B_i(x^\dagger - x_i)\| + \|g_l - g_i - B_i(x_l - x_i)\| + \|g_l\| \\ \leq & (c\|x^\dagger - x_i\| + \sigma)\|g_i\| + (c\|x_l - x_i\| + \sigma)\|g_l - g_i\| + \|g_l\| \\ \leq & (c\|x^\dagger - x_i\| + 2\sigma + c\|x_l - x_i\|)\|g_i\| + (c\|x_l - x_i\| + \sigma + 1)\|g_l\| \\ \leq & (3c\|x^\dagger - x_i\| + 2c\|x^\dagger - x_l\| + 3\sigma + 1)\|g_i\| \\ \leq & (5\bar{\gamma} + 3\sigma + 1)\|g_i\| = \tilde{c}\|g_i\|\end{aligned}$$

where $\tilde{c} = 5\bar{\gamma} + 3\sigma + 1$, so that

$$\|B_i e_l\| \leq \tilde{c}\|g_i\|, \tag{5.33}$$

for all $j \leq i < k$. Taking into account that p_k belongs to the range space of $B_k^{1/2}$, from (5.10) and (5.33) we obtain that for $k > j \geq \bar{k}$:

$$\begin{aligned}| < e_l - e_k, e_l > | = & \left| \sum_{i=l}^{k-1} < p_i, e_l > \right| = \left| \sum_{i=l}^{k-1} < B_i^+ p_i, B_i e_l > \right| = \left| \sum_{i=l}^{k-1} \frac{1}{\lambda_i} < M_i^g(p_i), B_i e_l > \right| \\ \leq & \sum_{i=l}^{k-1} \frac{1}{\lambda_i} \|M_i^g(p_i)\| \|B_i e_l\| \leq \sum_{i=l}^{k-1} \frac{\tilde{c}}{\lambda_i} \|M_i^g(p_i)\| \|g_i\|.\end{aligned}$$

From (5.5)

$$| < e_l - e_k, e_l > | \leq \sum_{i=l}^{k-1} \frac{\tilde{c}}{q\lambda_i} \|M_i^g(p_i)\|^2.$$

Thus (5.11) yields

$$| < e_l - e_k, e_l > | \leq \sum_{i=l}^{k-1} \frac{\tilde{c}}{2q} \frac{\theta_i}{\theta_i - 1} (\|e_i\|^2 - \|e_{i+1}\|^2) \leq \beta_{\bar{k}} (\|e_l\|^2 - \|e_k\|^2), \tag{5.34}$$

where $\beta_{\bar{k}} = \dfrac{\tilde{c}}{2q} \dfrac{\theta_{\bar{k}}}{\theta_{\bar{k}} - 1}$ and we have used $\theta_i/(\theta_i - 1) < \theta_{\bar{k}}/(\theta_{\bar{k}} - 1)$ since function $\theta/(\theta - 1)$ is monotonic decreasing and sequence θ_k is monotonic increasing (see Lemma 5.10). Similarly, it is possible to show that

$$| < e_l - e_j, e_l > | \leq \beta_{\bar{k}} (\|e_j\|^2 - \|e_l\|^2). \tag{5.35}$$

Then from (5.34) and (5.35) it follows

$$\begin{aligned}\|e_k - e_l\|^2 &= 2 < e_l - e_k, e_l > + \|e_k\|^2 - \|e_l\|^2 \leq (2\beta_{\bar{k}} + 1)(\|e_l\|^2 - \|e_k\|^2), \\ \|e_l - e_j\|^2 &= 2 < e_l - e_j, e_l > + \|e_j\|^2 - \|e_l\|^2 \leq (2\beta_{\bar{k}} + 1)(\|e_j\|^2 - \|e_l\|^2), \\ \|x_k - x_j\|^2 &= \|e_k - e_j\|^2 \leq \|e_k - e_l\|^2 + \|e_l - e_j\|^2.\end{aligned}$$

Since the sequence $\{\|e_k\|\}$ is bounded from below and monotonic decreasing, hence convergent, it follows that $\{x_k\}$ is a Cauchy sequence, i.e. $\{x_k\}$ converges to a limit point x^*. As $x_k \in B_r(x^\dagger)$ for $k \geq \bar{k}$, it follows $\|x^* - x^\dagger\| \leq r$. □

5.3.2 Noisy case

Here, we assume $\delta > 0$ and we show the regularizing properties of the method in case of noisy data. We assume that there exists a specific iterate $x^\delta_{\bar{k}}$ sufficiently close to a solution $x^\dagger$ of (3.1) and that inequality (5.6) holds in a neighbourhood of such iterate. The proofs of some results are similar to those of their counterpart, stated in the zero residual case. Then, for ease of readability of the thesis, these proofs are reported in the Appendix.

Assumption 5.13. *Let $\delta > 0$ and $x^\dagger$ be a solution of (3.1). Suppose that for some iteration index $\bar{k} < k_*(\delta)$, with $k_*(\delta)$ defined in (5.2), there exist $r > 0$, $c > 0$ and $\sigma \in (0, q)$ such that inequality (5.6) holds for any $x, \tilde{x} \in B_{2r}(x^\delta_{\bar{k}})$. Moreover assume that it exists a positive constant K_J such that*

$$\|J(x)\| \le K_J$$

for any x belonging to the level set $\mathscr{L} = \{x \in \mathbb{R}^n \;\; s.t. \;\; f_\delta(x) \le f_\delta(x_0)\}$ and that $x^\delta_{\bar{k}}$ satisfies

$$\|x^\delta_{\bar{k}} - x^\dagger\| < \min\left\{\frac{(q-\sigma)\tau - K_J(\sigma+1)}{c(K_J+\tau)}, r\right\}, \qquad \text{with} \quad \tau > \frac{K_J(\sigma+1)}{q-\sigma}. \tag{5.36}$$

Notice that in problems we are dealing with, bound K_J is generally not large, typically of the order of the unit. Moreover, in the numerical results section we will show that the behaviour of our procedure does not depend strongly on the choice of q. Then, it is possible to ensure $q - \sigma$ to be positive and reasonably far from zero without affecting the method performance.

Lemma 5.14. *Assume that Assumption 5.13 holds and let $e_k = x^\dagger - x^\delta_k$ and p_k computed at step 2.2 of Algorithm 5.1. Then, it exists $\theta_k > 1$ such that condition (5.9) holds for all $\bar{k} \le k < k_*(\delta)$.*

Proof. By (5.6) and (3.2) we obtain

$$\begin{aligned}
\|M^g_{\bar{k}}(e_{\bar{k}})\| &= \|J(x^\delta_{\bar{k}})^T(F(x^\delta_{\bar{k}}) - y^\delta + J(x^\delta_{\bar{k}})(x^\dagger - x^\delta_{\bar{k}}))\| \\
&\le \|J(x^\delta_{\bar{k}})^T(y^\delta - y)\| + \|J(x^\delta_{\bar{k}})^T(F(x^\delta_{\bar{k}}) - y + J(x^\delta_{\bar{k}})(x^\dagger - x^\delta_{\bar{k}}))\| \\
&\le K_J\delta + (c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma)\,\|J(x^\delta_{\bar{k}})^T(y - F(x^\delta_{\bar{k}}))\| \\
&\le (1 + c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma)K_J\delta + (c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma)\,\|J(x^\delta_{\bar{k}})^T(y^\delta - F(x^\delta_{\bar{k}}))\|.
\end{aligned}$$

From the choice of Δ_k at step 1 of Algorithm 5.1 and Lemma 5.6 it follows that the step p_k satisfies condition (5.5). Then, at iteration $\bar{k}$, conditions (5.2) and (5.5) give

$$\begin{aligned}
\|M^g_{\bar{k}}(e_{\bar{k}})\| &\le \left(K_J\frac{1 + c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma}{\tau} + (c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma)\right)\|J(x^\delta_{\bar{k}})^T(y^\delta - F(x^\delta_{\bar{k}}))\| \\
&\le \left(K_J\frac{1 + c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma}{q\tau} + \frac{c\|x^\dagger - x^\delta_{\bar{k}}\| + \sigma}{q}\right)\|M^g_{\bar{k}}(p_{\bar{k}})\|,
\end{aligned}$$

which yields (5.9) at $k = \bar{k}$ with $\theta_{\bar{k}} = \dfrac{q\tau}{K_J + c(K_J+\tau)\|x^\dagger - x_{\bar{k}}^\delta\| + \sigma(K_J+\tau)} > 1$ from (5.36). Then, Lemma 5.2 holds for $k = \bar{k}$ and $\|x_{\bar{k}+1} - x^\dagger\| < \|x_{\bar{k}} - x^\dagger\|$, so that (5.36) also holds for $k = \bar{k}+1$. Repeating the above arguments, by induction we can prove that, for $\bar{k} < k < k_*(\delta)$, condition (5.9) holds, with $\theta_k = \dfrac{q\tau}{K_J + c(K_J+\tau)\|x^\dagger - x_k^\delta\| + \sigma(K_J+\tau)} >$ 1.

□

Next Lemma shows key properties of Algorithm 5.1. The proof follows the lines of Lemma 4.18 and it is reported in the Appendix.

Lemma 5.15. *Suppose that Assumptions 5.7 and 5.13 hold. Then, Algorithm 5.1 generates a sequence $\{x_k^\delta\}$ such that, for $\bar{k} \le k < k_*(\delta)$,*

(i) $\lambda_k > 0$ and x_k^δ belongs to $B_{2r}(x_{\bar{k}}^\delta) \cap B_r(x^\dagger)$;

(ii) $\|x_{k+1}^\delta - x^\dagger\| < \|x_k^\delta - x^\dagger\|$, $\theta_{k+1} > \theta_k$;

(iii) there exists a constant $\bar{\lambda} > 0$ such that $\lambda_k \le \bar{\lambda}$.

Finally, exploiting the previous results, in Theorem 5.16 we show that, given a sequence $\{\delta_n\}$ of noise levels, under suitable assumptions, the sequence of computed approximations $\{x_{k_*(\delta_n)}^{\delta_n}\}$ converges to a stationary point of (3.1) whenever δ_n tends to zero. As in Theorem 4.19 we assume that $\rho_k(x_{k+1} - x_k) \ne \eta$, for all $k \ge 0$. Under this assumption the Trust-Region radius Δ_k selected in Algorithm 5.1, and the scalar λ_k, implicitly defined by the Trust-Region problem, depend continuously on δ in a right interval of the origin. Indeed, the same considerations on the continuous dependence of the iterates on the noise, reported just before Theorem 4.19, also hold in this case. The proof of the Theorem is reported in the Appendix.

Theorem 5.16. *Suppose that Assumptions 5.7 and 5.13 hold.*

(i) The iterates generated by Algorithm 5.1 satisfy the stopping criterion (5.2) after a finite number $k_*(\delta)$ of iterations.

(ii) Suppose further that the sequence $\{x_k\}$ generated with the exact data y satisfies $\rho_k(x_{k+1} - x_k) \ne \eta$, for all k. Then the sequence $\{x_{k_*(\delta)}^\delta\}$ converges to a point belonging to $\mathscr{S} \cap \bar{B}_r(x^\dagger)$, where $\mathscr{S}$ is defined in (5.32), whenever δ tends to zero.

5.4 Constrained case

In many practical applications one has to deal with problems with constraints on the variables. Non-stationary iterated Tikhonov methods for linear least squares problems with convex constraints have been considered in [17]. Let $\Omega \in \mathbb{R}^n$ be a closed and convex set and consider the following problem:

$$\min_{x\in\Omega} f_\delta(x) = \frac{1}{2}\|F(x) - y^\delta\|^2. \tag{5.37}$$

Let $P_\Omega : \mathbb{R}^n \to \Omega$ be the metric projection of $\mathbb{R}^n$ on Ω:

$$P_\Omega(x) = \arg\min_{y\in\Omega} \frac{1}{2}\|x-y\|^2,$$

for all x in $\mathbb{R}^n$. We assume that a solution $x^\dagger \in \Omega$ exists and the computation of the projection P_Ω is not computationally expensive.

The procedure described in Section 5.2 can be modified as outlined in Algorithm 5.2 in order to handle the constraints and preserve its local properties. In what follows we consider the noisy case.

Algorithm 5.2 k-th iteration of the elliptical regularizing Trust-Region method for problem (5.37)

```
Input: $x_k^\delta$, $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < C_{\min} < C_{\max}$, $q \in (0,1)$, $y^\delta$.
1. Choose $\Delta_k \in \left[ C_{\min}\|B_k^{1/2} g_k\|, \min\left\{ C_{\max}, \frac{1-q}{\|B_k\|^2} \right\} \|B_k^{1/2} g_k\| \right]$.
2. Compute the solution $z_k$ of Trust-Region problem (5.16) and set $p_k = B_k^{1/2} z_k$.
3. Set $x_{k+1}^\delta = P_\Omega(x_k^\delta + p_k)$.
```

In Algorithm 5.2 we do not enforce the decrease of the objective function, as our aim here is just to sketch a local procedure for the constrained problem. A global convergent procedure for the noise free case would require more sophisticated strategies for handling the constraints and at the same time providing regularization properties. Then the role of the Trust Region is just that of providing a step satisfying (5.5). This step is used to compute the updated point $x_k^\delta + p_k$ that is then projected on the feasible set, so that the new solution approximation is computed as $x_{k+1}^\delta = P_\Omega(x_k^\delta + p_k)$. This way the generated sequence x_k^δ belongs to Ω. All the local properties of the procedure are maintained, in particular the monotone decrease of the error thanks to the following remark:

Remark 5.17. *Since $x^\dagger \in \Omega$, $\|P_\Omega(x_k^\delta + p_k) - x^\dagger\| \le \|x_k^\delta + p_k - x^\dagger\|$.*

Lemma 5.18. *Assume that $x^\dagger$ is a solution of (5.37). Assume that there exists $\theta_k > 1$ such that condition (5.9) holds. Let $x_{k+1}^\delta = P_\Omega(x_k^\delta + p_k)$ with p_k computed at step 2 of Algorithm 5.2. Then (5.11) holds.*

Proof.

$$\begin{aligned}\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k - x^\dagger\|^2 = &\|P_\Omega(x_k^\delta + p_k) - x^\dagger\|^2 - \|x_k - x^\dagger\|^2 \le \\ &\le \|x_k^\delta + p_k - x^\dagger\|^2 - \|x_k - x^\dagger\|^2,\end{aligned}$$

and the thesis can be obtained repeating the proof of Lemma 5.2 since the step computed in step 2 of Algorithm 5.2 satisfies (5.10). □

Thanks to this key result, proofs of Lemmas 5.14-5.15 and Theorem 5.16 can be repeated. In this regard we underline that the proof of point (iii) of Lemma 5.15 (see the Appendix) simplifies as the upper bound on λ_k is given by inequality (A.1) as Δ_k is chosen at step 1 of Algorithm 5.2 and it is not further reduced.

5.5 Numerical results

In this section, we report on the numerical behaviour of our procedure, that we are going to address as *elliptical regularizing Trust-Region*, in case of noisy data. We have selected four nonlinear ill-posed least squares problems. Problems `R1`, `R2` arise from the discretization of two parameter identification problems, while `R3`, `R4` are originally formulated as discrete problems. In the following we are going to denote with $\|\cdot\|$ the Euclidean norm.

- `R1: A 1D parameter identification problem.` We consider the following problem [21, 75]. We want to reconstruct c in the 1D-elliptic problem

$$-au_{xx} + cu = \varphi \quad \text{in } (0,1) \tag{5.38a}$$

$$u'(0) = 0, u'(1) = 0, \tag{5.38b}$$

given $u, \varphi \in L^2(0,1)$. Identifying c reduces to solve for a given approximation $\bar{u}$ of a solution of (5.38) the following nonlinear least squares problem:

$$\min_c \|F(x) - \bar{u}\|^2_{L^2},$$

for F the operator mapping c to the corresponding solution of (5.38). We choose $a = 4$, φ given by (5.38a) with

$$c(x) = \sqrt{2}cos(2\pi x) + 2, \tag{5.39}$$

$$u(x) = cos(2\pi x) + 2. \tag{5.40}$$

We assume the realistic situation in which both the solution u given in (5.40) of the partial differential equation and the function φ are known just in n points, $\{t_1, \ldots, t_n\} \subset (0,1)$. We define then $\tilde{\varphi}$ and $\tilde{u}$ the piecewise linear functions built interpolating respectively $\{(t_i, \varphi(t_i))\}$ and $\{(t_i, u(t_i))\}$ for $i = 1, \ldots, n$. We point out $\tilde{u}$ cannot be an attainable solution, since all the solutions $u(c)$ of (5.40) are such that $u(c) \in H^2(0,1) = \{f \in L^2(0,1) | D^\alpha f \in L^2(0,1) \, \forall \alpha : |\alpha| \leq 2\}$ for all $c \in L^2(0,1)$. We look for a piecewise linear approximation to c. We discretize the problem using finite differences. Let us denote with L the matrix arising from the discretization of the differential operator $-au_{xx}$ on the grid $x_i = (i-1)h$, $h = 1/(N-1)$, $i = 1, \ldots, N$. We choose $N = 113$ and $n = 39$. Let $\bar{\varphi}, \bar{u} \in \mathbb{R}^N$ be such that $\bar{\varphi}_i = \tilde{\varphi}(x_i)$, $\bar{u}_i = \tilde{u}(x_i)$, $i = 1, \ldots, N$ and define for $c \in \mathbb{R}^N$ $F(c) = (L + \texttt{diag}(c))^{-1}\bar{\varphi}$, with $\texttt{diag}(c) \in \mathbb{R}^{n \times n}$ and we solve

$$\min_{c \in \mathbb{R}^N} \frac{1}{2} \|F(c) - \bar{u}\|^2. \tag{5.41}$$

If we denote with c^* the solution approximation found with exact data, it holds $\|F(c^*) - \bar{u}\| \sim 1.e-3$. For this test problem the exact form of the the Jacobian matrix of F is given by:

$$J(c) = -(L + \texttt{diag}(c))^{-1}(\texttt{diag}(F(c))). \tag{5.42}$$

- `R2: A 2D parameter identification problem.` We consider the 2D version of problem `R1` with $a = 1$. Namely we want to reconstruct c in the 2D-elliptic problem

$$-\Delta u + cu = \varphi \text{ in } \Omega \tag{5.43a}$$

$$u = \zeta \text{ on } \partial\Omega \tag{5.43b}$$

from the knowledge of u in $\Omega = (0,1) \times (0,1)$, $\varphi \in L^2(\Omega)$ and ζ the trace of a function in $H^2(\Omega)$. This problem has been widely studied, see for example [92, 98].

We consider the discretized version of the arising nonlinear least squares problem, obtained as described in [92]. Namely problem (5.43a)-(5.43b) was discretized using finite differences choosing as grid points $x_i = y_i = \frac{i-1}{n-1}$, for $i = 1,\dots,n$ and $n = 50$, and using lexicographical ordering, denoted by $l : \{1,\dots,n^2\} \to \{1,\dots,n^2\}$. Let us denote by A the matrix arising from the discretization of the Laplacian operator, with $\bar{\varphi} = [\bar{\varphi}_1,\dots,\bar{\varphi}_{n^2}]^T$, where $\bar{\varphi}_{l(i,j)} = \varphi(x_i, y_j)$. Moreover for $c \in \mathbb{R}^{n^2}$ we define $F(c) = (A + \mathtt{diag}(c))^{-1}\bar{\varphi}$. Then, $F : \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$, and the resulting discrete problem is a nonlinear least squares problem of size $n^2 = 2500$:

$$\min_{c\in\mathbb{R}^{n^2}} \frac{1}{2}\|F(c) - \bar{u}\|^2,$$

for a given $\bar{u} \in \mathbb{R}^{n^2}$. For further details see [92]. Our experiments were conducted choosing as a parameter to be identified

$$c(x,y) = 1.5\sin(4\pi x)\sin(6\pi y) + 3((x-0.5)^2 + (y-0.5)^2) + 2.$$

The solution $u(x,y)$ of (5.43) corresponding to this choice of $c(x,y)$ is $u(x,y) = 16x(1-x)y(y-1)+1$. Function φ in (5.43) has been defined from (5.43a) and the data $\bar{u}$ are artificially set as a perturbation of $[u_1,\dots,u_{n^2}]$ with $u_{l(i,j)} = u(x_i, y_j)$, to let $c^\dagger = [c_1^\dagger,\dots,c_{n^2}^\dagger]^T$, where $c_{l(i,j)}^\dagger = c(x_i,y_j)$, be a stationary point with strictly positive residual. Specifically $\|J(c^\dagger)^T(F(c^\dagger) - \bar{u})\| = 0$ and $\|F(c^\dagger) - \bar{u}\| \simeq 0.1$, for J the Jacobian matrix of F.

For this test problem the exact form of the the Jacobian matrix of F is given by:

$$J(c) = -(A + \mathtt{diag}(c))^{-1}(\mathtt{diag}(F(c))). \tag{5.44}$$

- `R3: A test problem arising in geophysics [25].` Starting from electromagnetic data collected by a ground conductivity meter (GCM), the aim is to reconstruct the electrical conductivity x of the soil with respect to depth z. The GCM contains two small coils, a transmitter and a receiver, whose axes can be aligned either vertically or horizontally with respect to the ground surface. An alternating sinusoidal current in the transmitter produces a primary magnetic field , which induces small eddy currents in the subsurface. These currents, in turn, produce a secondary magnetic field, which is measured, together with the primary field, at the receiver. The ratio of the secondary to the primary magnetic fields is then used to estimate the conductivity of the subsurface. Starting from this ratio, one can obtain the predicted values of the apparent conductivity measurement $m^V(x,h)$ (vertical orientation of coils) and $m^H(x,h)$ (horizontal orientation of coils) at height h above the ground, which depend on the value x of the conductivity. The nonlinear model employed is the one described in [106, 108], and further analyzed and adapted to the case of a GCM in [57], which is derived from Maxwell's equations, keeping in mind the cylindrical symmetry of the problem, [25]. See [25, §2] for a more detailed description. We assume the soil to be divided in n layers, so that x_i is the conductivity in each layer and $x = (x_1,\dots,x_n)^T$. Multiple measurements are needed to recover the distribution of conductivity with respect to depth. In order to obtain such measurements, we assume to use the two admissible loop orientations and to record apparent conductivity at height $h_i, i = 1,\dots,m$. This generates 2 sets of m values: $b^V = (b_1^V,\dots,b_m^V)$ and $b^H = (b_1^H,\dots,b_m^H)$. Let us denote by $r(x)$ the error in the model prediction:

$$r(x) = b - m_C(x), \qquad b = \begin{bmatrix} b^V \\ b^H \end{bmatrix}, \qquad m_C(x) = \begin{bmatrix} m^V(x,h) \\ m^H(x,h) \end{bmatrix}.$$

The problem of data inversion consists of computing the conductivity x solving

$$\min_x \frac{1}{2}\|r(x)\|^2.$$

We assume that the conductivity distribution is a function of the depth, $x = \phi(z)$. In our experiments we used the piecewise linear function

$$\phi(z) = \begin{cases} \frac{8z+1}{5} & \text{if } z \le 0.5, \\ \frac{-2z+6}{5} & \text{if } z > 0.5, \end{cases}$$

expressed in Siemens/meter, with respect to the depth z, measured in meters. This implies the presence of a strongly conductive material at a given depth. We assume the measurements to be taken at different heights $h_i = (i-1)\bar{h}$ above the ground, $i = 1,\dots,m$, for a chosen height step $\bar{h}$. We divide the soil into $n = 60$ layers, up to the depth of 2.5 meters, each of thickness $\bar{d} = 2.5/(n-1)$, selecting different depths under the ground level, $[z_1,\dots,z_n]$, where we let $z_j = (j-1)/\bar{d}$, $j = 1,\dots,n$. We apply our method to synthetic data sets. We generate synthetic measurements at $m = 40$ equispaced heights up to 1.9 meters to let $x^\dagger = (\phi(z_1),\dots,\phi(z_n))$ be a stationary point such that $\|m_C(x^\dagger) - b\| \simeq 0.48$. Note that we are approximating the true electrical conductivity with a mathematical model, so it is reasonable to expect it to fit the data with a nonzero residual, even in the case of exact data. On this test problem also bound constraints are present, as the solution must be positive. Then, we employ the projection strategy described in Section 5.4.

- R4: A fitting of a sum of two exponentials. Given the model

$$y(t) = x_1 e^{-x_2 t} + x_3 e^{-x_4 t}, \tag{5.45}$$

  we would like to recover the set of parameters $x^\dagger$ solving the following discrete least squares problem:

$$\min_{x=[x_1,x_2,x_3,x_4]^T} \frac{1}{2}\|F(x)-y\|^2, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \quad F(x) = \begin{bmatrix} x_1 e^{-x_2 t_1} + x_3 e^{-x_4 t_1} \\ \vdots \\ x_1 e^{-x_2 t_m} + x_3 e^{-x_4 t_m} \end{bmatrix},$$

  where the observations (t_i, y_i) $i = 1,\dots,m$, are given as follows. The points t_i are equispaced in $[0,10]$ and y_i have been chosen to let $x^\dagger = [0.2, -5, 0.4, -100]^T$ be a minimum of the problem with nonlinear residual $\|F(x^\dagger) - y\|$ of the order of 0.54, when we fit the data with model (5.45). The experiments were conducted choosing $m = 1000$.

In the following, for uniformity of notation, for all the tests we assume that the minimization problem to be solved is

$$\min_x \frac{1}{2}\|F(x)-y\|^2$$

and we will denote with x^* the solution approached when the minimization problem is solved with exact data.

First of all, our method relies on Assumption 5.6 for the proof of the regularizing properties. In Chapter 6 we will prove that it holds for a slightly more general version of problem (5.38). Here, we report numerical evidence for the assumption, on all the test problems. Specifically, in Figure 5.2 we plot $\|\nabla f(\tilde{x}) - \nabla f(x_k) - J(x_k)^T J(x_k)(\tilde{x} - x_k)\|$ (dashed line) and $(c\|\tilde{x} - x_k\| + \sigma)\|\nabla f(\tilde{x}) - \nabla f(x_k)\|$ (solid line), where $\tilde{x} \in B_\rho(x_k)$. We have chosen $c = 0.1$, $\rho = 0.3$ and σ equal to the residual $\|F(x^*) - y\|$. We notice that condition

$$\|\nabla f(\tilde{x}) - \nabla f(x_k) - J(x_k)^T J(x_k)(\tilde{x} - x_k)\| \le (c\|\tilde{x} - x_k\| + \sigma)\|\nabla f(\tilde{x}) - \nabla f(x_k)\|,$$

is satisfied for x_k approaching x^*. We repeated these tests varying $\tilde{x} \in B_\rho(x_k)$ and we obtained qualitatively the same results.

We are then ready to describe the practical implementation of the method. All procedures were implemented in MATLAB and run using MATLAB 2015A on an Intel Core(TM) i5-2467M 1.6 GHz, 4 GB RAM; the machine precision is $\epsilon_m \sim 2 \cdot 10^{-16}$. The Trust-Region procedure was implemented according to Algorithm 5.1.

The major implementation issues are as follows. Regarding the Jacobian matrix of F the analytical expression was used for all test problems. Specifically for problems R1, R2 the exact Jacobian matrices are given in (5.42) and (5.44), for problem R3 the exact analytical formulae is developed in [25] and for problem R4 it is easily computable.

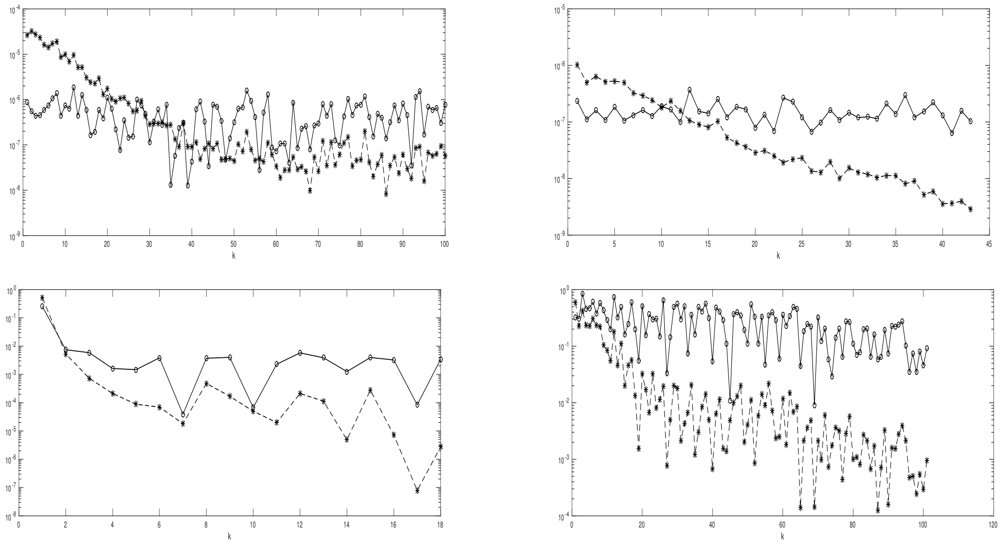


Figure 5.2: *Numerical evidence for Assumption 5.6: plot of* $\|\nabla f(\tilde{x})-\nabla f(x_k)-J(x_k)^T J(x_k)(\tilde{x}-x_k)\|$ *(dashed line) and* $(0.1\|\tilde{x}-x_k\|+\sigma)\|\nabla f(\tilde{x})-\nabla f(x_k)\|$ *(solid line) for* $\tilde{x}$ *randomly chosen in* $B_\rho(x_k)$*,* $\rho=0.3$*, for problem* `R1` *(top left,* $\sigma=4\cdot 10^{-3}$*),* `R2` *(top right,* $\sigma=0.1$*),* `R3` *(bottom left,* $\sigma=0.48$*),* `R4` *(bottom right,* $\sigma=0.54$*).*

To compute the square root of matrix B_k we use the singular value decomposition of the Jacobian provided by MATLAB function `svd`.

In case of noisy problems, given the error level δ, the exact data y was perturbed by normally distributed values with mean 0 and variance δ^2 using the MATLAB function `randn`.

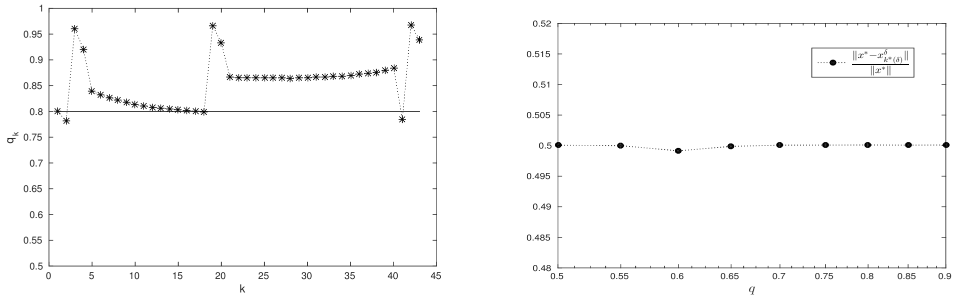
To compute the KKT point (z_k,λ_k) at Step 2 we need to solve (5.17). To compute the square root of B_k we need to know its singular value decomposition, that we compute with the Matlab function `svd`. The computation of λ_k can be accomplished solving the following nonlinear scalar equation:

$$\psi(\lambda)=\frac{1}{\|z(\lambda)\|}-\frac{1}{\Delta_k},$$

since the Trust Region is ensured to be active, see Section 2.2.1.1. Typically high accuracy in the solution of the above scalar equations is not needed, hence we terminated the Newton process as soon as $|\Delta_k-\|z(\lambda)\||\le 10^{-2}\Delta_k$ [22, §7.3.10]. The linear systems we have to deal with in the Newton method take the form $B_k^2+\lambda_k I$. Taking into account that λ_k is always bounded away from zero, this allows to overcome the ill-conditioning of B_k^2 that arises from the smallest singular values close to zero.

Algorithms 5.1 and 5.2 were run setting $\eta=10^{-1}$. In Step 1 the Trust-Region radius was updated as follows

$$\begin{aligned}
\Delta_0 &= \mu_0\|B_k^{1/2}g_k\| & \mu_0&=10^{-1}\\
\Delta_{k+1} &= \mu_{k+1}\|B_{k+1}^{1/2}g_{k+1}\|, & \mu_{k+1}&=\begin{cases}\frac{1}{6}\mu_k & \text{if } q_k<q \text{ or } \rho_k<\eta_2\\ 2\mu_k & \text{if } q_k>\nu q \text{ and } \rho_k>\eta_2\\ \mu_k & \text{otherwise}\end{cases},
\end{aligned}$$



(a) **(b)**

Figure 5.3: *(a): Test problem* R1, *$\delta = 10^{-2}$. Obtained values of q_k. (b): Test problem* R4, *$\delta = 10^{-2}$. Dependence of the relative error between computed solution $x^\delta_{k^*(\delta)}$ and x^* on the parameter q.*

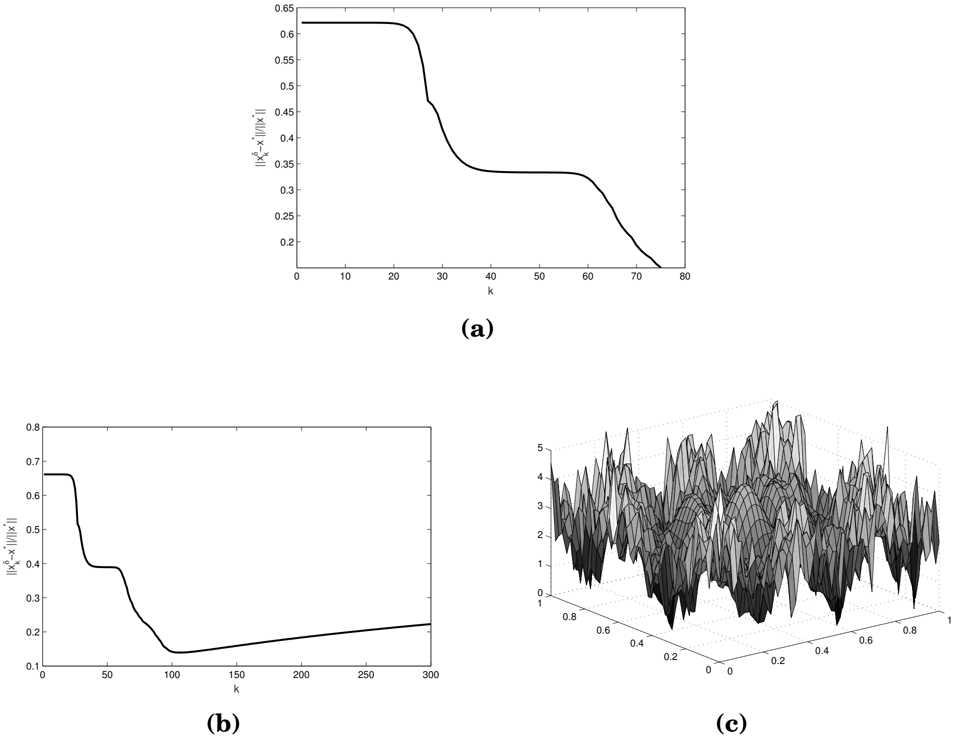
with $q_k = \frac{\|B_k p_k + g_k\|}{\|g_k\|}$, $\nu = 1.1$ and $\eta_2 = 0.25$. The maximum and minimum values for Δ_k were set to $\Delta_{\max} = 10^4$ and $\Delta_{\min} = 10^{-12}$ and the maximum value for μ_k was set to 10^5. This updating strategy is analogous to the one used in Chapter 4 and it is based on the same considerations, with the only exception that the Trust-Region radius is proportional to $\|B_k^{1/2} g_k\|$ rather than to the norm of the residual, see Section 4.4. This updating strategy turned out to be efficient in practice. As an example we report in Figure 5.3 (a) the obtained values of q_k for problem R1 for $\delta = 10^{-2}$. We can see that for almost all the iterations the values q_k are greater than the chosen value $q = 0.8$, marked by the horizontal solid line, so the generalized q-condition is fulfilled.

The free parameter q was set equal to 0.8, but this choice is not critical. Actually, the behaviour of the procedure does not seem to be deeply affected by the value of q. As an example we show in Figure 5.3 (b) for the test problem R4 the value of the relative error $\frac{\|x^* - x^\delta_{k^*(\delta)}\|}{\|x^*\|}$ between the solution x^* approached with exact data and that computed with $\delta = 10^{-2}$, for different values of parameter q. We notice that the error does not vary significantly depending on q.

The scalar τ in the discrepancy principle (4.7) was chosen adaptively as $\tau_k = \bar{\tau}\|J(x^\delta_k)\|$, with $\bar{\tau} = 0.1$. The value of $\bar{\tau}$ is not in agreement with Assumption 5.13, but in practice σ is not known and the numerical tests provide an evidence of the effectiveness of this stopping rule. As τ_k depends on k, the stopping rule changes at each iteration. However, τ_k varies only slightly along the iterations as $\|J(x^\delta_k)\|$ is almost constant. Values of τ_k used in our tests, for $\delta = 10^{-2}$, are as follows:

- R1 : $\|J(x^\delta_k)\| \simeq 0.04$ and $\tau_k \simeq 4 \cdot 10^{-3}$,
- R2 : $\|J(x^\delta_k)\| \simeq 10^{-2}$ and $\tau_k \simeq 10^{-3}$,
- R3 : $\|J(x^\delta_k)\| \simeq 10^{-1}$ and $\tau_k \simeq 10^{-2}$,
- R4 : $\|J(x^\delta_k)\| \simeq 5$ and $\tau_k \simeq 0.5$.

In agreement with the theory, the error is monotonically decreasing as long as the discrepancy principle is not satisfied, as we show for example in Figure 5.4 (a),



(a)

(b) **(c)**

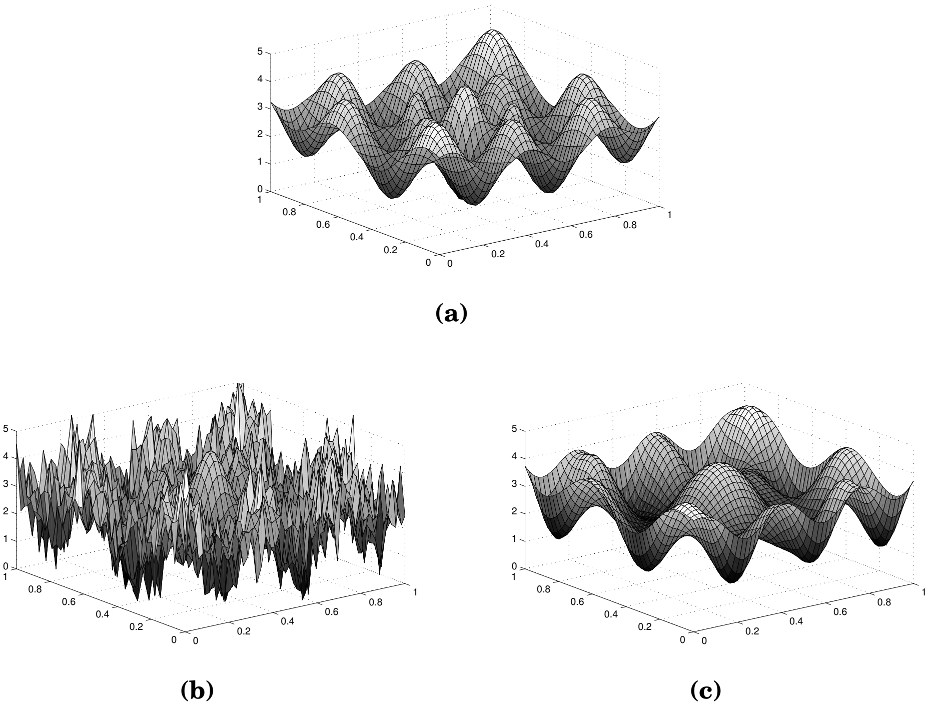
Figure 5.4: *Test* R2, $\delta = 10^{-2}$. *Upper part: The procedure is stopped when the discrepancy principle is satisfied, the norm of the error decreases monotonically (a). Lower part: The procedure is not stopped when the discrepancy principle is satisfied, the norm of the relative error increases (b) and a solution of the noisy problem is approached (c).*

in which we report the decrease of the relative error $\frac{\|x^* - x_k^\delta\|}{\|x^*\|}$ between the solution approached with exact data and the current iterate x_k^δ, varying k for problem R2 with $\delta = 10^{-2}$. Iterating further is not useful and it can also lead to a new increase in the norm of the error, as it is shown at the bottom of Figure 5.4. The procedure was not stopped when the discrepancy principle was satisfied and the error started to increase (b), as the sequence approaches a solution of the noisy problem (c).

The stopping criterion alone is not sufficient to obtain a regularizing method. To show this, we implemented a standard Trust-Region procedure, analogously to Section 4.4, see (4.46).

In Figure 5.5 we compare for problem R2 with $\delta = 10^{-2}$ the solution x^* approached with exact data (a), and the solution approximations computed by the standard (b) and the regularizing Trust-Region (c) approaches. It is clear that while the elliptical regularizing Trust-Region method manages to handle the noise in the data, the sequence generated by the standard Trust-Region method converges to a solution of the noisy problem.

In Figures 5.6 and 5.7 we study the behaviour of the method depending on the noise level. In Figure 5.6 we report the contour plots of the computed solution approximations for problem R2 with $\delta = 0, 10^{-2}, 10^{-4}$. We can see that as the noise level decreases, according to the theory, the computed solution approaches a minimum of the problem. In fact, as δ decreases, the contour plots become more and



(a)

(b)

(c)

Figure 5.5: *Test* R2*. Upper part: solution approximation x^* obtained with exact data (a). Lower part: solution approximations obtained with standard Trust-Region (b) and elliptical regularizing Trust-Region (c) for $\delta = 10^{-2}$.*

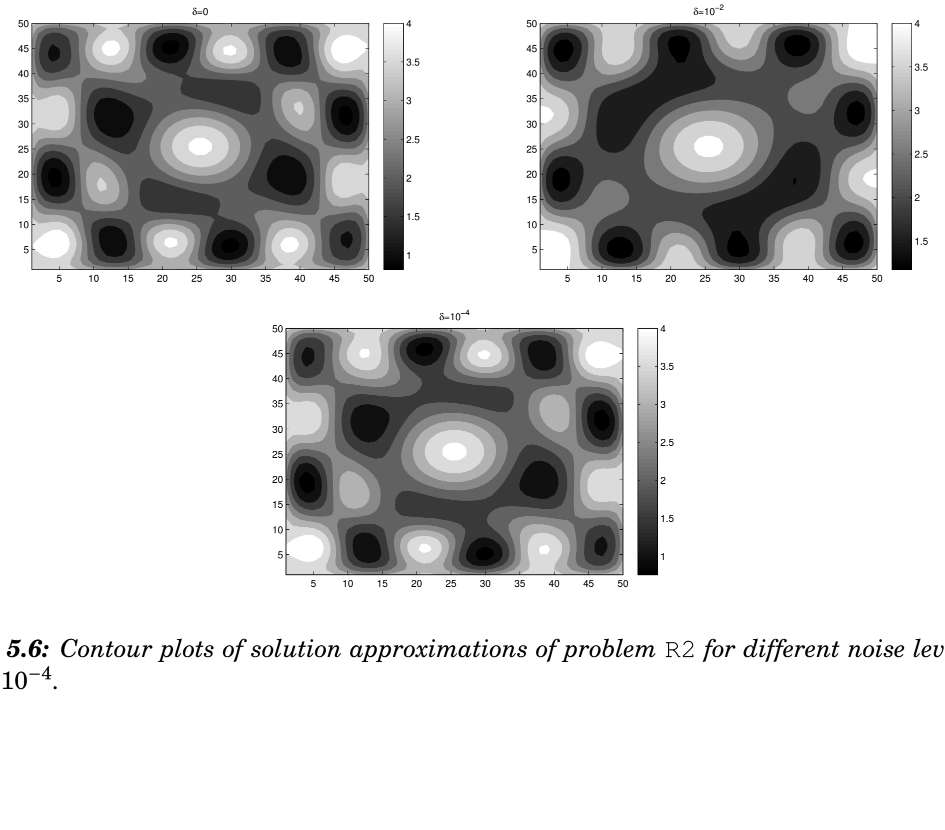
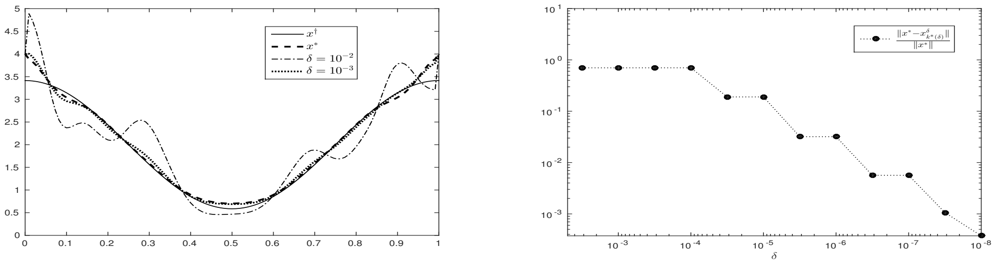
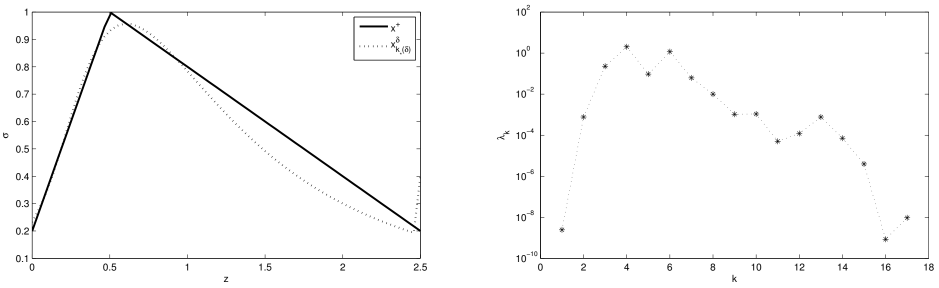


Figure 5.6: *Contour plots of solution approximations of problem* R2 *for different noise levels, $\delta = 0, 10^{-2}, 10^{-4}$.*



(a) **(b)**

Figure 5.7: *(a): Problem* R1*. Comparison of solution approximations computed with $\delta = 10^{-3}, 10^{-2}$ and solutions x^* and $x^\dagger = [c(x_1),\ldots,c(x_N)]^T$ with c in (5.39). (b): Problem* R4*. Reduction of the relative error $\frac{\|x^*-x^\delta_{k^*(\delta)}\|}{\|x^*\|}$ with the noise level δ.*



(a) **(b)**

Figure 5.8: *Test problem* R3 *with $\delta = 10^{-2}$: plot of the true solution $x^\dagger$ and of the computed solution $x^\delta_{k^*(\delta)}$ for $\delta = 10^{-2}$ (a), regularization parameters λ_k (b).*

more similar to that obtained for $\delta = 0$. In Figure 5.7 (a) we consider problem R1. We compare the solution approximations computed with $\delta = 0$ (x^*, dashed line), $\delta = 10^{-3}$ (dotted line) and $\delta = 10^{-2}$ (dash-dotted line), and $x^\dagger = [c(x_1),\ldots,c(x_N)]^T$ with c in (5.39) (solid line). We notice that the solution approximation improves with decreasing noise and that x^*, which is computed with exact data, is a good approximation to $x^\dagger$. In Figure 5.7 (b) we observe for problem R4 the reduction of the relative error $\frac{\|x^*-x^\delta_{k^*(\delta)}\|}{\|x^*\|}$ with the noise level δ.

In Figure 5.8 we consider test problem R3 with $\delta = 10^{-2}$. We report the plot of the true solution $x^\dagger$ and that of the computed solution $x^\delta_{k^*(\delta)}$ for $\delta = 10^{-2}$ (a), and the plot of the computed regularization parameters λ_k (b). We can see that the regularized solution is a good approximation of $x^\dagger$ and that, in accordance to the theory, the regularization parameters are strictly positive and bounded above.

5.6 Chapter conclusion

In this chapter we considered a wider class of problems than that considered in the previous one. We addressed the solution of problems of the form (3.1) for which a solution of the associated nonlinear system may not exist, so that the residual at any local minimum $x^\dagger$ will be $\|F(x^\dagger) - y\| \geq 0$. We proposed a non-stationary iterated Tikhonov procedure to solve nonlinear ill-posed least squares problems with small residual. This represents an improvement in the study of ill-posed nonlinear problems, as we are not aware of other methods specifically designed for this class of problems. As for the zero residual case we provided a Trust-Region reformulation of the method that allows us to set the regularization parameters λ_k in an automatic way. Along with a suitable choice of matrix L_k this ensures regularizing properties of the method and gives rise to a procedure able to find a stable approximation of a solution of the unperturbed problem, even in case of noisy data. We provided a theoretical analysis and a reliable implementation of the proposed method that has been validated on different problems. The obtained numerical results highlight the effectiveness of the procedure and its regularizing properties.

We point out that the method proposed in this chapter is suitable also for zero residual problems. However, if the problem is known to have zero residual it is more advisable to use the method presented in Chapter 4, as it based on conditions on the residual norm rather than on the norm on the gradient. Also, this method does not require the computation of the SVD of the Jacobian and it is then cheaper.

Acknowledgments We thank the authors of [25] for providing us the Matlab code for test problem `R3`.

CHAPTER

6 Infinite dimensional setting

Many inverse problems are naturally formulated as infinite dimensional problems. This is the case for example for Fredholm integral equations or parameter identification problems [21, 48, 98]. A large part of the literature on ill-posed inverse problems deals indeed with solution methods designed in Hilbert spaces. The following is the setting that is typically considered. Given Hilbert spaces $\mathscr{X}, \mathscr{Y}$ with inner products $< \cdot, \cdot >$ and norm $\|\cdot\|$, $F : \mathscr{D}(\mathscr{F}) \subseteq \mathscr{X} \to \mathscr{Y}$ is a nonlinear map defined on $\mathscr{D}(F)$, which is assumed to be an infinite dimensional set. The following problem is considered:

$$\min_{x \in \mathscr{D}(F)} f(x) = \frac{1}{2}\|F(x) - y\|^2, \tag{6.1}$$

that is assumed to be ill-posed, in the sense that the solutions do not depend continuously on the data. This is the case for example when F has compact Fréchet derivative F'.

Many different possibilities arise in the solution of such problem. Many authors propose procedures to deal directly with the infinite dimensional problem [31, 50, 54, 67], others are concerned with the solution of projected version of the continuous problem into finite dimensional subsets of the considered Hilbert space [47, 72, 83]. In this latter case it is still possible to choose whether to consider finite approximations of just one between $\mathscr{X}, \mathscr{Y}$ [66, 96] or to discretize both spaces [47, 48, 104]. In this cases usually a sequence of nested subspaces is considered, and convergence of the sequence of solutions of the finite dimensional problems to a solution of the infinite dimensional problem is studied.

In the following sections we consider then these two possible scenarios applied to the approaches presented in this thesis. More precisely, in Section 6.1 we show that the procedures that we have presented in the finite dimensional setting can be easily adapted to an infinite dimensional setting. In Section 6.2 we consider a sequence of solutions of finite dimensional problems, got projecting the infinite dimensional one onto a sequence of nested finite dimensional subspaces. We prove convergence of such a sequence to a solution of the infinite dimensional problem.

Finally, in Section 6.3 we consider Assumption 5.6. The methods in Chapters 4 and 5 rely respectively on Assumption 4.12 and Assumption 5.6 for the proof of their regularizing properties. While Assumption 4.12 is widely assumed in the literature and it has been proved for many infinite dimensional problems, see for example [31, Example 11.1] or [54], we introduced Assumption 5.6 in [S2]. We consider then a continuous model problem and we prove that Assumption 5.6 holds for it.

6.1 Procedures in the Hilbert setting

In this section we briefly point out how the procedures presented in a finite dimensional setting in the previous chapters can be extended to a infinite dimensional Hilbert setting. We particularly focus on what differs from the finite dimensional case. Theoretical results analogous to those established in the finite dimensional case can be proved, we do not repeat the proofs as their adaptation is straightforward. See for example [S2] for the extension of the method presented in Chapter 5 to an infinite dimensional Hilbert setting.

The underlying method is the same as in the finite dimensional case. The step is found solving a minimization problem, that for zero and nonzero residual problems would respectively be:

$$\min_p \frac{1}{2}\|F(x_k^\delta)-y^\delta+F'(x_k^\delta)p\|^2+\frac{\lambda_k}{2}\|p\|^2,$$
$$\min_p \frac{1}{2}\|F(x_k^\delta)-y^\delta+F'(x_k^\delta)p\|^2+\frac{\lambda_k}{2}\|L_k^{\frac{1}{2}}p\|^2,$$

for F' the Fréchet derivative of F and $L_k:\mathscr{X}\to\mathscr{X}$ a symmetric and positive definite regularizing operator.

Notations (3.6) are replaced by

$$B_k=F'(x_k^\delta)^*F'(x_k^\delta), \qquad\qquad f_k'=F'(x_k^\delta)^*(F(x_k^\delta)-y^\delta),$$

where F'^* denotes the adjoint operator of F'.

Given the Trust-Region radius $\Delta_k>0$ and the current iterate x_k^δ, the Trust-Region subproblems (4.21) and (5.16), solved at generic iteration k become respectively:

$$\begin{aligned}&\min_p \|F(x_k^\delta)-y^\delta+F'(x_k^\delta)p\|^2,\\ &\text{s.t. } \|p\|\le\Delta_k,\end{aligned}$$

and

$$\begin{aligned}&\min \frac{1}{2}<z,B_k^2z>+<B_k^{1/2}f_k',z>+f_\delta(x_k^\delta),\\ &\text{s.t. } \|z\|\le\Delta_k.\end{aligned}$$

Both problems have a unique solution from [1, Theorems 9.2.7, 10.2.15, 10.3.4], which is not as straightforward as in the finite dimensional case, as the closed ball in a infinite dimensional Hilbert space will not be compact. They can be solved respectively looking for a couple $(\lambda_k,p(\lambda_k))\in\mathbb{R}^+\times\mathscr{X}$ solution of the KKT conditions

$$
\begin{aligned}
&(F'(x_k^\delta)^*F'(x_k^\delta)+\lambda I)p(\lambda) = -F'(x_k^\delta)^*(F(x_k^\delta)-y^\delta),\\
&\lambda(\|p(\lambda)\|-\Delta_k)=0,\\
&\lambda \geq 0,\\
&\|p\| \leq \Delta_k,
\end{aligned}
$$

and for a couple $(\lambda_k, z(\lambda_k)) \in \mathbb{R}^+ \times \mathscr{X}$ solution of the KKT conditions

$$
\begin{aligned}
&(F'(x_k^\delta)^*F'(x_k^\delta)+\lambda I)z(\lambda) = -B_k^{\frac{1}{2}}f_k',\\
&\lambda(\|z(\lambda)\|-\Delta_k)=0,\\
&\lambda \geq 0,\\
&\|z(\lambda)\| \leq \Delta_k,
\end{aligned}
$$

which is used to set $p(\lambda_k) = B_k^{\frac{1}{2}}z(\lambda_k)$.

As in the finite dimensional setting, with the same choice of the Trust-Region radius, the step guarantees monotonic decrease of the norm of the error and the above proved regularizing properties. Indeed, all the results previously presented are easily extensible to this setting. For example, all the theoretical results obtained through the singular-value decomposition of the Jacobian matrix $J(x_k^\delta)$ can be repeated employing the singular value expansion of $F'(x_k^\delta)$. We indicate with $(\sigma_n; e_n, f_n)$, $n \in \mathbb{N}$, the singular value expansion of $F'(x_k^\delta)$, where $\{e_n\}_{n\in\mathbb{N}}$ and $\{f_n\}_{n\in\mathbb{N}}$ are a complete orthonormal system of eigenvectors for $F'(x_k^\delta)^*F'(x_k^\delta)$ and $F'(x_k^\delta)F'(x_k^\delta)^*$ respectively, and $\sigma_n > 0$ are written down in decreasing order with multiplicity, with 0 being the only accumulating point for the sequence $\{\sigma_n\}_{n\in\mathbb{N}}$ when $\dim \mathscr{R}(F'(x_k^\delta)) = \infty$. Then the following equalities hold:

$$
F'(x_k^\delta)h = \sum_{n=1}^{\infty} \sigma_n < h, e_n > f_n,\ h \in \mathscr{X}, \qquad F'(x_k^\delta)^*h = \sum_{n=1}^{\infty} \sigma_n < h, f_n > e_n,\ h \in \mathscr{Y}.
$$

Moreover, as $F'(x_k^\delta)$ is compact, its Moore-Penrose pseudoinverse can be defined as [31, §2.1]

$$
\begin{aligned}
F'(x_k^\delta)^\dagger h &= \sum_{n=1}^{\infty} \sigma_n^{-1} < h, f_n > e_n,\ \ h \in \mathscr{D}(F'(x_k^\delta)^\dagger),\\
\mathscr{D}(F'(x_k^\delta)^\dagger) &= \{h \in \mathscr{Y} \mid \sum_{n=1}^{\infty} \sigma_n^{-2} |< h, f_n >|^2 < \infty\}.
\end{aligned}
$$

Because for $h \in \mathscr{X}$, $B_k h = \sum_{n=1}^{\infty} \sigma_n^2 < h, e_n > e_n$, then also

$$
\begin{aligned}
B_k^\dagger h &= \sum_{n=1}^{\infty} \sigma_n^{-2} < h, e_n > e_n,\ h \in \mathscr{D}(B_k^\dagger),\\
\mathscr{D}(B_k^\dagger) &= \{h \in \mathscr{X} \mid \sum_{n=1}^{\infty} \sigma_n^{-2} |< h, f_n >|^2 < \infty\},
\end{aligned}
$$

With these results we can repeat all the proofs in the finite dimensional setting, that employ the singular value decomposition. All the other theoretical results follow as straightforward adaptations of the presented proofs.

6.2 Convergence to a solution of the infinite dimensional problem

When an inverse problem is considered, even if it is originally formulated in a Hilbert setting, for numerical computation one has to approximate the considered space by a sequence of finite dimensional subspaces. Then, another interesting topic is to consider the sequence of solutions of the finite dimensional problems got projecting the original problem onto a sequence of finite dimensional subspaces of increasing dimension. It is interesting to investigate if such sequence converges to a solution of the infinite dimensional problem, when the dimension of the subspaces tends to infinity.

Usually indeed, one assumes to consider two sequences of finite dimensional subspaces, $\{\mathscr{X}_n\} \subseteq \mathscr{X}$ and $\{\mathscr{Y}_m\} \subseteq \mathscr{Y}$ such that

$$\begin{aligned} &\mathscr{X}_{n+1} \subseteq \mathscr{X}_n, \qquad && |\mathscr{X}_n| = n, \\ &\mathscr{Y}_{m+1} \subseteq \mathscr{Y}_m, \qquad && |\mathscr{Y}_m| = m, \end{aligned}$$

where $|\cdot|$ denotes the space dimension. Then, defined $P_n : \mathscr{X} \to \mathscr{X}_n$ and $Q_m : \mathscr{Y} \to \mathscr{Y}_m$ the projection operators, one considers a sequence of such problems:

$$\min_x \frac{1}{2} \|Q_m(F(P_n x) - y)\|^2_{\mathscr{Y}}. \tag{6.4}$$

In general the sequence of projection operators $\{Q_m\}, \{P_n\}$ are assumed to pointwise converge to the identity:

$$\forall f \in \mathscr{Y} \;\; Q_m f \to f \;\text{ as } m \to \infty, \qquad \forall f \in \mathscr{X} \;\; P_n f \to f \;\text{ as } n \to \infty.$$

This latter condition implies uniform boundedness of the two operators.

For example, if $\mathscr{X}, \mathscr{Y} = L^2([a,b])$ we can choose as finite dimensional subspaces those of piecewise linear functions. Assume the nodes $\{z_j\}_{j=1}^{l}$ to partition interval $[a,b]$ into $l-1$ subintervals $I_j = [z_j, z_{j+1}]$ for $j = 1, \dots, l-1$. The space of piecewise linear functions arising from this partition is defined as:

$$\mathscr{P}_l = \{f \in C^0([a,b]) \,|\, f \text{ is linear in } I_j \text{ for each } j = 1, \dots, l-1\}.$$

The dimension of such space is l. We define then as Q_m, P_n the projection operator onto $\mathscr{P}_m, \mathscr{P}_n$ respectively. We assume then to use the procedure presented in Section 6.1 choosing $\mathscr{X} = \mathscr{P}_n$ and $\mathscr{Y} = \mathscr{P}_n$.

We want to study the convergence of the sequence of solutions found solving the projected problems to a solution of the infinite dimensional one when both n, m go to infinity.

We denote with $\hat{x}^*_{n,m}$ the solution found of problem (6.4) with $\mathscr{X} = \mathscr{P}\setminus$ and $\mathscr{Y} = \mathscr{P}_{\Updownarrow}$.

We assume F to be compact and weakly sequentially closed and we consider the noise free case.

We perform a local convergence analysis and we assume to have a starting guess $\hat{x}_0 \in \mathscr{P}_n$ close enough to a solution of (6.4). Boundedness of the sequence $\{\hat{x}^*_{n,m}\}$ follows from the equivalent version of Theorem 5.12 stated in a Hilbert setting (see for example Theorem 4.4 in [S2]) that states that for fixed n,m, provided that $\hat{x_0}$ is close to a solution of (6.4), it exists $\epsilon > 0$ such that $\hat{x}^*_{n,m} \in B_\epsilon(\hat{x}_0)$, i.e. it holds $\|\hat{x}^*_{n,m} - \hat{x}_0\|_{L^2} \le \epsilon$. Then $\|\hat{x}^*_{n,m} - \hat{x}_0\|_{L^2}$ is bounded for all n and $\{\hat{x}^*_{n,m}\}$ has a weakly convergent subsequence $\{\hat{x}^*_{n,m}\}_k = w_k$. By compactness of F and boundedness of $\|w_k - \hat{x}_0\|_{L^2}$, there exists a subsequence w_m of w_k such that $F(w_m)$ converges strongly to some $f \in \mathscr{Y}$. We want to show that the weak limit x^*_∞ of w_m is a solution of the original problem. It holds:

$$\begin{aligned}\|F'^*(F(w_m)-y)\|_{L^2} &\le \|P_nF'^*(F(w_m)-y)\|_{L^2} + \|(I-P_n)F'^*(F(w_m)-y)\|_{L^2}\\ &\le \|P_nF'^*Q_m(F(w_m)-y)\|_{L^2} + \|P_nF'^*(I-Q_m)(F(w_m)-y)\|_{L^2}\\ &+ \|(I-P_n)F'^*Q_m(F(w_m)-y))\|_{L^2} + \|(I-P_n)F'^*(I-Q_m)(F(w_m)-y)\|_{L^2}.\end{aligned}$$

It holds

$$\begin{aligned}&\|P_nF'^*(I-Q_m)(F(w_m)-y)\|_{L^2} + \|(I-P_n)(I-Q_m)F'^*(F(w_m)-y)\|_{L^2}\\ &\le (\|P_nF'^*\| + \|(I-P_n)F'^*\|)\|I-Q_m\|(\|F(w_m)-f\|_{L^2} + \|f-y\|_{L^2}) \xrightarrow{m\to\infty} 0\end{aligned}$$

from the pointwise convergence of Q_m to the identity, the strong convergence of $F(w_m)$ and the boundedness of $\|P_nF'^*\| + \|(I-P_n)F'^*\|$. With the same reasoning also $\|(I-P_n)F'^*(I-Q_m)(F(w_m)-y)\|_{L^2}$ tends to zero as m tends to infinity. Finally, as $\hat{x}^*_{n,m}$ is the solution to (6.4), $\|P_nF'^*Q_m(F(w_m)-y)\|_{L^2} = 0$ for fixed n,m. Then $\|F'^*(F(w_m)-y)\|_{L^2}$ tends to zero as n,m tend to infinity. Due to the weak sequential closedness of F, $x^*_\infty \in D(F)$ and $F'^*(F(x^*_\infty)-y) = 0$ so that x^*_∞ is a solution of th infinite dimensional problem.

The proof is the same in case we consider a sequence of data $\{y_l\}$ such that $\|y_l - y\| \le \delta_l$ for $\{\delta_l\}$ a sequence of noise levels tending to zero.

6.3 A model problem

In this section we consider a slightly more general version of the parameter identification problem (5.38) introduced in Section 5.5 and we prove that Assumption 5.6 holds for it.

Where not differently specified $\|\cdot\|$ indicates the L^2 norm. We drop the domain from all the spaces, as it is assumed to be $(0,1)$, for example $L^2 = L^2(0,1)$. We remind that $H^2 = H^2(0,1) = \{f \in L^2(0,1) | D^\alpha f \in L^2(0,1)\ \forall \alpha : |\alpha| \le 2\}$.

Let us consider the following problem [21]. We want to reconstruct c in the 1D-elliptic problem

$$-(au_x)_x + cu = \varphi \quad \text{in } (0,1) \tag{6.5a}$$

$$R_i u = 0,\ i = 1,2, \tag{6.5b}$$

given $u, \varphi \in L^2$, $a \in C^1$, $a(x) \geq \bar{a} > 0$, $R_i u = \alpha_{i1}u(0) + \alpha_{i2}u'(0) + \alpha_{i3}u(1) + \alpha_{i4}u'(1)$, $\alpha_{ij} \in \mathbb{R}$. Let $\mathscr{A}(c)$ be the differential operator in L^2 associated with (6.5), i.e.

$$\mathscr{A}(c)u = -(au_x)_x + cu, \qquad D(\mathscr{A}) = \{u \in L^2 \,:\, u \in H^2, R_i u = 0, i = 1,2\}.$$

Throughout we make the following assumptions:

(H1) There exist constants $\alpha \geq 0$ and $\xi > 0$ such that $< \mathscr{A}(c)u, u > \geq \xi \|u\|_{H^1}^2$, for all $u \in D(\mathscr{A})$ and $c \in Q = \{c \in L^2 \,|\, c(x) \geq \alpha \, a.e.\}$.

(H2) The boundary conditions R_i $i = 1,2$ in (6.5b) are such that $\mathscr{A}(c)$ is self-adjoint.

Let

$$\mathscr{U} = \{c \in L^2 \,|\, c(x) \geq \alpha \text{ a.e., } \|c\| \leq \gamma\},$$

for $\gamma > \alpha$ and $F : D(F) \to L^2$ be the operator mapping parameter c to the solution u of (6.5), with

$$D(F) = \{c \in L^2 \,|\, \|c - \tilde{c}\| \leq \epsilon, \text{ for some } \tilde{c} \in \mathscr{U}\} \supseteq \mathscr{U}.$$

The following result will be useful in the analysis.

Lemma 6.1. *[21, Lemma 2.1] Let (H1) hold. Then there exist constants $\kappa_1 > 0$ and $\kappa_2 = \|a\|_{C^1} + \epsilon + \gamma$ such that*

$$\kappa_1 \|u\|_{H^2} \leq \|A(c)u\| \leq \kappa_2 \|u\|_{H^2}$$

holds for all $c \in D(F)$ and $u \in D(\mathscr{A})$.

We allow c to vary in a finite dimensional subspace of $\mathscr{U}$, as it is the case in numerical practice. Let $H_N \subset L^\infty$ be a finite dimensional subspace of L^2, and define $\mathscr{U}_N = \mathscr{U} \cap H_N$. Identifying c reduces to solving the following nonlinear problem

$$\min_{c \in \mathscr{U}_N} f(c) = \frac{1}{2}\|F(c) - \tilde{u}\|^2, \tag{6.6}$$

for $\tilde{u}$ the actual observation. We define the set of attainable observations $\mathscr{V}_N$ as $\mathscr{V}_N = \{F(c) \,:\, c \in \mathscr{U}_N\} \subset H^2$. Considering modelling errors, it is not reasonable in general to assume that $\tilde{u} \in \mathscr{V}_N$, [21, p.2].

It can be proven that F is twice continuously Fréchet differentiable (cf. Lemma 2.4 and Lemma 2.5 in [21]). The first Fréchet derivative of F' and its adjoint, for $h \in L^2$, are given by [21]

$$F'(c)h = -\mathscr{A}(c)^{-1}(hu(c)), \qquad F'(c)^* w = -u(c)\mathscr{A}(c)^{-1}w.$$

The second Fréchet derivative $F''(c)(h,k) = \xi(h,k) = \xi$ is the unique solution of $\mathscr{A}(c)\xi = -kF'(c)h - hF'(c)k$ for $h, k \in L^2$ [21].

Let us consider a solution $c^* \in \mathscr{U}_N$ of (6.6). From the Taylor expansion of f' and (5.3) it holds

$$\|f'(c^*+p)-f'(c^*)-F'(c^*)^*F'(c^*)p\| \le$$
$$\left\| \int_0^1 [F'(c^*+tp)^*F'(c^*+tp)-F'(c^*)^*F'(c^*)]p\,dt \right\| + \left\| \int_0^1 S(c^*+tp)p\,dt \right\|$$

If the residual is small enough, it exists $\sigma < 1$ such that $\|S(c)\| \le \sigma$ for all c in a neighbourhood of the solution, $c \in U(c^*) \cap D(F)$. Then,

$$\begin{aligned} &\|f'(c^*+p)-f'(c^*)-F'(c^*)^*F'(c^*)p\| \\ &\qquad \le \int_0^1 \|F'(c^*+tp)^* - F'(c^*)^*]F'(c^*)p\,dt\| \\ &\qquad + \int_0^1 \|F'(c^*+tp)^*[F'(c^*+tp)-F'(c^*)]p\,dt\| + \sigma\|p\|. \end{aligned} \tag{6.7}$$

The proof of the following Lemma follows the lines of [98, Lemma 2.4], which is concerned with the 2D version of problem (6.5) with $a(x) \equiv 1$.

Lemma 6.2. *Let H1 and H2 hold. Then, there exist K_0, K_1 such that for every c and d in a neighbourhood of the solution $U(c^*) \cap D(F)$ and $h \in L^2$*

$$\begin{aligned} \|F'(c)h - F'(d)h\| &\le K_0\|h\|\|c-d\|, \\ \|F'(c)^*h - F'(d)^*h\| &\le K_1\|h\|\|c-d\|. \end{aligned} \tag{6.8}$$

Proof. Let $c, d \in U(c^*) \cap D(F)$. It holds

$$(\mathscr{A}(c)-\mathscr{A}(d))F'(c)h + \mathscr{A}(d)(F'(c)-F'(d))h = (u(d)-u(c))h,$$

so that

$$(F'(c)-F'(d))h = \mathscr{A}(d)^{-1}[(\mathscr{A}(d)-\mathscr{A}(c))F'(c)h + (u(d)-u(c))h] = \mathscr{A}(d)^{-1}w$$

with $w = (c-d)\mathscr{A}(c)^{-1}hu(c) + (u(d)-u(c))h$. Notice that $u(c)$ and $u(d)$ satisfy

$$-(a(u(c)-u(d))_x)_x + c(u(c)-u(d)) = (d-c)u(d)$$

with $u(c)-u(d) \in D(\mathscr{A})$. Thus, $\mathscr{A}(c)(u(c)-u(d)) = (d-c)u(d)$. From Lemma 6.1 and the fact that $u(d) \in L^\infty$, we find

$$\|u(c)-u(d)\|_{L^\infty} = \|\mathscr{A}(c)^{-1}(d-c)u(d)\|_{L^\infty} \le C\|c-d\|,$$

for a positive constant $C > 0$. Then,

$$\begin{aligned} \|w\| &\le \|\mathscr{A}(c)^{-1}hu(c)\|_{L^\infty}\|c-d\| + \|(u(d)-u(c))h\| \\ &\le 2C\|c-d\|\|h\|. \end{aligned}$$

For the second result in (6.8) we consider that

$$F'(c)^*h - F'(d)^*h = -[u(c)(\mathscr{A}(c)^{-1}h - \mathscr{A}(d)^{-1}h) + (u(c)-u(d))\mathscr{A}(d)^{-1}h]$$

and we can proceed analogously as before. □

From (6.8) we obtain

$$\int_0^1 \|[F'(c^*+tp)^* - F'(c^*)^*]F'(c^*)p\|dt \le K(K_1/2)\|p\|^2,$$
$$\int_0^1 \|F'(c^*+tp)^*[F'(c^*+tp) - F'(c^*)]p\,dt\| \le K(K_0/2)\|p\|^2,$$

for a positive constant $K>0$. Then, from (6.7) it follows

$$\|f'(c^*+p) - f'(c^*) - F'(c^*)^*F'(c^*)(p)\| \le \tilde{\gamma}\|p\|^2 + \sigma\|p\|. \tag{6.9}$$

with $\tilde{\gamma}>0$.

In [21] it is proved that, both in case of attainable data and in case of small enough residual, it holds

$$f''(c^*)(h,h) \ge \beta\|h\|^2, \tag{6.10}$$

for an appropriately defined constant $\beta>0$ independent of $h \in H_N$. This comes from Lemma 2.6, Theorem 5.1, Lemma 5.1, Theorem 5.2 in [21]. Let $V = \{v \in C^\infty | R_i v = 0, i = 1,2\}$, and define $\tilde{H}^{-2}$ the completion of C^∞ with respect to the $\|\cdot\|_{\tilde{H}^{-2}}$ norm defined for $u \in C^\infty$ as [21, p.6]

$$\|u\|_{\tilde{H}^{-2}} = \sup_{v\in V} \frac{|<u,v>|}{\|v\|_{H^2}}.$$

Theorem 6.3. *Let H1 and H2 hold.*

- *Then, $\kappa_1\|u\| \le \|A(c)u\|_{\tilde{H}^{-2}} \le \kappa_2\|u\|$ holds for all $c \in D(F)$ and $u \in D(\mathscr{A})$, for κ_1, κ_2 defined in Lemma 6.1.*
- *Moreover, let $c^* \in \mathscr{U}_N$ be a local solution for (6.6) and suppose $u(c^*)(x) > 0$ or $u(c^*)(x) < 0$ on $[0,1]$. If $u(c^*) \notin \mathscr{V}_N$ let $\kappa > 0$ be such that $\|hu(c^*)\|_{L^\infty} \le \kappa\|hu(c^*)\|_{\tilde{H}^{-2}}$ and assume further that*

  $$\|F(c^*) - \tilde{u}\| \le \frac{1}{2}\kappa_1\kappa_2^{-1}\kappa^{-1}\min_x \|F(c^*)(x)\|.$$

  Then, (6.10) holds for $\beta>0$ independent of h.

From this result it follows that f is strongly convex in a neighbourhood of the solution, leading to

$$\beta\|c-c^*\|^2 \le < f'(c) - f'(c^*), c-c^* > .$$

Taking into account also that

$$< f'(c) - f'(c^*), c-c^* > \le \|f'(c) - f'(c^*)\| \|c-c^*\|$$

it follows $\beta\|c-c^*\| \le \|f'(c) - f'(c^*)\|$. This can be used in (6.9) to prove condition (5.6).

Part III

Large scale noisy nonlinear least squares problems

Chapter 7
Introduction to Part III

In this part we consider large scale nonlinear least squares problems of the form (1.2). Let x^* be a solution of (1.2).

We are interested in problems for which the exact values of the objective function and of its gradient cannot be employed along all the optimization process. This framework includes various situations. One is the case in which the exact objective function is unknown, and just noisy approximations are available. Another case is that where the objective function evaluation is the result of a computation whose accuracy can vary and must be specified in advance. For instance the evaluation of the objective may involve the solution of a nonlinear equation or an inversion process. These are performed through an iterative process that can be stopped when a certain accuracy level is reached. Varying the accuracy level clearly affects the computational cost of the procedure. More generally, we consider also all those problems for which an exact evaluation of the function is computationally demanding and can be replaced by cheaper approximations. In all these cases one may wonder if it could be possible to exploit this feature by asking the lowest possible accuracy in the value of the objective, but sufficient to guarantee progress of the minimization, with the ultimate goal of saving computing time [22, §10.6].

In the literature several methods dealing with problems with noisy functions and gradients have been proposed. In [22, Section 10.6] a trust-region approach has been proposed to deal with dynamic accuracy levels. Classical methods for noisy functions are Simplex Gradient and Implicit Filtering, Direct Search [70, §6-8], but also evolutionary techniques have been considered [4]. In [74] minimization problems with convex objective function, whose exact evaluation is not possible, are considered. They are reformulated in terms of constrained optimization and handled with an Inexact Restoration technique. The approach is extended in [12] to stochastic optimization. Recently an intensive study has focused on stochastic methods, that employ random models of the objective function, that may result from a sampling procedure [5, 13, 23]. This kind of techniques have been widely employed in the last years to solve data-fitting problems arising in machine learning and in data assimilation [16, 45]. This applications usually involve objective functions given by a sum over a large number of terms, whose evaluation is expensive or prohibitive if a huge amount of data is available. Also, there is often an approximate form of redundancy in the measurements, which means that a full evaluation of the function or the gradient may be unnecessary to make progress in solving (1.2) [16, 43]. This motivates the derivation of methods that approximate the function and/or the gradient and even the Hessian through subsampling techniques. Both first order and second order methods have been proposed, see [16] for a recent review on the topic and references therein.

Here, we are interested in problems for which it is convenient to rely on approximations f_δ to f to recover x^*. It is assumed that the accuracy level of the function approximations can be evaluated and improved when judged to be too low to proceed successfully with the optimization process. This is not the case of ill-posed problems presented in Part II, as the noise arises from measurements and it cannot be modified without repeating new measurements. Then, if further measurements are not possible, the noise is a fixed quantity. Moreover as opposed to Part II, here the noise is not assumed to be limited to the data, so that also the Jacobian matrix of R is assumed to be affected by noise.

We propose a Levenberg-Marquardt method able to take into account the presence of noise in the objective function, in its gradient and in the Jacobian matrix of R. It is aimed at finding a solution of problem (1.2) considering a sequence of approximations f_{δ_k} of known and increasing accuracy, and we assume that we have access to approximate function and gradient values at any accuracy level. What we consider at each iteration k is then a noisy problem of the form:

$$f_{\delta_k}(x) = \frac{1}{2}\|R_{\delta_k}(x)\|^2, \tag{7.1}$$

where R_{δ_k} is the approximation of R at iteration k. We denote by $J_{\delta_k}(x) \in \mathbb{R}^{m\times n}$ the approximation to the Jacobian matrix of $R(x)$ and with $\nabla f_{\delta_k}(x) = J_{\delta_k}(x)^T R_{\delta_k}(x) \in \mathbb{R}^n$ the gradient approximation. Notice that in this part we use a different notation from the previous ones. Specifically, the subscripts referring to the noise are iteration dependent, as the noise can vary from iteration to iteration.

Having in mind large scale problems, the linear algebra operations will be handled by an iterative solver and inexact solutions of the subproblems will be sought for, cf. Section 2.4.1.

Let us outline briefly our solution method. We start the optimization process with a given noise level $\delta = \delta_0$. We rely during the iterative process on a noise control that allows us to judge whether the noise is too large, and needs to be reduced. In this case the accuracy is changed, making possible the use of more accurate approximations of function, gradient and Jacobian in further iterations. Our method is also based on an update of the regularization parameters that is different from the standard ones presented in Section 2.4. This is used to prevent the sequence from being attracted by one of the solution of the noisy problems. Drawing upon the Trust-Region radius Δ_k updates that drive the radius to zero (as those presented in Part II and in the references in Section 2.4) and taking into account the relation between Δ_k and λ_k we focused on in Section 2.4, we generate a non-decreasing sequence of regularization parameters.

We assume that given an accuracy level $\delta_k \geq 0$ it is possible to compute an approximation f_{δ_k} to f such that:

$$\left|f_{\delta_k}(x) - f(x)\right| \leq \delta_k, \tag{7.2}$$

for $x \in \mathscr{L} = \{x \,|\, f(x) \leq f(x_0)\}$. We will refer to δ_k as the accuracy or noise level. We also assume that it is possible to compute an approximation to the gradient of the same level of accuracy. Namely we assume that it exists $\bar{K} \geq 0$ such that:

$$\|\nabla f_{\delta_k}(x) - \nabla f(x)\| \leq \bar{K}\delta_k. \tag{7.3}$$

It is reasonable to ask for an approximation to the gradient of the same accuracy as the function, as the quality of both approximations of f and ∇f depend on the distance $\max\{\|R_{\delta_k}(x) - R(x)\|, \|J_{\delta_k}(x) - J(x)\|\}$, as follows:

$$\begin{aligned}
\left|f_{\delta_k}(x) - f(x)\right| &\leq \frac{1}{2}\|R_{\delta_k}(x) - R(x)\| \sum_{j=1}^{m} |R_j(x) + (R_{\delta_k})_j(x))|, \\
\|\nabla f(x) - \nabla f_{\delta_k}(x)\| &\leq \|J_{\delta_k}(x) - J(x)\|\,\|R(x)\| + \|J_{\delta_k}(x)\|\,\|R_{\delta_k}(x) - R(x)\|.
\end{aligned}$$

Our intention is to rely on less accurate (and hopefully cheaper quantities) whenever possible in earlier stages of the algorithm, increasing only gradually the demanded accuracy, to obtain a reduced computational time for the overall solution process.

We prove both global and local convergence of the method. We are not aware of Levenberg-Marquardt methods specially designed for nonzero residual noisy nonlinear least squares problems, for which both local and global convergence is proved. Contributions on this topic are given by [11], where a Levenberg-Marquardt method is proposed that deals with exact functions and the noise is limited to the gradient of f and to the Jacobian matrix of R. There only global convergence is proved. In the problems considered in Part II the Jacobian matrix is not affected by noise and only local convergence of the methods is considered.

Importantly enough, the method and the related theory also apply to the situation where the output space of R_{δ_k} has smaller dimension than that of R, i.e. $R_{\delta_k} : \mathbb{R}^n \to \mathbb{R}^{K_k}$ with $K_k \leq m$ for some k. This is the case for example when approximations to f stem from a subsampling technique and R_{δ_k} is obtained by selecting some components of R. In this case it is possible to obtain a better approximation to f, and so to reduce the noise, by adding more observations to the considered subset, i.e. increasing K_k, until the maximum value m is reached. We denote accordingly by $J_{\delta_k}(x) \in \mathbb{R}^{K_k \times n}$ the Jacobian matrix of $R_{\delta_k}(x)$.

This part is organized as follows. We describe the proposed Levenberg-Marquardt approach in Chapter 8, focusing on the strategy to control the noise level. We analyze also the asymptotic behaviour of the sequence of regularization parameters generated. In Section 8.1 global convergence to first-order critical points is proved.

In Section 8.2 we analyze the asymptotic local behaviour of the procedure. In Section 8.3 we provide a global complexity bound for the proposed method showing that it shares its complexity properties with the steepest descent and trust-region methods. Then, in Chapter 9 we consider the numerical behaviour of the method. We numerically illustrate the approach on test problems arising in data assimilation (Section 9.1) and in machine learning (Section 9.2), one of which is a real life problem, arising in the design of turbomachinery components. We show that our procedure is able to handle the noise and find a solution of the unperturbed problem. Moreover we show that when the unperturbed function is available, but it is expensive to optimize, the use of our noise control strategy allows us to obtain large computational savings.

Notations As previously stated, we will denote with $f_{\delta_k}(x) = \frac{1}{2}\|R_{\delta_k}(x)\|^2$ the approximation of f at iteration k corresponding to noise level δ_k, $R_{\delta_k} : \mathbb{R}^n \to \mathbb{R}^{K_k}$, $K_k \leq m$ is the approximation of R at iteration k. We denote by $J_{\delta_k}(x) \in \mathbb{R}^{K_k \times n}$ the approximation to the Jacobian matrix of $R(x)$ and with $\nabla f_{\delta_k}(x) = J_{\delta_k}(x)^T R_{\delta_k}(x) \in \mathbb{R}^n$ the gradient approximation. Here, for ease of notation, all the iterates are denoted as $\{x_k\}$, even if noisy functions are considered. In this chapter indeed, just noisy problems are considered and there is not possibility of misunderstanding as in the previous ones.

CHAPTER

8 Levenberg-Marquardt method for problems with dynamic noise

We consider an inexact Levenberg-Marquardt approach for problem (1.2), cf. Section 2.4. At each iteration, given an appropriately chosen regularization parameter λ_k, we consider a subproblem of the form:

$$\min_{p\in\mathbb{R}^n} m_k^{LM}(x_k+p) = \frac{1}{2}\|R_{\delta_k}(x_k)+J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2, \tag{8.1}$$

and we seek for an approximate solution, according to Definition 2.14. Namely, our step has to provide at least as much reduction in m_k^{LM} as that achieved by the Cauchy step (2.22):

$$m_k^{LM}(x_k) - m_k^{LM}(x_k+p) \geq \frac{\theta}{2}\frac{\|J_{\delta_k}(x_k)^T R_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2+\lambda_k}, \qquad \theta > 0. \tag{8.2}$$

To achieve this, we solve approximately the normal equations, as in (2.24) and we make the following assumption on the step:

Assumption 8.1. *Assume to compute a step p_k^{LM} satisfying*

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k^{LM} = -J_{\delta_k}(x_k)^T R_{\delta_k}(x_k) + r_k \tag{8.3}$$

for a residual r_k satisfying

$$\|r_k\| \leq \epsilon_k \|J_{\delta_k}(x_k)^T R_{\delta_k}(x_k)\|, \qquad 0 \leq \epsilon_k \leq \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2+\lambda_k}}, \tag{8.4}$$

for some $\theta_2 \in \left(0, \frac{1}{2}\right]$.

Remark 8.2. *From Lemma 2.15 such a step achieves the Cauchy decrease (8.2) with $\theta = 2(1-\theta_2) \in [1,2)$.*

This key result will be used to prove the global convergence of the method we propose. To compute p_k^{LM} we can then use an iterative method to solve the normal equations

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p = -J_{\delta_k}(x_k)^T R_{\delta_k}(x_k), \tag{8.5}$$

that will be stopped as soon as the desired accuracy is reached, i.e. as soon as (8.4) is satisfied.

Let us now describe in details the way the noise level is controlled along the iterations. In classical globally convergent Levenberg-Marquardt methods, as described in Section 2.4, at each iteration the acceptance of the trial step and the update of the parameter are based on the reduction gained in the objective function. Here, we deal with objective functions affected by noise. Then, we need a condition to check if the noise level is small enough to ensure that the decrease in function values observed after a successful iteration, is not merely an effect of the inaccuracy in these values, but corresponds to a true decrease also in the exact objective function. In [22, Section 10.6], it is proved that this is achieved if the noise level δ_k is smaller than a multiple of the reduction in the model:

$$\delta_k \leq \eta_0[m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM})],$$

with $\eta_0 > 0$.

We will prove in Remark 8.7 and numerically illustrate in Section 9, that for our approach

$$m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM}) = O(\lambda_k \|p_k^{LM}\|^2). \tag{8.6}$$

According to this and following [22], we control the noise level asking that

$$\delta_k \leq \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2, \tag{8.7}$$

for constants $\kappa_d > 0$ and $\alpha \in \left[\frac{1}{2}, 1\right)$. Parameter α in (8.7) is introduced to guarantee global convergence of the procedure, as shown in Section 8.1. Notice also that (8.7) is an implicit relation, as p_k^{LM} depends on the noise. If condition (8.7) is not satisfied at iteration k, the uncertainty in the function values is considered too high and the noise level is decreased. We will prove in Lemma 8.4 that after a finite number of reductions condition (8.7) is met.

Our approach is sketched in Algorithm 8.1. At each iteration k a trial step p_k^{LM} is computed using the noise level of the previous successful iteration. The norm of the trial step is then used to check condition (8.7). In case it is not satisfied the noise is reduced in the loop at steps 1-2 until (8.7) is met. On the other hand, when the condition is satisfied the function approximation is not changed for next iteration and it is not necessary to estimate the noise again. The value δ_k obtained at the end of the loop is used to compute $f_{\delta_k}(x_k + p_k^{LM})$. Then, the ratio between the actual and the predicted reduction

$$\rho_k(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM})} \tag{8.8}$$

is computed to decide whether to accept the step or not. Practically, notice that if at iteration k the noise level is changed, i.e. $\delta_k \neq \delta_{k-1}$, the function is not evaluated again in x_k to compute $\rho_k(p_k^{LM})$, and the ratio is evaluated computing the difference between $f_{\delta_{k-1}}(x_k)$ (evaluated at the previous step), and the new computed value $f_{\delta_k}(x_k + p_k^{LM})$. The step acceptance and the updating of the regularization parameter are based on this ratio. As in standard Levenberg-Marquardt

Algorithm 8.1 Levenberg-Marquardt method for problem (1.2) using dynamic noise

```
Input: $x_0$, $\delta_0$, $\kappa_d \geq 0$, $\alpha \in \left[\frac{1}{2}, 1\right)$, $\beta > 1$, $\eta_1 \in (0,1)$, $\eta_2 > 0$, $\lambda_{\max} \geq \lambda_0 > 0$, $\gamma > 1$.
Compute $f_{\delta_0}(x_0)$ and set $\delta_{-1} = \delta_0$.
for $k = 0, 1, 2, \ldots$ do
    1. Compute an approximate solution of (8.1) satisfying (8.3) and let $p_k^{LM}$
    denote such a solution.
    2. If $\delta_k \leq \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2$
            Compute $f_{\delta_k}(x_k + p_k^{LM})$, and set $\delta_{k+1} = \delta_k$.
        Else
            Reduce $\delta_k$: $\delta_k = \frac{\delta_k}{\beta}$ and go back to 1.
    3. Compute $\rho_k(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM})}$.
    4. If $\rho_k(p_k^{LM}) \geq \eta_1$
            4.1 Set $x_{k+1} = x_k + p_k^{LM}$ and

            $$\lambda_{k+1} = \begin{cases} \min\{\gamma\lambda_k, \lambda_{\max}\} & \text{if } \|\nabla f_{\delta_k}(x_k)\| < \eta_2/\lambda_k, \\ \lambda_k & \text{if } \|\nabla f_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k. \end{cases}$$

        Else
            4.2 Set $x_{k+1} = x_k$, $\lambda_{k+1} = \gamma\lambda_k$ and $\delta_{k+1} = \delta_{k-1}$.
end for
```

methods, the step is successful if $\rho_k(p_k^{LM}) \geq \eta_1$, and is unsuccessful otherwise. Deviating from classical Levenberg-Marquardt and following [5, 11], λ_k is increased not only in case of unsuccessful iterations, but also if the inverse of the regularization parameter is big compared to the norm of the gradient model (condition $\|\nabla f_{\delta_k}(x_k)\| < \eta_2/\lambda_k$ in Algorithm 8.1), otherwise it is left unchanged. The logic behind this update is the same as in [5, 11] and it is intended to generate a sequence of increasing parameters, that would not be generated by a standard update like (2.20), to produce a regularizing effect and prevent the generated sequence to converge to solutions of the noisy problems. This allows also to control the noise level through (8.7) and to decrease it especially in the last stage of the procedure. It represents a counterpart of the update of the trust-region radius in Part II.

Notice that in case the step is unsuccessful λ_k is increased and noise reductions performed at steps 1-2 are not taken into account. That is, the subsequent iteration $k+1$ is started with the same starting noise level of iteration k (see step 4.2).

First we prove the well-definition of Algorithm 8.1. Specifically, in Lemma 8.4 we prove that the loop at steps 1-2 of Algorithm 8.1 terminates in a finite number of steps. To this aim we need the following assumption:

Assumption 8.3. *Let $\{x_k\}$ be the sequence generated by Algorithm 8.1. It exists a positive constant κ_J such that, for all $k \geq 0$ and all $x \in [x_k, x_k + p_k^{LM}]$, $\|J_{\delta_k}(x)\| \leq \kappa_J$.*

Lemma 8.4. *Let Assumption 8.3 hold and let p_k^{LM} be defined as in Assumption 8.1. If x_k is not a stationary point of f, the loop at steps 1-2 of Algorithm 8.1 terminates in a finite number of steps.*

Proof. If δ_k tends to zero, $\nabla f_{\delta_k}(x_k)$ tends to $\nabla f(x_k)$ from (7.3). Also, (8.4) yields $\epsilon_k \leq \sqrt{\theta_2}$, and from (8.3) it follows

$$\begin{aligned} \|p_k^{LM}\| = & \|(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)^{-1}(-\nabla f_{\delta_k}(x_k) + r_k)\| \geq \\ \geq & \frac{(1-\epsilon_k)\|\nabla f_{\delta_k}(x_k)\|}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k} \geq \frac{(1-\sqrt{\theta_2})\|\nabla f_{\delta_k}(x_k)\|}{\kappa_J^2 + \lambda_k}. \end{aligned} \tag{8.9}$$

Then,

$$\liminf_{\delta_k \to 0} \|p_k^{LM}\| \geq \frac{(1-\sqrt{\theta_2})}{\kappa_J^2 + \lambda_k}\|\nabla f(x_k)\| > 0$$

as $\nabla f(x_k) \neq 0$, so for δ_k small enough (8.7) is satisfied. □

As far as the sequence of regularization parameters is concerned, we notice that it is bounded from below, as $\lambda_{\min} = \lambda_0 \leq \lambda_k$ for all k. Moreover an upper bound $\lambda_{\max}$ is provided for successful iterations in step 4.1, so that the procedure gives rise to a sequence of regularization parameters with different behaviour than the one generated in [11], where the upper bound is not present and λ_k is free to go to infinity. It is possible to prove that the bound is reached and for k large enough $\lambda_k = \lambda_{\max}$ on the subsequence of successful iterations, while in [11] the sequence is shown to diverge. The result is proved in the following lemma.

Lemma 8.5. *Let Assumption 8.3 hold and let p_k^{LM} be defined as in Assumption 8.1. It exists $\bar{k} \geq 0$ such that the regularization parameters $\{\lambda_k\}$ generated by Algorithm 8.1 satisfy $\lambda_k = \lambda_{\max}$ for any successful iteration k, with $k \geq \bar{k}$.*

Proof. If the result is not true, there exists a bound $0 < B < \lambda_{\max}$ such that the number of times that $\lambda_k < B$ happens is infinite. Because of the way λ_k is updated one must have an infinity of iterations for which $\lambda_{k+1} = \lambda_k$, and for them one has $\rho_k(p_k^{LM}) \geq \eta_1$ and $\|\nabla f_{\delta_k}(x_k)\| \geq \eta_2/B$. Thus, from Lemma 2.15 and relation (8.2)

$$\begin{aligned} f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM}) & \geq \eta_1(m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM})) \\ & \geq \frac{\eta_1}{2}\frac{\theta\|\nabla f_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k} \\ & \geq \frac{\eta_1}{2}\frac{\theta}{\kappa_J^2 + B}\left(\frac{\eta_2}{B}\right)^2. \end{aligned}$$

Since f_{δ_k} is bounded below by zero and the sequence $\{f_{\delta_k}(x_{k+1})\}$ is decreasing and hence convergent, the number of such iterations cannot be infinite, hence we derive a contradiction. Then, for an infinite number of iterations λ_{k+1} is set as in steps 4.1 or 4.2, so that $\lambda_{k+1} > \lambda_k$. For the subsequence of successful iterations it exists $\bar{k}$ large enough for which $\lambda_k = \lambda_{\max}$ for all $k \geq \bar{k}$. □

Remark 8.6. *The regularization parameters form a non decreasing sequence. As we have noticed that the reciprocal of λ_k plays the role of the radius in a trust-region scheme, a Levenberg-Marquardt method with non decreasing sequence of parameters is equivalent to a trust-region scheme with trust-region radius converging to zero, as the ones we have employed in the previous chapter or as the ones presented for example in [5, 23] and references in Section 2.4. As we have previously noticed indeed, having a radius converging to zero, ensures regularizing properties to the method.*

Remark 8.7. *Lemma 8.5 enables us to prove (8.6) and to motivate condition (8.7). From the model definition and (8.3) it holds*

$$\begin{aligned} m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM}) &= -\frac{1}{2}(p_k^{LM})^T (J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k^{LM} - (p_k^{LM})^T \nabla f_{\delta_k}(x_k) \\ &= \frac{1}{2}\|J_{\delta_k}(x_k) p_k^{LM}\|^2 + \frac{1}{2}\lambda_k \|p_k^{LM}\|^2 - (p_k^{LM})^T r_k. \end{aligned}$$

Considering that from (8.4) and (8.9)

$$(p_k^{LM})^T r_k \le \epsilon_k \|p_k^{LM}\| \, \|\nabla f_{\delta_k}(x_k)\| \le \frac{\sqrt{\theta_2}}{1-\sqrt{\theta_2}}(\kappa_J^2 + \lambda_k)\|p_k^{LM}\|^2,$$

and that parameters λ_k form a non decreasing sequence, we can conclude that (8.6) holds.

In the following section, we will prove that the sequence generated by Algorithm 8.1 converges globally to a solution of (1.2).

8.1 Global convergence

In this section we prove the global convergence of the sequence generated by Algorithm 8.1. We still assume to compute an inexact Levenberg-Marquardt step according to Definition 2.14, but we need to make a slightly stronger assumption on the residual of the normal equations (8.5) than in Assumption 8.1, to prove the global convergence.

Assumption 8.8. *Let p_k^{LM} satisfy*

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k^{LM} = -\nabla f_{\delta_k}(x_k) + r_k$$

for a residual r_k satisfying $\|r_k\| \le \epsilon_k \|\nabla f_{\delta_k}\|$, with

$$0 \le \epsilon_k \le \min\left\{ \frac{\theta_1}{\lambda_k^\alpha}, \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}} \right\} \tag{8.10}$$

where $\theta_1 > 0$, $\theta_2 \in \left(0, \frac{1}{2}\right]$ and $\alpha \in \left[\frac{1}{2}, 1\right)$ is defined in (8.7).

As stated in Lemma 2.15, this step achieves the Cauchy decrease. The new bound in (8.10) will be used in the convergence analysis.

We now report a result relating the step length and the norm of the noisy gradient at each iteration, that is going to be useful in the following analysis.

Lemma 8.9. *Let Assumptions 8.3 and 8.8 hold. Then*

$$\|p_k^{LM}\| \leq \frac{2\|\nabla f_{\delta_k}(x_k)\|}{\lambda_k}. \tag{8.11}$$

Proof. Taking into account that from Assumption 8.8 $\|r_k\| \leq \epsilon_k \|\nabla f_{\delta_k}\| \leq \|\nabla f_{\delta_k}\|$, it follows

$$\|p_k^{LM}\| = \|(J_{\delta_k}^T J_{\delta_k} + \lambda_k I)^{-1}(-\nabla f_{\delta_k}(x_k) + r_k)\| \leq \frac{2\|\nabla f_{\delta_k}(x_k)\|}{\lambda_k}.$$

□

In the following lemma we establish a relationship between the exact and the noisy gradient which holds for λ_k large enough. Notice that by (8.7) the noise level depends on λ_k. Specifically, employing (8.7), Lemma 8.9 and the following Lemma 8.10, we can see that δ_k decreases as λ_k gets larger. Then, relation (8.13) shows that our noise level control strategy, combined with the updating rule of λ_k, imposes to gradually reduce the noise level in order to obtain a small relative error on the gradient's norm, whenever λ_k is sufficiently large.

Lemma 8.10. *Let Assumptions 8.3 and 8.8 hold. For λ_k sufficiently large, i.e. for*

$$\lambda_k \geq \nu\lambda^* := \nu\left(2\sqrt{\delta_0\kappa_d}\bar{K}\right)^{\frac{2}{2-\alpha}} \qquad \nu > 1, \tag{8.12}$$

it exists $c_k \in (0,1)$ such that the following relation between the exact and the perturbed gradient holds:

$$\frac{\|\nabla f(x_k)\|}{(1+c_k)} \leq \|\nabla f_{\delta_k}(x_k)\| \leq \frac{\|\nabla f(x_k)\|}{(1-c_k)}, \text{ with } c_k = \frac{2\bar{K}\sqrt{\delta_0\kappa_d}}{\lambda_k^{1-\alpha/2}} = \left(\frac{\lambda^*}{\lambda_k}\right)^{1-\alpha/2}. \tag{8.13}$$

Proof. From (7.3), (8.7), (8.11) and the fact that $\delta_k \leq \delta_0$ it follows

$$|\|\nabla f(x_k)\| - \|\nabla f_{\delta_k}(x_k)\|| \leq \|\nabla f(x_k) - \nabla f_{\delta_k}(x_k)\| \leq \bar{K}\sqrt{\delta_0}\sqrt{\delta_k} \leq \bar{K}\sqrt{\delta_0\kappa_d\lambda_k^{\alpha}\|p_k^{LM}\|^2} =$$

$$\bar{K}\sqrt{\delta_0\kappa_d}\lambda_k^{\alpha/2}\|p_k^{LM}\| \leq 2\bar{K}\sqrt{\delta_0\kappa_d}\frac{\|\nabla f_{\delta_k}(x_k)\|}{\lambda_k^{1-\alpha/2}} = c_k\|\nabla f_{\delta_k}(x_k)\|$$

where we have set $c_k = \frac{2\bar{K}\sqrt{\delta_0\kappa_d}}{\lambda_k^{1-\alpha/2}}$. Then,

$$\|\nabla f(x_k) - \nabla f_{\delta}(x_k)\| \leq c_k\|\nabla f_{\delta_k}(x_k)\|, \tag{8.14}$$

$$(1-c_k)\|\nabla f_{\delta_k}(x_k)\| \leq \|\nabla f(x_k)\| \leq (1+c_k)\|\nabla f_{\delta_k}(x_k)\|, \tag{8.15}$$

and for $\lambda_k > \lambda^*$, the thesis follows.

□

From the updating rule of the noise in step 2 of Algorithm 8.1, if δ_{k-1} is the successful noise level at iteration $k-1$, the successful noise level at iteration k is

$$\delta_k = \frac{\delta_{k-1}}{\beta^{n_k}} \tag{8.16}$$

where $n_k \geq 0$ counts the number of times the noise is reduced in the loop at steps 1-2, that is finite from Lemma 8.4. We can also prove that the sequence $\{\beta^{n_k}\}$ is bounded from above. To this aim, we need the following assumption, which is standard in Levenberg-Marquardt methods:

Assumption 8.11. *Assume that function f has Lipschitz continuous gradient with corresponding constant $L > 0$: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.*

Lemma 8.12. *Let Assumptions 8.3, 8.8, 8.11, hold and λ^* be defined in (8.12). Then, if $\lambda_k \geq \nu\lambda^*$ for $\nu > 1$, there exists a constant $\bar{\beta} > 0$ such that $\beta^{n_k} \leq \bar{\beta}$.*

Proof. Let δ_{k-1} be the successful noise level at iteration $k-1$. Then, it holds

$$\delta_{k-1} \leq \kappa_d \lambda_{k-1}^{\alpha} \|p_{k-1}^{LM}\|^2. \tag{8.17}$$

If in (8.16) $n_k \leq 1$ there is nothing to prove, so let us assume $n_k > 1$. If $n_k > 1$ it holds

$$\beta\delta_k > \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2,$$

for δ_k the noise level at the end of the loop 1-2 of Algorithm 8.1. From the updating rule at step 4 of Algorithm 8.1 it follows

$$\lambda_{k-1} \leq \lambda_k \leq \gamma\lambda_{k-1}. \tag{8.18}$$

Using the first inequality in (8.18) and (8.17) we obtain from (8.16) that

$$\beta^{n_k - 1} = \frac{\delta_{k-1}}{\beta\delta_k} < \frac{\kappa_d \lambda_{k-1}^{\alpha} \|p_{k-1}^{LM}\|^2}{\kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2} \leq \frac{\|p_{k-1}^{LM}\|^2}{\|p_k^{LM}\|^2}.$$

Then, from Assumption 8.8 we have

$$\begin{aligned}\beta^{n_k - 1} &\leq \frac{\|(J_{\delta_{k-1}}(x_{k-1})^T J_{\delta_{k-1}}(x_{k-1}) + \lambda_{k-1} I)^{-1}(-\nabla f_{\delta_{k-1}}(x_{k-1}) + r_{k-1})\|^2}{\|(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)^{-1}(-\nabla f_{\delta_k}(x_k) + r_k)\|^2} \leq \\ &\leq \frac{\|J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I\|^2}{\|-\nabla f_{\delta_k}(x_k) + r_k\|^2} \|(J_{\delta_{k-1}}(x_{k-1})^T J_{\delta_{k-1}}(x_{k-1}) + \lambda_{k-1} I)^{-1}\|^2 \|-\nabla f_{\delta_{k-1}}(x_{k-1}) + r_{k-1}\|^2.\end{aligned}$$

Recalling again Assumption 8.8, the fact that from (8.10) $\epsilon_k < \sqrt{\theta_2} < 1$, and (8.13) we have

$$\begin{aligned}\|-\nabla f_{\delta_{k-1}}(x_{k-1}) + r_{k-1}\|^2 &< 4\|\nabla f_{\delta_{k-1}}(x_{k-1})\|^2 \leq \frac{4}{(1 - c_{k-1})^2}\|\nabla f(x_{k-1})\|^2, \\ \|-\nabla f_{\delta_k}(x_k) + r_k\|^2 &> (1 - \sqrt{\theta_2})^2 \|\nabla f_{\delta_k}(x_k)\|^2 \geq \frac{(1 - \sqrt{\theta_2})^2}{(1 + c_k)^2}\|\nabla f(x_k)\|^2.\end{aligned}$$

Then, recalling also Assumption 8.3 we obtain

$$\beta^{n_k - 1} \leq \frac{4}{(1 - \sqrt{\theta_2})^2} \left(\frac{\kappa_J^2 + \lambda_k}{\lambda_{k-1}}\right)^2 \frac{(1 + c_k)^2}{(1 - c_{k-1})^2} \frac{\|\nabla f(x_{k-1})\|^2}{\|\nabla f(x_k)\|^2}.$$

By (8.18) and the fact that $\lambda_k \geq \lambda_{\min} = \lambda_0$ for all k, it follows

$$\left(\frac{\kappa_J^2 + \lambda_k}{\lambda_{k-1}}\right)^2 \leq \left(\frac{\kappa_J^2}{\lambda_{\min}} + \gamma\right)^2.$$

From (8.13)

$$c_k = \left(\frac{\lambda^*}{\lambda_k}\right)^{1 - \alpha/2} \leq \nu^{\alpha/2 - 1} < 1. \tag{8.19}$$

Then, from this and the first inequality in (8.18) it follows:

$$\begin{aligned} &1+c_k<2,\\ &1-c_{k-1}=1-\left(\frac{\lambda^*}{\lambda_{k-1}}\right)^{1-\alpha/2}\geq 1-\gamma^{1-\alpha/2}. \end{aligned}$$

Then,

$$\beta^{n_k-1}\leq \frac{16}{(1-\sqrt{\theta_2})^2}\left(\frac{\kappa_J^2}{\lambda_{\min}}+\gamma\right)^2\left(\frac{1}{1-\gamma^{1-\alpha/2}}\right)^2\left(\frac{\|\nabla f(x_{k-1})\|}{\|\nabla f(x_k)\|}\right)^2.$$

Let us now consider the term $\frac{\|\nabla f(x_{k-1})\|}{\|\nabla f(x_k)\|}$. By the Lipschitz continuity of the gradient, (8.11) and (8.19) we obtain:

$$\begin{aligned} \frac{\|\nabla f(x_{k-1})\|}{\|\nabla f(x_k)\|} &\leq 1+\frac{\|\nabla f(x_{k-1})-\nabla f(x_k)\|}{\|\nabla f(x_k)\|}\leq 1+\frac{L\|p_k^{LM}\|}{\|\nabla f(x_k)\|}\\ &\leq 1+\frac{2L\|\nabla f_{\delta_k}(x_k)\|}{\lambda_k\|\nabla f(x_k)\|}\leq 1+\frac{2L}{(1-c_k)\lambda_k}\\ &\leq 1+\frac{2L}{(1-\nu^{\frac{\alpha}{2}-1})\lambda_{\min}}. \end{aligned}$$

We can then conclude that sequence β^{n_k} is bounded from above by a constant for λ_k sufficiently large:

$$\beta^{n_k}\leq \beta\frac{16}{(1-\sqrt{\theta_2})^2}\left(\frac{\kappa_J^2}{\lambda_{\min}}+\gamma\right)^2\left(\frac{1}{1-\gamma^{1-\alpha/2}}\right)^2\left(1+\frac{2L}{(1-\nu^{\frac{\alpha}{2}-1})\lambda_{\min}}\right)$$

□

The result in Lemma 8.12 can be employed in the following Lemma, to prove that for sufficiently large values of the parameter λ_k the iterations are successful.

Lemma 8.13. *Let Assumptions 8.3, 8.8 and 8.11 hold. Assume that*

$$\lambda_k>\max\{\nu\lambda^*,\bar\lambda\} \tag{8.20}$$

with λ^* defined in (8.12) and

$$\bar\lambda=\left(\frac{\varphi}{1-\eta_1}\right)^{\frac{1}{1-\alpha}}\qquad \varphi=\left(\frac{\kappa_J^2/\lambda_{\min}+1}{\theta}\right)\left(\frac{2\theta_1}{\lambda_{\min}^{2\alpha-1}}+\frac{2L}{\lambda_{\min}^{\alpha}}+4(3+\bar\beta)\kappa_d+\frac{8\kappa_d\bar g}{\lambda_{\min}}\right), \tag{8.21}$$

with η_1, $\bar\beta$, θ_1, θ, α, L defined respectively in Algorithm 8.1, Lemma 8.12, Assumption 8.8, (8.2), (8.7) and Assumption 8.11, and $\bar g=\kappa_J\sqrt{2f_{\delta_0}(x_0)}$. If x_k is not a critical point of f then $\rho_k(p_k^{LM})\geq\eta_1$.

Proof. We consider

$$1-\frac{\rho_k(p_k^{LM})}{2}=\frac{-(p_k^{LM})^T(J_{\delta_k}(x_k)^TJ_{\delta_k}(x_k)+\lambda_kI)p_k^{LM}-2(p_k^{LM})^T\nabla f_{\delta_k}(x_k)}{2(m_k(x_k)-m_k(x_k+p_k^{LM}))} \tag{8.22}$$

$$+\frac{\frac12\|R_{\delta_k}(x_k+p_k^{LM})\|^2-\frac12\|R_{\delta_{k-1}}(x_k)\|^2}{2(m_k(x_k)-m_k(x_k+p_k^{LM}))}. \tag{8.23}$$

Let us consider the numerator in (8.23). Let us rewrite

$$f_{\delta_k}(x_k + p_k^{LM}) - f_{\delta_{k-1}}(x_k) = f_{\delta_k}(x_k + p_k^{LM}) - f_{\delta_{k-1}}(x_k) \pm f(x_k + p_k^{LM}) \pm f_{\delta_k}(x_k) \pm (p_k^{LM})^T \nabla f_{\delta_k}(x_k).$$

From the Taylor expansion of f and denoting with $\bar{\pi}$ the remainder, we obtain

$$f(x_k + p_k^{LM}) = f(x_k) + (p_k^{LM})^T \nabla f(x_k) + \bar{\pi}.$$

Then,

$$f_{\delta_k}(x_k + p_k^{LM}) - f_{\delta_{k-1}}(x_k) = f_{\delta_k}(x_k) - f_{\delta_{k-1}}(x_k) + (p_k^{LM})^T \nabla f_{\delta_k}(x_k) + \pi,$$

where

$$\pi = (f_{\delta_k}(x_k + p_k^{LM}) - f(x_k + p_k^{LM})) + (f(x_k) - f_{\delta_k}(x_k)) + (p_k^{LM})^T (\nabla f(x_k) - \nabla f_{\delta_k}(x_k)) + \bar{\pi}.$$

From condition (7.2) it follows

$$|f_{\delta_k}(x_k) - f_{\delta_{k-1}}(x_k)| \le \delta_k + \delta_{k-1}.$$

From (8.7) and the fact that from Lemma 8.12 $\delta_{k-1} = \beta^{n_k} \delta_k \le \bar{\beta} \delta_k$ if $\lambda_k > \lambda^*$, it follows

$$f_{\delta_k}(x_k + p_k^{LM}) - f_{\delta_{k-1}}(x_k) \le (1 + \bar{\beta}) \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2 + (p_k^{LM})^T \nabla f_{\delta_k}(x_k) + \pi.$$

Moreover, by (7.2), (7.3) and (8.7) we can conclude that

$$|\pi| \le \left((2 + \|p_k^{LM}\|) \kappa_d \lambda_k^{\alpha} + \frac{L}{2} \right) \|p_k^{LM}\|^2.$$

Then, from Lemma 8.9, Assumption 8.8 it follows that the numerator in (8.22)-(8.23) can be bounded above by

$$\begin{aligned}
&-(p_k^{LM})^T(-\nabla f_{\delta_k}(x_k) + r_k) - (p_k^{LM})^T \nabla f_{\delta_k}(x_k) + \pi + (1 + \bar{\beta}) \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2 \le \\
&\le \|p_k^{LM}\| \|r_k\| + \left(\kappa_d \lambda_k^{\alpha} \left(2 + \|p_k^{LM}\| \right) + \frac{L}{2} \right) \|p_k^{LM}\|^2 + (1 + \bar{\beta}) \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2 \le \\
&\le \left(\frac{2\theta_1}{\lambda_k^{1+\alpha}} + \frac{2L}{\lambda_k^2} + \frac{4(3 + \bar{\beta})\kappa_d}{\lambda_k^{2-\alpha}} + \frac{8\kappa_d \bar{g}}{\lambda_k^{3-\alpha}} \right) \|\nabla f_{\delta_k}(x_k)\|^2,
\end{aligned}$$

with $\bar{g} = \kappa_J \sqrt{2 f_{\delta_0}(x_0)}$. From (8.2) it follows

$$1 - \frac{\rho_k(p_k^{LM})}{2} \le \left(\frac{\kappa_J^2 / \lambda_{\min} + 1}{\theta} \right) \left(\frac{2\theta_1}{\lambda_k^{\alpha}} + \frac{2L}{\lambda_k} + \frac{4(3 + \bar{\beta})\kappa_d}{\lambda_k^{1-\alpha}} + \frac{8\kappa_d \bar{g}}{\lambda_k^{2-\alpha}} \right) \le \frac{\varphi}{\lambda_k^{1-\alpha}},$$

with φ defined in (8.21) and from (8.20) $\rho_k(p_k^{LM}) \ge 2\eta_1 > \eta_1$. □

We can now state the following result, which guarantees that eventually the iterations are successful, provided that

$$\lambda_{\max} > \max\{\nu \lambda^*, \bar{\lambda}\}. \tag{8.24}$$

Lemma 8.14. *Let Assumptions 8.3, 8.8 and 8.11 hold. Assume that $\lambda_{\max}$ is chosen to satisfy (8.24). Then, there exists an iteration index $\bar{k}$ such that the iterations generated by Algorithm 8.1 are successful for $k > \bar{k}$. Moreover,*

$$\lambda_k \le \max\left\{\gamma \max\{\nu\lambda^*, \bar{\lambda}\}, \lambda_{\max}\right\} \quad k > 0. \tag{8.25}$$

Proof. Notice that by the updating rules at step 3 of Algorithm 8.1, λ_k increases in case of unsuccessful iterations and it is never decreased. Therefore, after a finite number of unsuccessful iterations it reaches the value $\max\{\nu\lambda^*, \bar{\lambda}\}$. Moreover, condition (8.24) and the Algorithm's updating rules guarantee that $\lambda_k > \max\{\nu\lambda^*, \bar{\lambda}\}$ for all the subsequent iterations. Then, by Lemma 8.13 it follows that eventually the iterations are successful. Finally, the parameter updating rules yield (8.25). □

We are now ready to state and prove the global convergence of Algorithm 8.1 under the following further assumption:

Assumption 8.15. *Assume that $\lambda_{\max}$ is chosen large enough to satisfy*

$$\lambda_{\max} > \gamma \max\{\nu\lambda^*, \bar{\lambda}\}. \tag{8.26}$$

Notice that under this assumption $\lambda_k \le \lambda_{\max}$ for all $k > 0$. Parameters λ^*, $\bar{\lambda}$ depend on known algorithm's parameters, on the gradient Lipschitz constant L, on the bound κ_J and on $\bar{K}$ in (7.3). Then, to compute a value of $\lambda_{\max}$ satisfying (8.26) we need to estimate these three latter quantities. However, in numerical practice the choice of this value is not crucial. If a rather large value is set for this quantity the stopping criterion is usually satisfied before that value is reached.

Theorem 8.16. *Let Assumptions 8.3, 8.8, 8.11 and 8.15 hold. The sequences $\{\delta_k\}$ and $\{x_k\}$ generated by Algorithm 8.1 are such that*

$$\lim_{k\to\infty} \delta_k = 0, \qquad\qquad \lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Proof. From the updating rule of the noise, $\{\delta_k\}$ is a decreasing sequence and so it is converging to some value δ^*. Denoting with k^s the first successful iteration and summing up over all the infinite successful iterations, from Lemma 2.15 and Assumption 8.15 we obtain

$$\begin{aligned} f_{\delta_{k^s-1}}(x_{k^s}) - \lim_{k\to\infty} f_{\delta_k}(x_{k+1}) \ge \sum_{k_{succ}} (f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_{k+1})) \ge \\ \frac{\eta_1}{2} \frac{\theta}{\kappa_J^2 + \lambda_{\max}} \sum_{k_{succ}} \|\nabla f_{\delta_k}(x_k)\|^2, \end{aligned}$$

so $\sum\limits_{k_{succ}} \|\nabla f_{\delta_k}(x_k)\|^2$ is a finite number and $\|\nabla f_{\delta_k}(x_k)\| \to 0$ on the subsequence of successful iterations, so that $\lim_{k\to\infty} \|\nabla f_{\delta_k}(x_k)\| = 0$, taking into account that by Lemma 8.14 the number of unsuccessful iterations is finite. Finally from (8.7) and (8.11) we have that

$$\delta_k \le \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2 \le 4\kappa_d \frac{\|\nabla f_{\delta_k}(x_k)\|^2}{\lambda_{\min}^{2-\alpha}},$$

so we can conclude that the noise level δ_k converges to zero and by (7.3) it follows that $\lim_{k\to\infty}\|\nabla f(x_k)\| = 0$. □

8.2 Local convergence

In this section we report on the local convergence of the proposed method. To this aim, it is useful to study the asymptotic behaviour of the inexact step. We first establish that, as a consequence of employing a non decreasing sequence of parameters, the step p_k^{LM} asymptotically tends to assume the direction of the negative perturbed gradient $-\nabla f_{\delta_k}(x_k)$. Then, we study the local convergence of the gradient method with a perturbed gradient step, where the accuracy in the gradient is driven by (8.13).

Lemma 8.17. *Let Assumptions 8.3, 8.8, 8.11 and 8.15 hold. Then*

$$\lim_{k\to\infty}(p_k^{LM})_i + \frac{\theta}{\kappa_J^2+\lambda_k}(\nabla f_{\delta_k}(x_k))_i = 0 \quad \textit{for} \quad i = 1,\dots,n,$$

where $(\cdot)_i$ *denotes the* i*-th vector component.*

Proof. From (8.2)

$$\begin{aligned}
\frac{\theta}{2}\frac{\|\nabla f_{\delta_k}(x_k)\|^2}{\kappa_J^2+\lambda_k} &\leq m_k^{LM}(x_k) - m_k^{LM}(x_k+p_k^{LM})\\
&= -(p_k^{LM})^T\nabla f_{\delta_k}(x_k) - \frac{1}{2}(p_k^{LM})^T(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k)+\lambda_k I)p_k^{LM}\\
&\leq -(p_k^{LM})^T\nabla f_{\delta_k}(x_k) - \frac{1}{2}\lambda_k\|p_k^{LM}\|^2.
\end{aligned}$$

Therefore, as from Remark 8.2 it holds $\theta\in[1,2)$, it follows

$$\frac{\theta\|\nabla f_{\delta_k}(x_k)\|^2}{\kappa_J^2+\lambda_k} + 2(p_k^{LM})^T\nabla f_{\delta_k}(x_k) \ + \frac{\lambda_k}{\theta}\|p_k^{LM}\|^2 < 0.$$

We can rewrite this as

$$\left\|\sqrt{\frac{\theta}{\kappa_J^2+\lambda_k}}\nabla f_{\delta_k}(x_k) + \sqrt{\frac{\kappa_J^2+\lambda_k}{\theta}}p_k^{LM}\right\|^2 \leq \frac{\kappa_J^2}{\theta}\|p_k^{LM}\|^2.$$

Then, from Lemma 8.9

$$\left\|\frac{\theta}{\kappa_J^2+\lambda_k}\nabla f_{\delta_k}(x_k) + p_k^{LM}\right\|^2 \leq \frac{\kappa_J^2}{\kappa_J^2+\lambda_k}\|p_k^{LM}\|^2 \leq \frac{4\kappa_J^2\|\nabla f_{\delta_k}(x_k)\|^2}{\kappa_J^2\lambda_{\min}^2}$$

and the thesis follows as the right-hand side goes to zero when k tends to infinity from Theorem 8.16. □

From Lemma 8.17, if λ_k is large enough p_k^{LM} tends to assume the direction of $\nabla f_{\delta_k}(x_k)$ with step-length $\frac{\theta}{\kappa_J^2+\lambda_k}$. Then, eventually the method reduces to a perturbed steepest descent method with step-length and accuracy in the gradient inherited by the updating parameter and noise control strategies employed. As we will see in the numerical results section (cf. p.148), overall the procedure benefits from the use of a genuine Levenberg-Marquardt method till the last stage of convergence, gaining an overall faster convergence rate compared to a pure steepest descent method. Moreover this can be gained at a modest cost, as we solve the normal system (8.5) only to a low accuracy by an iterative solver. Then, the number of inner iterations is small, specially when the regularization term is large as $B_k + \lambda_k I \simeq \lambda_k I$. As a result in the last stage of the procedure the cost per iteration is comparable to that of a first order method.

In the following theorem we prove local convergence for the steepest descent step resulting from our procedure. The analysis is inspired by the one reported in [82, §1.2.3] which is extended to allow inaccuracy in gradient values.

Theorem 8.18. *Let x^* be a solution of problem (1.2). Let Assumptions 8.3 and 8.11 hold and let $\{x_k\}$ be a sequence such that*

$$x_{k+1} = x_k + p_k^{SD}, \quad k = 0,1,2,\dots$$

with

$$p_k^{SD} = -h(\lambda_k)\nabla f_{\delta_k}(x_k), \tag{8.27}$$

the perturbed steepest descent step with step-length $h(\lambda_k) = \frac{\theta}{\kappa_J^2+\lambda_k}$. Assume that there exists $r > 0$ such that f is twice differentiable in $\mathscr{B}_r(x^*)$ and let H be its Hessian matrix. Assume that $\|H(x) - H(y)\| \le M\|x - y\|$ for all $x, y \in \mathscr{B}_r(x^*)$ and let $0 < l \le \tilde{L} < \infty$ be such that $lI \preceq H(x^*) \preceq \tilde{L}I$. Assume that there exists an index $\bar{k}$ for which $\|x_{\bar{k}} - x^*\| < \bar{r}$ and

$$\lambda_k > \max\left\{\frac{\theta(\tilde{L}+l)}{2}, \lambda^*\left(1+\frac{2L}{l}\right)^{2/(2-\alpha)}\right\}, \tag{8.28}$$

where λ^* is defined in (8.12) and $\bar{r} = \min\{r, \frac{l}{M}\}$. Then for all $k \ge \bar{k}$ the error is decreasing, i.e. $\|x_{k+1} - x^*\| < \|x_k - x^*\|$, and $\|x_k - x^*\|$ tends to zero.

Proof. We follow the lines of the proof of Theorem 1.2.4 in [82] for an exact gradient step, taking into account that our step is computed using a noisy gradient. As $\nabla f(x^*) = 0$,

$$\nabla f(x_k) = \nabla f(x_k) - \nabla f(x^*) = \int_0^1 H(x^* + \tau(x_k - x^*))(x_k - x^*)\,d\tau := G_k(x_k - x^*),$$

where we have defined $G_k = \int_0^1 H(x^* + \tau(x_k - x^*))\,d\tau$. From (8.27),

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - h(\lambda_k)\nabla f(x_k) + h(\lambda_k)(\nabla f(x_k) - \nabla f_{\delta_k}(x_k)) = \\ &= (I - h(\lambda_k)G_k)(x_k - x^*) + h(\lambda_k)(\nabla f(x_k) - \nabla f_{\delta_k}(x_k)). \end{aligned}$$

From (8.14)

$$\|\nabla f_{\delta_k}(x_k) - \nabla f(x_k)\| \le c_k \|\nabla f_{\delta_k}(x_k)\| \le c_k \|\nabla f_{\delta_k}(x_k) - \nabla f(x_k)\| + c_k \|\nabla f(x_k)\|. \quad (8.29)$$

Notice that $c_k = \left(\frac{\lambda^*}{\lambda_k}\right)^{1-\frac{\alpha}{2}}$ (see (8.13)). If we let $k \ge \bar{k}$, (8.28) ensures $\lambda_k > \lambda^*$, and $c_k < 1$. Then, from (8.29) and the Lipschitz continuity of ∇f we obtain

$$(1 - c_k)\|\nabla f_{\delta_k}(x_k) - \nabla f(x_k)\| \le c_k \|\nabla f(x_k) - \nabla f(x^*)\| \le L c_k \|x_k - x^*\|.$$

Then, as (8.28) also yields $\lambda_k^{1-\frac{\alpha}{2}} - (\lambda^*)^{1-\frac{\alpha}{2}} \ge \frac{2L}{l}(\lambda^*)^{1-\frac{\alpha}{2}}$, it follows

$$\|\nabla f_{\delta_k}(x_k) - \nabla f(x_k)\| \le \frac{L c_k}{1 - c_k}\|x_k - x^*\| \le \frac{l}{2}\|x_k - x^*\|.$$

Let us denote $e_k = \|x_k - x^*\|$. Then it holds

$$e_{k+1} \le \|I - h(\lambda_k)G_k\| e_k + h(\lambda_k)\|\nabla f(x_k) - \nabla f_{\delta_k}(x_k)\| \le \|I - h(\lambda_k)G_k\| e_k + \frac{h(\lambda_k)l}{2} e_k. \quad (8.30)$$

From [82], Corollary 1.2.1

$$H(x^*) - \tau M e_k I_n \preceq H(x^* + \tau(x_k - x^*)) \preceq H(x^*) + \tau M e_k I_n.$$

Then,

$$\left(l - \frac{e_k}{2}M\right) I_n \preceq G_k \preceq \left(\tilde{L} + \frac{e_k}{2}M\right) I_n,$$
$$\left[1 - h(\lambda_k)\left(\tilde{L} + \frac{e_k}{2}M\right)\right] I_n \preceq I_n - h(\lambda_k)G_k \preceq \left[1 - h(\lambda_k)\left(l - \frac{e_k}{2}M\right)\right] I_n.$$

If we denote with

$$a_k(h(\lambda_k)) = \left[1 - h(\lambda_k)\left(l - \frac{e_k}{2}M\right)\right], \qquad b_k(h(\lambda_k)) = \left[1 - h(\lambda_k)\left(\tilde{L} + \frac{e_k}{2}M\right)\right],$$

we obtain $a_k(h(\lambda_k)) > -b_k(h(\lambda_k))$ as by (8.28) $h(\lambda_k) < \frac{2}{l+\tilde{L}}$.
Then it follows

$$\|I_n - h(\lambda_k)G_k\| \le \max\{a_k(h(\lambda_k)), -b_k(h(\lambda_k))\} = 1 - h(\lambda_k)l + \frac{Mh(\lambda_k)}{2}e_k.$$

From (8.30)

$$e_{k+1} \le \left(1 - \frac{h(\lambda_k)l}{2} + \frac{Mh(\lambda_k)e_k}{2}\right) e_k < e_k$$

if $e_k < \bar{r} = \frac{l}{M}$.

Let us estimate the rate of convergence. Let us define $q_k = \frac{lh(\lambda_k)}{2}$ and $\tilde{m}_k = \frac{Mh(\lambda_k)}{2} = \frac{q_k}{\bar{r}}$. Notice that as $e_k < \bar{r} < \frac{q_k+1}{\tilde{m}_k} = \frac{2}{Mh(\lambda_k)} + \frac{l}{M}$, then $1 - \tilde{m}_k e_k + q_k > 0$. So

$$e_{k+1} \le (1 - q_k)e_k + \tilde{m}_k e_k^2 = e_k \frac{1 - (\tilde{m}_k e_k - q_k)^2}{1 - (\tilde{m}_k e_k - q_k)} \le \frac{e_k}{1 - \tilde{m}_k e_k + q_k}$$
$$\frac{1}{e_{k+1}} \ge \frac{1 + q_k - \tilde{m}_k e_k}{e_k} = \frac{1 + q_k}{e_k} - \tilde{m}_k = \frac{1 + q_k}{e_k} - \frac{q_k}{\bar{r}},$$
$$\frac{1}{e_{k+1}} - \frac{1}{\bar{r}} \ge (1 + q_k)\left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) \ge (1 + q_M)\left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) > 0,$$

with $q_M = \frac{l\theta}{2(\kappa_J^2+\lambda_{\max})}$. Then, we can iterate the procedure obtaining

$$\frac{1}{e_k} \geq \left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) \geq (1+q_M)^{k-\bar{k}} \left(\frac{1}{e_{\bar{k}}} - \frac{1}{\bar{r}}\right),$$
$$e_k \leq \left(\frac{1}{1+q_M}\right)^{k-\bar{k}} \frac{\bar{r} e_{\bar{k}}}{\bar{r}-e_{\bar{k}}},$$

and the convergence of $\|x_k - x^*\|$ to zero follows. □

Note that if we choose $\lambda_{\max} > \max\left\{\gamma\lambda^*, \gamma\bar{\lambda}, \frac{\theta(\tilde{L}+l)}{2}, \lambda^*\left(1+\frac{2L}{l}\right)^{2/(2-\alpha)}\right\}$ we have that it exists $\bar{k}$ such that for $k \geq \bar{k}$, all the iterations are successful and (8.28) is satisfied. Then, Theorem 8.18 shows the local behaviour of our procedure.

8.3 Complexity

In this section we provide a global complexity bound for the procedure sketched in Algorithm 8.1. The analysis is inspired by that reported in [114]. We will prove that the number of iterations required to obtain an ϵ-accurate solution, i.e. a solution such that $\|\nabla f_{\delta_k}(x_k)\| \leq \epsilon$, is $O(\epsilon^{-2})$, as for standard Levenberg-Marquardt methods, see Section 2.4.2.

Notice that the regularization parameter at the current iteration depends on the outcome of the previous iteration and consequently let us define the following sets:

$$S_1 = \{k+1 : \rho_k(p_k^{LM}) \geq \eta_1;\ \|\nabla f_{\delta_k}(x_k)\| < \eta_2/\lambda_k\}, \tag{8.31}$$
$$S_2 = \{k+1 : \rho_k(p_k^{LM}) \geq \eta_1;\ \|\nabla f_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k\}, \tag{8.32}$$
$$S_3 = \{k+1 : \rho_k(p_k^{LM}) < \eta_1\}. \tag{8.33}$$

Let $N_i = |S_i|$ for $i = 1,2,3$, so that the number of successful iterations is $N_1 + N_2$ and the number of unsuccessful iterations is N_3. Moreover S_1 can be split into two subsets

$$S_1 = A \cup B = \{k+1 \in S_1 : \gamma\lambda_k < \lambda_{\max}\} \cup \{k+1 \in S_1 : \gamma\lambda_k \geq \lambda_{\max}\},$$

taking into account that if $k+1 \in S_1$ from the updating rule at step 4.1 either $\lambda_{k+1} = \gamma\lambda_k$ (A), or $\lambda_{k+1} = \lambda_{\max}$ (B).

The analysis is made under the following Assumption:

Assumption 8.19. *Let us assume that the procedure sketched in Algorithm 8.1 is stopped when* $\|\nabla f_{\delta_k}(x_k)\| \leq \epsilon$.

In the following Lemma we provide an upper bound for the number of successful iterations.

Lemma 8.20. *Let Assumptions 8.3, 8.8, 8.11, 8.15 and 8.19 hold. Let k_s be the index of the first successful iteration.*

1. *The number N_1 of successful iterations belonging to set S_1 is bounded above by:*
$$N_1 \leq f_{\delta_{k_s-1}}(x_{k_s}) \frac{2}{\eta_1} \frac{\kappa_J^2 + \lambda_{\max}}{\theta \epsilon^2} = O(\epsilon^{-2}).$$
2. *The number N_2 of successful iterations belonging to set S_2 is bounded above by a constant independent of ϵ:*
$$N_2 \leq f_{\delta_{k_s-1}}(x_{k_s}) \frac{2}{\eta_1} \frac{\kappa_J^2 + \lambda_{\max}}{\theta} \left(\frac{\lambda_{\max}}{\eta_2} \right)^2.$$

Proof. From (8.2), as $\lambda_k \leq \lambda_{\max}$ for all k, it follows

$$m_k(x_k) - m_k(x_k + p_k^{LM}) \geq \frac{\theta}{2} \frac{\|\nabla f_{\delta_k}(x_k)\|^2}{\kappa_J^2 + \lambda_{\max}}.$$

Then, as the iteration is successful

$$\begin{aligned} f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM}) &\geq \eta_1 (m_k^{LM}(x_k) - m_k^{LM}(x_k + p_k^{LM})) \\ &\geq \frac{\eta_1}{2} \frac{\theta \|\nabla f_{\delta_k}(x_k)\|^2}{\kappa_J^2 + \lambda_{\max}}. \end{aligned}$$

For all k it holds $\|\nabla f_{\delta_k}(x_k)\|^2 \geq \epsilon^2$ and in particular for $k \in S_2$

$$\|\nabla f_{\delta_k}(x_k)\|^2 \geq \left(\frac{\eta_2}{\lambda_{\max}} \right)^2.$$

Then

$$\begin{aligned} f_{\delta_{k_s-1}}(x_{k_s}) &\geq \sum_{j \in S_1 \cup S_2} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) \\ &= \sum_{j \in S_1} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) + \sum_{j \in S_2} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) \\ &\geq \frac{\eta_1 N_1}{2} \frac{\theta}{\kappa_J^2 + \lambda_{\max}} \epsilon^2 + \frac{\eta_1 N_2}{2} \frac{\theta}{\kappa_J^2 + \lambda_{\max}} \left(\frac{\eta_2}{\lambda_{\max}} \right)^2, \end{aligned}$$

and the thesis follows.

□

In the following Lemma we provide an upper bound for the number of unsuccessful iterations.

Lemma 8.21. *Let Assumptions 8.3, 8.8, 8.11, 8.15 and 8.19 hold. The number of unsuccessful iterations N_3 is bounded above by a constant independent of ϵ:*

$$N_3 \leq \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma}.$$

Proof. Notice that from Assumption 8.15 it is not possible to have an iteration index in B before the last unsuccessful iteration. Then, if we denote with $\bar{N}$ the last unsuccessful iteration index, if $k < \bar{N}$ is a successful iteration, it belongs to A. Denoting with N_a the number of such iterations, it follows

$$\lambda_{\bar{N}} = \gamma^{N_a+N_3}\lambda_0 \le \lambda_{\max}.$$

Then

$$N_3 \le N_a + N_3 \le \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma},$$

and the thesis follows. □

Then, taking into account the results proved in the previous lemmas, we can state the following complexity result, that shows that the proposed method shares the known complexity properties of the steepest descent and trust-region procedures.

Lemma 8.22. *Let Assumptions 8.3, 8.8, 8.11, 8.15 and 8.19 hold, and let N_T be the total number of iterations performed. Then,*

$$N_T \le f_{\delta_{k_s-1}}(x_{k_s}) \frac{2}{\eta_1} \frac{\kappa_J^2 + \lambda_{\max}}{\theta} \left(\frac{1}{\epsilon^2} + \left(\frac{\lambda_{\max}}{\eta_2} \right)^2 \right) + \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma} = O(\epsilon^{-2}). \qquad (8.34)$$

We underline that $\lambda_{\max}$ and therefore the constant multiplying ϵ^{-2} in (8.34) may be large if κ_J is large. On the other hand in next section we will show that in the applications of our interest $\kappa_J \simeq 1$.

8.4 Chapter conclusion

In this chapter, we have proposed an inexact Levenberg-Marquardt approach to solve large scale nonlinear least squares problems with expensive objective function. The method relies on the solution of a sequence of problems with noisy functions and gradients approximating the original problem. We assume to be able to estimate and reduce the noise, if needed. We propose a rule to decide if the noise level allows to successfully continue with the optimization process and a suitable update of the regularization parameters to handle the noise in the function. Having in mind to work with large scale problems, we handled the linear algebra phase by an iterative solver.

We proved that the resulting approach guarantees global convergence to a solution of the unperturbed problem and that asymptotically the step tends to the direction of the negative perturbed gradient. Then, we performed a local analysis for the arising perturbed steepest descent methods. We also provided a global complexity bound for the proposed method.

CHAPTER

9 Application to data assimilation and machine learning

In this chapter, we analyze the numerical behaviour of the Levenberg-Marquardt method described in Algorithm 8.1. We will denote it as LMN (Levenberg-Marquardt method for Noisy problems). We show the results of its application to nonlinear least squares problems arising in data assimilation and machine learning. These problems can be written as

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|R(x)\|^2 + \frac{1}{2}\|x\|^2 = \frac{1}{2}\sum_{j=1}^{m} R_j(x)^2 + \frac{1}{2}\|x\|^2, \tag{9.1}$$

with $R_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, \ldots, m$.

Specifically we consider four test problems:

- We consider two instances of the same data assimilation problem. We want to reconstruct the initial state of a system whose evolution is governed by a 1D wave equation, given observations at subsequent time instants. They are described in Section 9.1.

- We consider two instances of the same machine learning problem. The aim is to perform a logistic regression on given data samples. One dataset arises from a real world application in the field of turbomachinery design. They are described in Section 9.2.

In these test problems the objective function is a sum over a large number m of terms and it is therefore expensive to evaluate. Our method is suitable for problems for which m is large but it is not huge, as we need to estimate the noise in the function approximations. In such cases it is possible to compute the exact function but it is expensive. Then sporadic computations of the full function are possible, but using it along all the optimization process would be computationally heavy. Such situations arise for example in applications where obtaining the labels of the samples is expensive, so one cannot expect to have at disposal a huge number of samples, but this number is still too large to be efficiently handled by a standard inexact method. This is the case of the real life application we present in Section 9.2.2, where the computation of the labels of the training samples is the result of the expensive solution of high-dimensional partial differential equations. It is impossible to obtain too many samples, as for industrial needs the whole process must be performed quickly.

In some of the test problems the noise in the function arises from the fact that approximations are built considering not all the terms in the sum. Each term corresponds to an observed data, that are usually referred to as samples or observations, and the terms to be considered are selected through the use of subsampling techniques. At each iteration a subset of the available samples indexes $X_k \subseteq \{1,\dots,m\}$ is randomly selected such that $|X_k| = K_k$ for each k. The approximations R_{δ_k} to R are built considering just the K_k terms corresponding to the samples indexed by X_k. Each time condition (8.7) is not verified we increase the size of the subsampled set adding new samples (randomly selected) to decrease the noise level. The size increase is performed in a linear way by a factor K_*, so that if the loop 1-2 of Algorithm 8.1 is performed n_k times it holds

$$|X_{k+1}| = K_*^{n_k}|X_k|. \tag{9.2}$$

Notice that other updates, different from linear, could be used affecting the speed of convergence of the procedure, see for example [15, 43]. Moreover, the subsampling is performed in a random way, even if for some problems, like data assimilation problems, it is possible to devise more efficient strategies. Such strategies, taking into account the particular structure of the problem, lead to a quicker decrease in the noise, the number of samples being the same [46]. We will not consider these aspects here.

The procedure was implemented in MATLAB and run using MATLAB 2015A on an Intel(R) Core(TM) i7-4510U 2.00GHz, 16 GB RAM; the machine precision is $\epsilon_m \sim 2{\cdot}10^{-16}$. We run LMN with $\eta_1 = 0.25$, $\eta_2 = 10^{-3}$, $\gamma = 1.001$, $\alpha = 0.9$, $\lambda_{\max} = 10^6$, $\lambda_{\min} = 1$ for data-assimilation problems and $\lambda_{\min} = 0.1$ for the machine-learning problems. In order to compute the step, the linear subproblems (8.5) are solved by the Matlab function `cgls` available at [95], that implements conjugate gradient (CG) method for least squares. We set the stopping tolerance to 10^{-1}, that is we used $\epsilon_k = 10^{-1}$ in (8.4). We set to 20 the maximum number of iterations `cgls` is allowed to perform. We will see in the numerical tests that the average number of `cgls` iterations per outer iteration is really low, and this maximum number is never reached.

In the numerical tests, the performance of LMN has been compared to that of inexact Levenberg-Marquardt methods, for which the linear algebra phase is handled in the same way, but they employ the unperturbed function and gradient. In particular, we have considered:

- FLM, the full inexact Levenberg-Marquardt method, i.e. the procedure described in Algorithm 8.1, but run computing at each iteration the exact value of the objective function,
- SLM, an inexact Levenberg-Marquardt method based on a standard update of the regularization parameters as in (2.20) (with $\lambda_0 = 0.1$, $\gamma_0 = 2, \gamma_1 =$

$0.5, \eta_1 = 0.25, \eta_2 = 0.75$). It also uses exact values of the objective function.

For problems in which subsampling techniques are employed, using the exact function amounts to use all the available samples, so that $K_k = m$ for all k and the noise is zero along all the optimization process. The difference between these two approaches lies uniquely in the update of the Levenberg-Marquardt parameter. We considered both these inexact methods, as in case noise is not present it is not necessary to employ the update of the parameters in Algorithm 8.1. We want to show that the proposed noise control mechanism makes the procedure computationally less expensive, even compared to a more standard inexact Levenberg-Marquardt method (that is supposed to converge more quickly), without loss in solution accuracy.

To evaluate the performance of LMN and compare it to that of the other inexact methods, we use two different counters, one for the nonlinear function evaluations and one for matrix-vector products involving the Jacobian matrix (transposed or not). This includes both the products necessary to compute the gradient, to evaluate the model and those performed in the conjugate gradients iterations. We remind that `cgls` requires two matrix-vector products per iteration, plus another two in the initialization phase.

Notice that to take into account the use of subsampling techniques, the counters are intended to be cost counters, i.e. each function evaluation and each product is weighted according to the size of the samples set. The cost of a function evaluation or of a matrix-vector product is considered unitary when the full samples set is considered, otherwise it is weighted as $\frac{|X_k|}{m}$. If a sampling technique is not used, than all the weights will be one and the counters will simply count the number of function evaluations or of products.

The solution x computed by the LMN is compared to x^*, approximation computed by the FLM, stopped when the norm of the gradient reaches 10^{-6}. The distance is measured by the *Root Mean Square Error* (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x^*(i) - x(i))^2}{n}}.$$

In the tables, the column heads have the following meanings: **it**: counter of outer iterations of the Levenberg-Marquardt method, $\mathbf{CG_{it}}$: average number of `cgls` iterations per outer iteration, $\mathbf{cost_f}$: function evaluation cost counter, $\mathbf{cost_p}$: products cost counter, **RMSE**: root mean square error, $\mathbf{save_f}$, $\mathbf{save_p}$: savings gained by LMN compared to FLM respectively in function evaluations and products (computed from $\mathbf{cost_f}$ and $\mathbf{cost_p}$). In the tables, we report just results referring to FLM, and not to SLM, as their performance is comparable; for each test problem, we comment on the difference between FLM and SLM in the text.

When a sampling technique is used also the value $|\mathbf{X_{it}}|$ of the cardinality of the samples set at the end of the process is reported.

9.1 Data assimilation problems

In this section we consider the data assimilation problem described in [46]. The basic purpose of a data assimilation problem is to combine different sources of information to estimate at best the state of a system. These sources generally are observations and a numerical model. Usually a model is jointly used with observations, as they are sparse or partial in geophysics, and the model is used to interpolate the information from observations to unobserved regions or quantities [14].

We consider a one-dimensional wave equation system, whose dynamics is governed by the following nonlinear wave equation:

$$\frac{\partial^2 u(z,t)}{\partial t^2} - \frac{\partial^2 u(z,t)}{\partial z^2} + \tilde{f}(u) = 0, \quad \tilde{f}(u) = \mu e^{\nu u}, \tag{9.3a}$$

$$u(0,t) = u(1,t) = 0, \tag{9.3b}$$

$$u(z,0) = u_0(z), \ \frac{\partial u(z,0)}{\partial t} = 0, \tag{9.3c}$$

$$0 \le t \le T, \ 0 \le z \le 1. \tag{9.3d}$$

The system is discretized using a mesh involving $n = 360$ grid points for the spatial discretization and $N_t = 64$ for the temporal one. We assume to have an observation for each grid point, so that the total number of observations is $m = n \cdot N_t = 23040$. We denote with $x(t_j)$ the state vector, namely the solution of the nonlinear model (9.3) at time t_j. We assume to have at disposal a priori estimate $x_b \in \mathbb{R}^n$ that is called the background vector, and a set of observations. With $y_j \in \mathbb{R}^{m_j}$ we denote the vector of observations at time t_j and with H_j the operator modelling the observation process at the same time.

We look for the initial state $u_0(z)$, which is possible to recover solving the following data assimilation problem [46]:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_{j=1}^{N_t} \|H_j(x(t_j)) - y_j\|^2_{O_j^{-1}} \tag{9.4}$$

where, given a symmetric positive definite matrix M, $\|x\|^2_M$ denote $x^T M x$. Matrices $B \in \mathbb{R}^{n\times n}$ and $O_j \in \mathbb{R}^{m_j \times m_j}$ represent the background-error covariance and the observation-error covariance at time t_j respectively. In our tests we build the background vector and the observations from a chosen initial true state x_T by adding a noise following the normal distribution $N(0,\sigma_b^2)$ and $N(0,\sigma_o^2)$ respectively. We have chosen $\sigma_b = 0.2$, $\sigma_o = 0.05$. For further details on the test problem see [46]. We can reformulate (9.4) as a least squares problem (9.1), defining

$$R(x) = \begin{bmatrix} \|H_1(x(t_1)) - y_1\|_{O_1^{-1}} \\ \vdots \\ \|H_{N_t}(x(t_{N_t})) - y_{N_t}\|_{O_{N_t}^{-1}} \end{bmatrix},$$

where $(H_j(x(t_j)) - y_j) \in \mathbb{R}^{m_j}$ for $j = 1,\dots,N_t$. In general high accuracy is not required in practical applications, so the optimization process is stopped as soon as the residuals are detected to be Gaussian. As a normality test we employ the Anderson-Darling test [2], which tests the hypothesis that a sample has been drawn from a population with a specified continuous cumulative distribution function Φ, in this case the Gaussian distribution. Assuming to have a sample of n ordered data $\{x_1 \le x_2 \le \cdots \le x_{n-1} \le x_n\}$, a statistic is built in the following way:

$$W_n^2 = -n - \frac{1}{n}\sum_{i=1}^{n}(\ln(\Phi(x_i)) + \ln(1-\Phi(x_{n-i+1}))).$$

If W_n^2 exceeds a given critical value the hypothesis of normality is rejected with some significance level. We used the critical value for significance level 0.01 which is 1.092 [100]. This stopping criterion is used for all the considered procedures.

9.1.1 Data assimilation problem with subsampling

In this section we consider the case in which the noise in the function arises from the fact that not all the available observations are considered along all the optimization process. The full problem (9.1) is obtained when all the observations are considered, in this case $m_j = 360$ for every j, while the approximations R_{δ_k} are obtained selecting randomly K_k observations among the available ones, until the desired cardinality of the samples sets is reached. In this case the vectors $(H_j(x(t_j)) - y_j)$ have dimension $m_j \le 360$, and it may be $m_j \ne m_i$ if $i \ne j$.

We consider two different problems of the form (9.4), corresponding to two different values of μ in (9.3a). We consider a mildly nonlinear problem, choosing $\mu = 0.01$ because this is usually the case in practical data assimilation applications and then we increase μ to 0.5 to consider the effect of the nonlinearity on our procedure.

In these tests we assume the covariances matrices to be diagonal: $B = \sigma_b^2 I_n$ and $O_j = \sigma_o^2 I_{m_j}$ for each j.

We denote with x^* the solution approximation found by FLM, computed stopping the procedure as soon as $\|\nabla f_{\delta_k}(x_k)\| < 10^{-6}$. If we compare this approximation to the true state x_T we obtain a RMSE $\simeq 5.2e-3$. Then, we study the effect of the presence of noise in the function arising from the use of subsampling techniques and we compare the solution found by LMN to x^*. Taking into account (7.2) the noise level δ_k is approximated in the following way. At the beginning of the iterative process δ_0 is set to $|f_{\delta_0}(x_0) - f(x_0)|$. Then, it is left constant and updated only when condition (8.7) is not satisfied as follows

$$\delta_k \simeq |f_{\delta_k}(x_k) - f(x_k)|. \tag{9.5}$$

We stress that this computation is not particularly expensive as it does not affect the product counter and marginally contributes to the function evaluations counter. In fact, the evaluation of the full function is required only when condition (8.7) is not met and not at each iteration. Remarkably, in this case this evaluation is performed just once for a fixed iteration index, even in case the noise is reduced more than once in the loop 1-2 of Algorithm 8.1, as x_k is fixed.

The performance of our procedure is affected by mainly three choices: the cardinality of the starting set of observations K_0, factor K_* in (9.2) and the parameter κ_d in (8.7). The choice of K_* determines how much the noise is reduced at each loop at steps 1-2 of Algorithm 8.1. A too small value could lead to a too low noise reduction, gaining a noise level that still does not satisfy condition (8.7), so that the loop should be performed $n_k > 1$ times. Each noise reduction requires the computation of a trial step through the solution of a linear system (8.3) of increasing size, so it is advisable to consider a reasonable increase in the subsets size. Again too small values of κ_d generally lead to a too frequent noise reductions. In this section we investigate the effect of parameter κ_d combined with different values of K_0, while K_* is kept fixed, $K_* = 1.5$. Its effect will be studied in Section 9.2.

We run the procedure choosing two different values of K_0, combined with different values of κ_d, Tables 9.1 and 9.2 refer to problem $\mu = 0.5$ for $K_0 = 2000$ and $K_0 = 5000$ respectively, while Table 9.3 refers to test problem $\mu = 0.01$ for $K_0 = 2000$ and $K_0 = 7000$. In the first column we report the results of the optimization process performed by FLM and in the others those corresponding to LMN with different choices of κ_d. In the last two rows the savings gained by LMN in function evaluations $\mathbf{save_f}$ and products $\mathbf{save_p}$ are reported. In the tests presented in this section the number of iterations is quite low, so the rule for updating the regularization parameter does not have a great impact on the procedure and the performance of FLM method and of SLM method is quite the same.

We notice that LMN requires on average a higher number of `cgls` iterations than FLM and this is due to the need of recomputing the step when (8.7) is not satisfied. However, notice that each `cgls` iteration will be less expensive than those required by the FLM as the systems involved have smaller dimension. This number is affected by the choice of parameter κ_d, and generally it decreases with κ_d. This is less evident for $\mu = 0.5$, while it is more evident for $\mu = 0.01$. Moreover, the value of κ_d does not affect the number of outer iterations performed by LMN, while it has a deep impact on the procedure cost, as we can see from the significant variation of function evaluations and matrix-vector products counters.

We notice that in all cases our procedure is much less expensive than FLM, and consistent savings are provided by higher values of κ_d. In these cases indeed the noise is reduced less frequently, as condition (8.7) is more likely satisfied, and as a result the overall process is performed with less observations and is less expensive, at the cost however of a less accurate solution.

Indeed, if κ_d is too large the noise control strategy is not effective, the noise may be never reduced and the sequence may approach a solution of the noisy problem, that can be a bad approximation to that of (9.4). In Figure 9.1 we compare solution approximations for $\mu = 0.5$ provided by: FLM (up left), LMN with $K_0 = 5000$ and $\kappa_d = 10$ (up right), LMN with $K_0 = 2000$ (bottom left) and $K_0 = 5000$ (bottom right) and $\kappa_d = 10000$. In all the plots the solid line represents the true state x_T and the dotted line the computed solution approximation. It is evident that in the bottom left plot, corresponding to the last column of Table 9.1, the solution found is less accurate. In fact, due to the high value of κ_d the noise is never

	FLM	$\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	12	12	12	11
$\mathbf{CG_{it}}$	2.4	5.4	4.9	4.2	4.2	3.9
$\mathbf{cost_f}$	10	9.7	6.1	3.3	3.2	2.0
$\mathbf{cost_p}$	67	46.1	26.8	14.9	13.5	10.3
$\mathbf{\vert X_{it}\vert}$	23040	15188	6750	3000	3000	2000
RMSE	1.2e-2	3.0e-2	2.8e-2	3.8e-2	4.4e-2	7.8e-2
$\mathbf{save_f}$		3%	39%	67%	68%	80%
$\mathbf{save_p}$		31%	60%	78%	80%	85%

Table 9.1: *Performance of LMN for test problem $\mu = 0.5$ and $K_0 = 2000$.*

	FLM	$\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	11	12	12	12
$\mathbf{CG_{it}}$	2.4	4.1	3.9	4.0	4.1	3.7
$\mathbf{cost_f}$	10	9.1	6.5	5.1	4.9	3.6
$\mathbf{cost_p}$	67	54.8	37.2	34.6	32.9	27.3
$\mathbf{\vert X_{it}\vert}$	23040	16875	11250	7500	7500	5000
RMSE	1.2e-2	2.7e-2	3.0e-2	2.1e-2	2.1e-2	2.7e-2
$\mathbf{save_f}$		9%	35%	49%	51%	64%
$\mathbf{save_p}$		18%	44%	48%	51%	59%

Table 9.2: *Performance of LMN for test problem $\mu = 0.5$ and $K_0 = 5000$.*

reduced and the problem is solved considering just the samples in the initial subset, which are not sufficient to obtain the same accuracy gained by the FLM. Then κ_d should not be chosen too high, especially if K_0 is small.

On the other hand, if K_0 is large enough one can expect to gain good solution accuracy even with a higher κ_d. For example $K_0 = 5000$ is large enough to obtain a good solution approximation, so the best performance is obtained with large κ_d. Then, κ_d should be chosen in relation to K_0 and according to the desired solution accuracy.

In Figure 9.2 we relate the savings gained with the corresponding solution accuracy. The solid lines refer to Table 9.1 while the dotted ones to Table 9.2. In the left plot we report the savings in function evaluations (lines marked by stars) and in matrix-vector products (lines marked by circles), while in the right plot the RMSE, versus κ_d . If $K_0 = 5000$ the accuracy is almost the same for all choices of κ_d but the savings increase with κ_d, while if $K_0 = 2000$ the most significant savings are obtained choosing large κ_d, but at the expense of a lower solution accuracy.

Notice also that in the tests the final value $|X_{it}|$ is always less than m, which confirms that it is not necessary to use all the available observations to obtain a good solution approximation.

In Figure 9.3 we report as an example for problem $\mu = 0.5$ and $K_0 = 2000, \kappa_d =$ 10 the behaviour of the noise (left plot) and that of the error (right plot) through iterations. We underline that condition (8.7) is not violated at each iteration, then

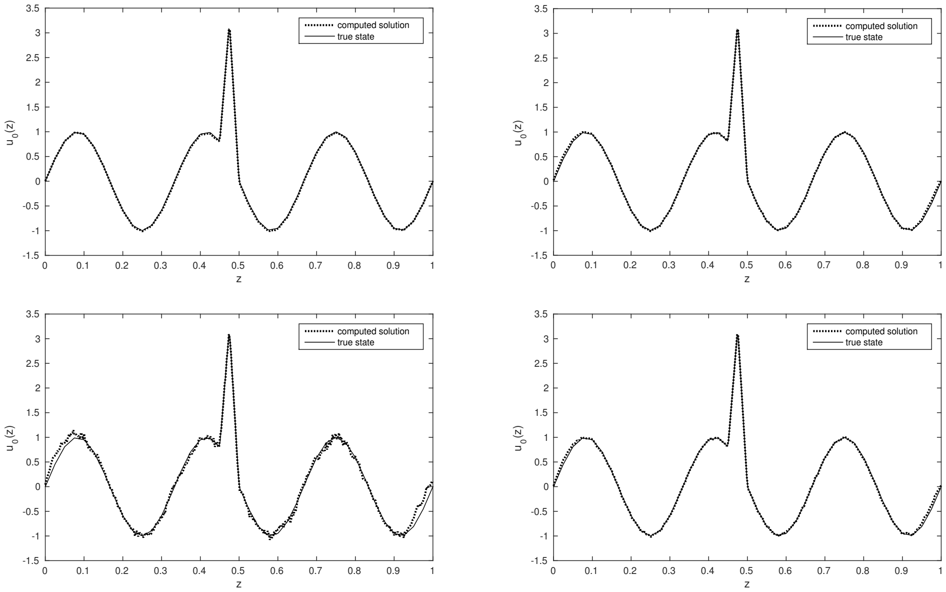


Figure 9.1: *Problem $\mu = 0.5$. Comparison of true state (solid line) and computed solution (dotted line) computed by FLM (up left), LMN with $K_0 = 5000$ and $\kappa_d = 10$ (up right), LMN with $K_0 = 2000$ (bottom left) and $K_0 = 5000$ (bottom right) with $\kappa_d = 10000$.*

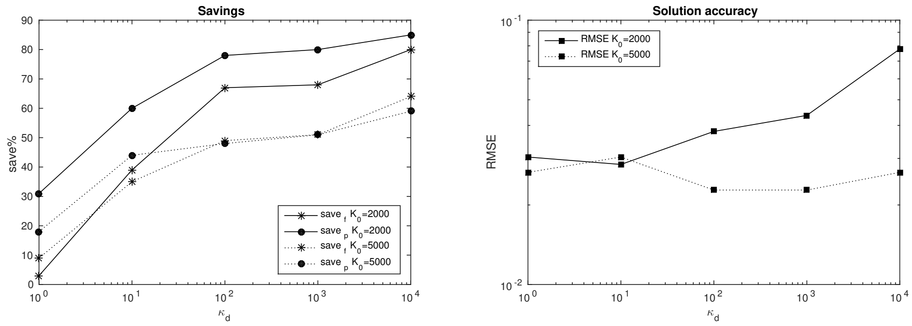


Figure 9.2: *Left plot: save$_f$ (lines marked by a star) and save$_p$ (lines marked by a circle) for tests in Tables 9.1 (solid lines), 9.2 (dotted lines). Right plot: corresponding solution accuracy.*

the noise is kept fixed for some consecutive iterations, and the evaluation of the full function, by the computation of the remaining components, is only sporadically necessary.

Regarding the choice $\mu = 0.01$ we report statistics in Table 9.3. The problem is almost linear, so it is solved in few iterations. Due to the really low number of iterations, it is advisable to start with a rather large initial set to avoid converging to a solution of the noisy problem and to gain the same accuracy as FLM. In this case the procedure is less sensitive to the choice of parameter κ_d than in the other case and only significant changes in κ_d affect its performance. Also for this problem the use of LMN provides significant savings compared to FLM.

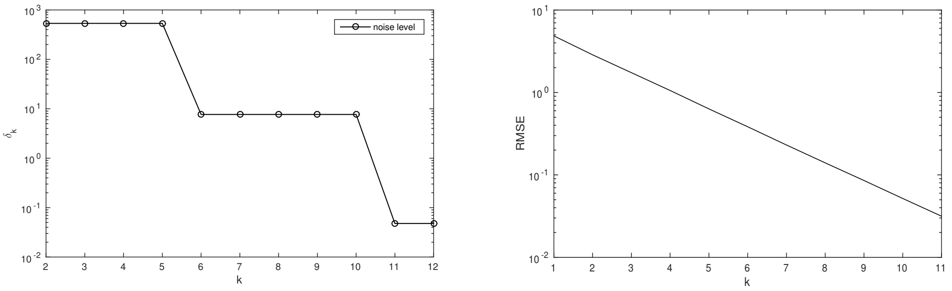


Figure 9.3: *Problem $\mu = 0.5$, $K_0 = 2000$, $\kappa_d = 10$. Left: log plot of noise level versus iterations. Right: log plot of the RMSE versus iterations.*

		$K_0 = 2000$			$K_0 = 7000$	
	FLM	$\kappa_d = 1$	$\kappa_d = 10,100$	$\kappa_d = 1000$	$\kappa_d = 1,10$	$\kappa_d = 100,1000$
it	3	3	4	3	3	3
$\mathbf{CG_{it}}$	3.0	12.3	9.5	6.0	5.7	4.0
$\mathbf{cost_f}$	4	2.9	3.5	1.3	3.1	1.9
$\mathbf{cost_p}$	27	12.6	10.8	3.9	15.3	10.0
$\mathbf{\lvert X_{it}\rvert}$	23040	6750	4500	2000	10500	7000
RMSE	6.8e-3	2.0e-2	1.1e-2	3.4e-2	1.5e-2	1.6e-2
$\mathbf{save_f}$		27%	12%	67%	22%	52%
$\mathbf{save_p}$		53%	60%	85%	43%	63%

Table 9.3: *Performance of LMN for test problem $\mu = 0.01$.*

9.1.2 Data assimilation problem with non-diagonal covariance matrix.

In this section we consider the same problem as in the previous one, but we focus on the case in which the background-error covariance matrix is not diagonal. Often the dimension of this matrix is large and it is not even available, and only its action onto a vector is known. Then it is not straightforward to invert it to evaluate the objective function. It is necessary to use an iterative method to evaluate $B^{-1}(x_k - x_b)$ solving approximately the system

$$Bz = x_k - x_b. \tag{9.6}$$

The accuracy in the solution of such systems has to be fixed in advance and the noise in the objective function arises from the fact that it is the result of this computation, that is performed with a certain accuracy.

To simulate such situation we choose $B = \sigma_b^2 T_b$ where T_b is a three-diagonal Toeplitz matrix such that $T_{i,j} = e^{-|i-j|}$ if $i = j, j+1, j-1$ [30]. We solve the systems (9.6) with `cgls` and we stop the procedure as soon as

$$\|\rho_k\| = \|Bz - x_k + x_b\| \le tol_k \|x_k - x_b\|,$$

where tol_k is a positive parameter to be set. We set to 40 the maximum number of iterations. To make the comparisons easier, we consider as FLM the procedure that employs an accurate solution of the systems (9.6), i.e. choosing $tol_k \equiv 10^{-6}$

	it	$\mathbf{CG_{it}}$	$\mathbf{cost_p}$	RMSE	$\mathbf{save_p}$
FLM	11	42.0	981	4.1e-2	
LMN	10	23.4	526	4.9e-2	46%

Table 9.4: *Performance of LMN for data assimilation problem with non-diagonal background-error covariance matrix.*

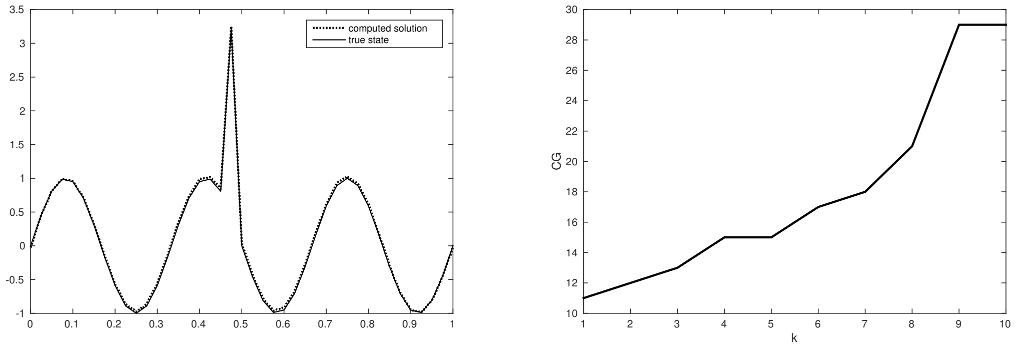


Figure 9.4: *LMN applied to data assimilation problem with non-diagonal matrix. Left: solution approximation. Right: total number of* `cgls` *iterations.*

in (8.4). The noisy approximations are built solving the systems with a looser tolerance.

The error between the exact and the approximated function can be estimated as

$$\delta_k \leq \frac{1}{2}\|x_k - x_b\|^2\|B^{-1}\|tol_k.$$

The eigenvalues of matrix B are known, so we can bound $\|B^{-1}\|$ with the reciprocal of the smallest eigenvalue of B: $\frac{1}{\sigma_b^2(1-2\sqrt{e^{-2}})}$.

We start the optimization process choosing $tol_0 = 10^{-2}$ and we decrease it by a factor 10 each time (8.7) is not satisfied, where we choose $\kappa_d = 1$.

In Table 9.4 we compare the performance of FLM and LMN. The number of function evaluations is not reported, as it is not significant in this case because its cost depends on tol_k and is better measured by the number of `cgls` iterations. In this context $cost_p$ includes all the products performed by `cgls` both to find the step and to invert B. As expected, solving the systems with high accuracy requires far more iteration of the linear solver and the savings provided by our procedures are considerable. This is obtained without loosing accuracy in the solution approximation. In Figure 9.4 we report the plot of the solution approximation computed by LMN on the left and the plot of the total number of `cgls` iterations (including those required both to find the step and to solve (9.6)) per outer iteration on the right . The number of `cgls` iterations required to invert matrix B increases as δ_k is decreased by our noise check strategy, tol_k is indeed initially set to 10^{-2} and brought to 10^{-4} toward the end of the process. The tight tolerance 10^{-6} is anyway never reached.

9.2 Machine learning problems

In this section we consider the application of our method to problems arising in machine learning. The aim of machine learning techniques is that of building models, usually called meta-models, that are able to learn a task, for example to approximate a function or classify data, from given examples arising from the considered application. The examples are usually couples given by a n-dimensional vector called *sample* and a target value called *label* that corresponds to the result of the process that the model is expected to perform. For example in case of regression problems the target is the value of the function to predict, in classification problems the label of the class the sample belongs to. The employment of machine learning meta-models is based on two different steps, [16]:

- a *training phase*, in which the model learns the task it has to perform from some given samples that form the so-called *training set*. Specifically, in common practice one selects a class of functions in which the model will be chosen. The candidate models depend on some hyper-parameters that need to be tuned. Then, during this phase optimization processes are performed for different values of the hyper-parameters to choose a set of candidate models in the selected class, to pinpoint a small subset of viable candidates. Then, a *validation set* is used in order to estimate the prediction error of each of these remaining candidates and to select the best performing of them which is chosen as the selected model.
- an *execution phase*, in which the trained model is used to perform the learned task on new samples whose labels are unknown, that form a *testing set*. The testing set is used to estimate the generalized performance of this selected function.

In this section, we will consider the problem of binary classification. It is assumed that the samples can be divided into two classes, labelled as $+1$ and -1. The aim is to learn from given data how to divide them into the two classes and to assign a new sample to the right one.

We suppose to have at disposal a training set composed of pairs $\{(z^i, y^i)\}$ with $z^i \in \mathbb{R}^n$, $y^i \in \{-1, +1\}$ and $i = 1, \dots, m$, where y_i denotes the correct sample classification. We perform a logistic regression, then we consider as a training objective function the logistic loss with l_2 regularization [15]:

$$f(x) = \frac{1}{2m} \sum_{i=1}^{m} \log(1 + \exp(-y^i x^T z^i)) + \frac{\sigma}{2} \|x\|^2.$$

Following [15], the regularization term is set to $\sigma = \frac{1}{m}$ so that our model becomes:

$$f(x) = \frac{1}{2m} \sum_{i=1}^{m} \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2m} \|x\|^2. \tag{9.7}$$

The training phase in logistic regression consists of minimizing function (9.7). To this aim we employ the proposed LMN method. Since this is a convex nonlinear programming problem, it could potentially be solved also by a subsampled Newton method. Here, for sake of gaining more computational experience with our approach, we reformulate it as a least squares problem (9.1), scaled by m, where $R:\mathbb{R}^n \to \mathbb{R}^m$ is given by

$$R(x) = \begin{bmatrix} \sqrt{\log(1+\exp(-y^1 x^T z^1))} \\ \vdots \\ \sqrt{\log(1+\exp(-y^m x^T z^m))} \end{bmatrix}.$$

We build the approximations f_{δ_k} as:

$$f_{\delta_k}(x) = \frac{1}{2K_k} \sum_{i\in X_k} \log(1+\exp(-y^i x^T z^i)) + \frac{1}{2K_k}\|x\|^2.$$

We stop the procedure when $\|\nabla f_{\delta_k}(x_k)\| \le 10^{-4}$.

Once the problem is solved, the computed solution x^* is used to classify the samples in the testing set. The execution phase indeed consists of computing labels $\hat{y}^i$ that estimate y^i for z^i in the testing set as follows:

$$\hat{y}^i = \begin{cases} +1 \text{ if } \sigma(z^i) \ge 0.5 \\ -1 \text{ if } \sigma(z^i) < 0.5, \end{cases} \qquad \sigma(z) = \frac{1}{1+\exp(-z^T x^*)}.$$

If the correct classification of the samples in the testing set is known, the classification error can be simply computed comparing directly the predicted and the expected labels. In the other case, following [15], the classification error is defined as

$$\mathbf{e_{te}} = \frac{1}{2\tilde{m}} \sum_{i=1}^{\tilde{m}} \log(1+\exp(-\hat{y}^i x^{*T} z^i)), \tag{9.8}$$

which corresponds to $f(x^*)$ in (9.7) omitting the regularization term $\frac{1}{2\tilde{m}}\|x\|^2$ and setting $y^i = \hat{y}^i$.

We consider two different datasets. First, the CINA dataset available from [19] that is typically used to test methods for machine learning problems [15, 73], and then a dataset arising from a real life problem, the design of turbomachinery components.

9.2.1 CINA dataset

For this dataset the number of features is $n = 132$ and the samples are divided into a training set of size $m = 16033$ and a testing set of size $\tilde{m} = 10000$. The testing error is computed as in (9.8).

We start the optimization process with $K_0 = 1000$. Parameter κ_d is set to 100. The noise level is computed as outlined in Section 9.1.1 (see (9.5)).

We study the effect of the choice of K_* in (9.2) on the procedure performance. Then, we fix $K_0 = 132$ and $\kappa_d = 100$.

Notice that in these runs all the available training samples are used during the optimization process, so that the final value $|X_{it}|$ always reaches the maximum

	FLM	$K_* = 1.1$	$K_* = 1.5$	$K_* = 2$	$K_* = 2.5$	$K_* = 3$	$K_* = 3.5$
it	52	82	43	38	39	34	53
$\mathbf{CG_{it}}$	5.7	8.5	8.0	7.5	7.3	7.2	5.5
$\mathbf{cost_f}$	53	19.8	14.1	15.9	21.2	16.5	37.7
$\mathbf{cost_p}$	808	671.2	351.3	316.7	400.7	310.4	521.1
$\mathbf{\lvert X_{it}\rvert}$	16033	16033	16000	16033	16033	16033	16033
RMSE	6.0e-2	1.0e-1	6.6e-2	5.4e-2	4.7e-2	4.1e-2	3.9e-2
$\mathbf{e_{te}}$	0.185	0.180	0.181	0.187	0.184	0.183	0.185
$\mathbf{save_f}$		63%	74%	70%	60%	69%	29%
$\mathbf{save_p}$		17%	56%	61%	50%	62%	35%

Table 9.5: *Performance of LMN for machine learning test problem for different values of K_*.*

value m. The results reported in Table 9.5 show that for every choice of K_* LMN provides significant savings compared to FLM.

For this test problem, differently from those presented in the previous section, the number of iterations is quite high. Then, the choice of the updating rule for the regularization parameter deeply affects the cost of the procedure. When SLM is considered, the convergence is much faster compared to FLM (the number of outer iterations performed is half of those required by FLM), but the average number of `cgls` iterations per outer iteration is more than the double. Indeed, the regularization parameters are decreasing and the linear systems to be solved are less regularized, so that `cgls` requires more iterations to converge. As a result, the cost of SLM compared to FLM is lower in terms of function evaluations ($cost_f = 23$) but it is almost the same in terms of products ($cost_p = 738$). Thus, the proposed procedure is also less expensive compared to SLM.

The savings in function evaluations and products result in reduced computational time. For example for $K_* = 2$ the time is reduced of about 1/3 on the machine we performed the runs on. The solution accuracy, as it is shown both by the RMSE and the testing error, is not worsted. The counters anyway are affected by the choice of κ_d, both too small and too large values lead to a more expensive LMN procedure. The effect of small parameter values is clearly shown in Figure 9.5. In each plot the values of n_k (dotted line) and the number of `cgls` iterations (dashed line) for each outer iteration k are reported, for $K_* = 1.1$ (up left), $K_* = 2$ (up right) and $K_* = 3.5$ (bottom). The values of n_k indicate how many times loop 1-2 in Algorithm 8.1 is performed ($n_k = 1$ means that the noise is reduced once at iteration k, $n_k = 0$ means that the noise is kept constant for that iteration). We notice that the number of `cgls` iterations performed in an outer iteration in which the noise is reduced is much higher than that required by iterations in which it is kept constant, as the linear system (8.3) is solved more than once. When K_* is small, the noise is reduced of a small amount when condition (8.7) is not satisfied, and this lead to more frequent noise reductions and then to perform a higher number of linear iterations, as it is shown in Table 9.5 and in Figure 9.5. On the other hand, with large values of K_* the noise is reduced less often, but a too large choice leads K_k to quickly reach the maximum value m, so that many expensive

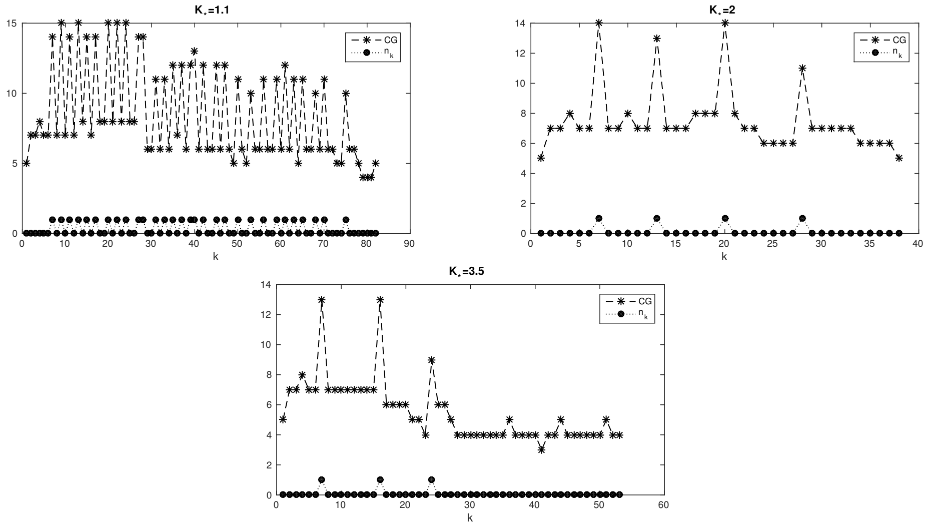


Figure 9.5: *Values of n_k (dotted line) and number of linear iterations (dashed line) per outer iteration, for $K_* = 1.1$ (up left), $K_* = 2$ (up right) and $K_* = 3.5$ (bottom).*

iterations are performed and again the computational costs are higher. In the left plot of Figure 9.6 we report values of $|X_k|$ along the iterations for different values of K_*. We notice that when K_* is small the size is often increased of a small amount while for larger values it raises quickly.

In the right plot of Figure 9.6 we compare the cost of matrix-vector products at each iteration of LMN for various K_* and of the FLM (solid line). We can see that significant savings are obtained at the beginning of the optimization process, due to the reduced size of the subproblems, which compensate the greater costs in the final stage, when the samples subsets are of size close to m and additional linear iterations are required when condition (8.7) is not satisfied.

Notice that in all the tests performed the average number of `cgls` iterations is generally low and the maximum number of allowed iterations is never reached. This is due to the low accuracy we solve the linear systems with, which anyway is enough to achieve convergence. Moreover our method still gains the benefits of a quicker convergence compared to first-order methods, and at no great expense, as the number of linear iterations is extremely low. To show this, we used the Matlab function `steep`, available at [71], implementing the steepest descend method to solve the noise free problem. Its performance can then be compared to that of FLM. After 1000 iterations the desired accuracy was not yet reached and the norm of the gradient was of the order of 10^{-3}. Then our procedure results to be much quicker, despite the fact that asymptotically the step tends to a steepest descent step.

Finally we want to highlight that the evaluation of the full function is needed only sporadically (only when condition (8.7) is not satisfied) and that such evaluations do not deeply affect the procedure's cost. We compare the results provided by the same procedure, in which however the noise level is not computed from (9.5)

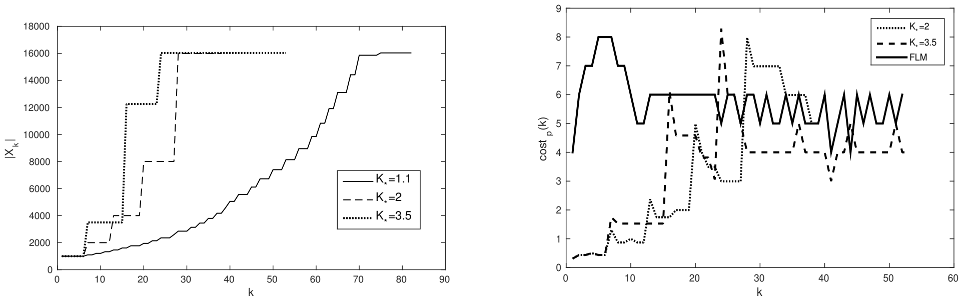


Figure 9.6: *Left: values of $|X_k|$ versus iterations for different values of K_*. Right: Comparison of matrix-vector cost counter for FLM (solid line) and LMN with various choices of K_*.*

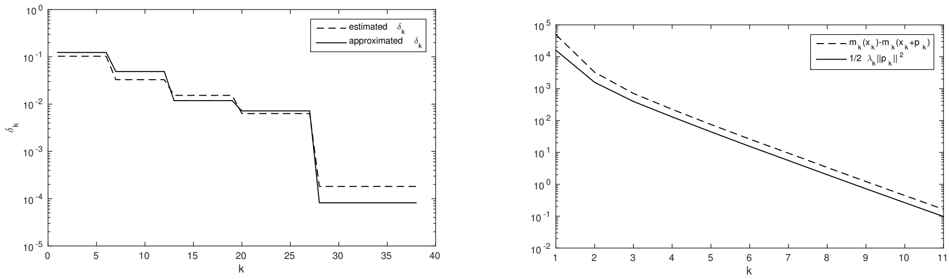


Figure 9.7: *Left: comparison of approximated noise level (solid line) and estimated noise level (dashed line) during run of LMN. Right: decrease in the model $m_k(x_k) - m_k(x_k + p_k^{LM})$ versus iterations (dashed line) compared to decrease of $\frac{1}{2}\lambda_k \|p_k^{LM}\|^2$ (solid line).*

but it is estimated in the following way:

$$\delta_k \simeq \frac{\sqrt{2(m-K_k)}}{K_k}, \quad \text{with} \quad K_k = |X_k|. \tag{9.9}$$

This approximation is based on the observation that if the components $R_i(x)$ of $R(x)$ were Gaussian, $\sum_{i \notin X_k} R_i(x)^2$ would follow a Chi-squared distribution with standard deviation $\sqrt{2(m-K_k)}$. Even if the normality assumption does not hold, this estimation works well in practice. In the left plot of Figure 9.7 we consider LMN with $K_* = 2$ and we compare the approximation of the noise provided by (9.5) (solid line) with the noise estimated through (9.9) (dashed line). The noise estimate is good enough to ensure the procedure run with the estimated noise (LMN$_{est}$) to achieve the same performance as that run approximating the noise via (9.5) (LMN$_{appr}$), as it is shown in Table 9.6. Comparing the costs for function evaluations we see that there is not a considerable difference. The approximation of the noise through (9.5) is affordable, as the noise reduction, and so the evaluation of the full function is needed only sporadically. For example for $K_* = 2$ it is needed just four times along all the optimization process, as it is evident from Figure 9.5 (up right) or Figure 9.7 (left).

Finally, in the right plot of Figure 9.7 we compare the decrease in the model $m_k(x_k) - m_k(x_k + p_k^{LM})$ to that of the term $\frac{1}{2}\lambda_k \|p_k^{LM}\|^2$ used to approximate it in the noise control (8.7). As we have claimed in Section 8, the approximation is good, showing that in practice the assumption we made in (8.6) is verified, as we have also proved in Remark 8.7.

Solver	it	$\mathbf{CG_{it}}$	$\mathbf{cost_f}$	$\mathbf{cost_p}$	$\mathbf{\lvert X_{it}\rvert}$	err	$\mathbf{e_{te}}$
LMN_{appr}	38	7.5	15.9	316.7	16000	5.4e-2	0.187
LMN_{est}	37	7.4	17.7	318.1	16000	5.7e-2	0.186

Table 9.6: *Comparison of LMN with noise approximated by (9.5) (first row, LMN_{appr}) and estimated noise (second row, LMN_{est}).*

9.2.2 A real life machine learning problem

In this section we consider a classification problem arising from a real life application, i.e. the parametric design of a family of centrifugal pumps. This problem is described in more details in [J2], here we just give a brief outline, to understand where the classification problem arises from.

9.2.2.1 Parametric design of a family of centrifugal pumps

This problem consists of parameterizing all the pumps of a family in a continuous manner, and of finding among them the one representing the best compromise among all the customers' requirement. The pumps performance is indeed usually measured by several functions, such as aerodynamic efficiency, flow rate. To meet all the customer requirements, it is necessary to accept a compromise between reliability, low-cost manufacturing and high aerodynamic efficiency [59]. The design reduces then to a multi-objective problem.

The design scheme we consider here is described in [20]. That is an extension of the scheme commonly used for the design of a component of a single pump. Usually indeed, the customer is provided with a catalogue of sample pumps, and the one that is closest to his requirements is individuated among them. Then, starting from this baseline configuration the new pump is built, optimizing the parameters describing its geometry.

When a whole family of pumps is considered, the designer provides to the customer the possibility of choosing a pump with characteristics that are intermediate among the ones of the pumps already in the catalogue. Then, a baseline configuration is not available and the design starts from scratch.

For the design, the evaluation of the objective functions of the different pumps is crucial. Nowadays, the exponential increase of computational power allows to do it through CFD (Computational Fluid Dynamics) analysis [89]. Anyway these calculations are computationally expensive, as they are based on the solution of partial differential equations. Even if reliability is still the most important aspect that guides the choice of the final geometry, the competitiveness of the business requires the design process to be as short as possible.

In [20] a method based on a regression meta-model is used to speed up the optimization process. In this approach, the regression meta-model is employed to predict values of the performance functions, to reduce the amount of required CFD computations. These indeed can be restricted just to the amount necessary to build a training set for the meta-model. Thus, the challenge is to perform the lowest number of computations and to obtain the highest accuracy in the approximation of the functions [93, 94].

The procedure is outlined in Framework 1. It is based on coupling CFD computations for solving Reynolds Averaged Navier-Stokes (RANS) equations and feed-forward Artificial Neural Networks as a regression meta-model [56]. It is assumed that the machine performance is evaluated through h scalar performance functions: $f_1, \dots, f_h$ and $\mathbf{f} = [f_1, \dots, f_h]$. We denote a sample by $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$. A sample in this case is a set of n parameters describing the pump geometry. The samples are obtained sampling randomly the design space, namely the space that contains all the parameter combinations that correspond to pumps. It is obtained as the Cartesian product $\mathscr{S}$ of the range of variation of each parameter x_i, cf. (9.10). It is crucial to notice that not all the samples in $\mathscr{S}$ correspond to manufacturable machines, many combinations will result to be non feasible.

The procedure is composed of two parts: Phase 1 of ANN training and Phase 2 of research of an optimal solutions set. In Phase 1, h ANN models, one for each performance function, are trained. They will be used in Phase 2 to build the response surface. In Phase 2 indeed, the meta-models are used to predict performance functions of new pumps samples, to reduce the amount of CFD computations required for the design. A multi-objective algorithm is then run to find the set of optimal geometries among those ones.

Framework 1. Parametric design of a family of turbomachinery components, coupling CFD and ANN.

Phase 1: ANN training

1. **Geometry parameterization**. *Choose n parameters (degrees of freedom) to describe the machine geometry, so that each machine will be identified by a vector $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$.*

2. **Sampling of the design space**. *Taking into account the range of variation of each parameter, the design space is built. Assuming that $x_i^{\min} \le x_i \le x_i^{\max}$ for $i = 1, \dots, n$, the resulting design space is defined as:*

$$\mathscr{S} = [x_1^{\min}, x_1^{\max}] \times \cdots \times [x_n^{\min}, x_n^{\max}] \subseteq \mathbb{R}^n. \tag{9.10}$$

   The space is randomly sampled to generate a dataset $\mathscr{D}_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$.

3. **CFD simulations**. *CFD computations are performed on $\mathscr{D}_0$ to divide the features in the sets $\mathscr{F}'$ of feasible samples and $\mathscr{U}'$ of unfeasible samples. The machine performance functions of feasible samples are evaluated: $\mathbf{f}_j = [f_1(\mathbf{x}_j), \dots, f_h(\mathbf{x}_j)]^T$ for $\mathbf{x}_j \in \mathscr{F}'$ and a performance database $\mathscr{D}_{\mathscr{F}'}$ is built, $\mathscr{D}_{\mathscr{F}'} = \{(\mathbf{x}_j, \mathbf{f}_j),\ \mathbf{x}_j \in \mathscr{F}'\}$.*

4. **ANN training**. *The performance database is used to train the ANN models, that learning from the examples in $\mathscr{D}_{\mathscr{F}'}$, build their own functions $\bar{f}_i$, that are approximations to the true performance functions: $\bar{f}_i \simeq f_i$, $i = 1, \ldots, h$.*

Phase 2: Research of an optimal solutions set

1. **Sampling of the design space.** *The design space is sampled again producing a new dataset $\mathscr{D}_1$.*

2. **ANN execution**. *ANN models are used to predict function $\mathbf{f} = [f_1, \ldots, f_h]$ on all the new samples in $\mathscr{D}_1$ through function $\bar{\mathbf{f}} = [\bar{f}_1, \ldots, \bar{f}_h]$ built at step 4 of Phase 1.*

3. **Multi-objective algorithm**. *A multi-objective algorithm is run to find the set of optimal solutions $\mathscr{D}_{ott}$.*

4. **CFD validation of the optimal solutions set**. *The found solutions set $\mathscr{D}_{ott}$ is validated by CFD computations to discard the unfeasible samples, arising from the sampling at step 1.*

When the optimization procedure starts from a baseline configuration that is geometrically close to the final one, all the tools (e.g. for geometry parameterization, mesh generation, CFD solver etc.) involved in the process are automated and fine tuned for the specific application [59]. Moreover, all the manufacturing and geometrical constraints can be taken a priori into account during the tuning of the tools. As a result, almost all the samples in $\mathscr{S}$ will result to be feasible and all the computations performed can be used to form the training set.

The case in which a parametric design has to be faced is different [79]. Generally, a new design starts from scratch and relies on a quick and flexible design tool, capable of describing in a continuous manner the whole range of geometrical variability of a family of components. The design space investigated to meet the customer requirements becomes really vast and it is impossible to take a priori into account all the manufacturing or geometrical constraints. Then, as we have already anticipated, many of the analyzed geometries will result non feasible from a manufacturing point of view. In the following these geometries will be addressed as *unfeasible*, while all the others will be addressed as *feasible*. As an outcome, most of the CFD computations performed to build the training set are useless. The number of computations necessary to generate a suitable performance database, with enough feasible features, will increase exponentially, and consequently the time needed to perform computations. Also, in Phase 2 the trained meta-model is used to predict the objective functions of new geometries randomly selected. The sampling is performed within the whole design space, and again the most part of the geometries provided as meta-model input will result unfeasible. The predicted performance functions values will then be meaningless and it is possible that step 3 of Phase 2 yields a set of optimal solutions consisting of just unfeasible samples, leading to the need of repeating the procedure again.

Then, to make the approach described in Framework 1 practical, we propose in [J2] to include also the use of a classifier with the aim of discarding the unfeasible parameters combinations in a cheap and fast way.

The classification problem that has to be faced is then that of dividing the samples into the two classes $\mathscr{F}$, of feasible samples, labelled with $+1$, and $\mathscr{U}$ of unfeasible samples labelled with -1. The resulting strategy is outlined in Framework 2, where the added classification procedures are highlighted in italic font.

Framework 2. Hybrid approach: Parametric design of a family of turbomachinery components coupling CFD, ANN, Classifier.

Phase 1: ANN training

1. **Geometry parameterization**. *Choose n parameters (degrees of freedom) to describe the machine geometry, so that each machine will be identified by a vector $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$.*

2. **Sampling of the design space**. *Taking into account the range of variation of each parameter, the design space is built. Assuming that $x_i^{\min} \leq x_i \leq x_i^{\max}$ for $i = 1, \dots, n$, the resulting design space is defined as:*

$$\mathscr{S} = [x_1^{\min}, x_1^{\max}] \times \cdots \times [x_n^{\min}, x_n^{\max}] \subseteq \mathbb{R}^n.$$

The design space is randomly sampled to generate a dataset $\mathscr{D}_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$.

3. ***Classification***. *The samples in $\mathscr{D}_0$ are given in input to the classifier which divides them into the two classes $\mathscr{F}$ (feasible), $\mathscr{U}$ (unfeasible). Features in $\mathscr{U}$ are not taken in further account and just those in $\mathscr{F}$ are considered in the next steps.*

4. **CFD simulations**. *CFD computations are performed on the samples in $\mathscr{F}$. The outliers are eliminated obtaining a subset $\mathscr{F}'$ of just feasible features for which the machine performance functions are evaluated: $\mathbf{f}(\mathbf{x}_j) = [f_1(\mathbf{x}_j), \dots, f_h(\mathbf{x}_j)]$ and $\mathbf{x}_j \in \mathscr{F}'$. The performance database $\mathscr{D}_{\mathscr{F}'}$ is built, which is a set of pairs: $\mathscr{D}_{\mathscr{F}'} = \{(\mathbf{x}_j, \mathbf{f}_j), \mathbf{x}_j \in \mathscr{F}'\}$, where $\mathbf{f}_j = [f_1(\mathbf{x}_j), \dots, f_h(\mathbf{x}_j)]$.*

5. **ANN training**. *The performance database is used to train the ANN models, that learning from the examples in $\mathscr{D}_{\mathscr{F}'}$, build their own functions $\bar{f}_i$ that approximate the true performance functions: $\bar{f}_i \simeq f_i$, $i = 1,\dots,h$.*

Phase 2: Research of an optimal solutions set

1. **Sampling of the design space.** *The design space is sampled again producing a new dataset $\mathscr{D}_1$.*

2. ***Classification***. *Samples in $\mathscr{D}_1$ are given in input to th classifier which divides them into the two classes $\mathscr{F}'', \mathscr{U}''$. Features in $\mathscr{U}''$ are not taken in further account and just those in $\mathscr{F}''$ are considered in the next step.*

3. **ANN execution**. *The ANN model is used to predict function* $\mathbf{f}$ *of the samples in $\mathscr{F}''$.*

4. **Multi-objective algorithm**. *A multi-objective algorithm is run to find the set of optimal solutions $\mathscr{D}_{ott}$.*

5. **CFD validation**. *The optimal solutions set $\mathscr{D}_{ott}$ found is validated trough CFD computations to eliminate possible outliers.*

The two classification procedures inserted at step 3 of Phase 1 and step 2 of Phase 2 are intended to mitigate the drawbacks previously outlined.

Indeed, the first classification allows to restrict the CFD computations performed at step 4 of Phase 1 just to the set $\mathscr{F}$ of features classified as feasible by SVM, with the aim of eliminating outliers. This produces a great saving in CFD computations, as $|\mathscr{F}| \ll |\mathscr{D}_0|$ and CFD performed on samples in $\mathscr{U}$ would be of no use.

The second process allows to predict function values just of samples in $\mathscr{F}''$ that is mainly composed of feasible features. Some outliers will anyway be still present in the set, then step 5 of Phase 2 is still necessary, but it will be much less expensive than the corresponding step 4 of Phase 2 in Framework 1.

9.2.2.2 Classification problem resolution

In [J2] the classification phase is handled by Support Vector Machine, [99]. Three different datasets are considered, characterized by different number of degree of freedom and different ratio between feasible and unfeasible samples. Here, we consider just one dataset as an example. We use a logistic regression and as in Section 9.2.1 we minimize the likelihood function (9.7) by the LMN method. Summing up, in this case a sample is vector $x = (x_1,...,x_n) \in \mathbb{R}^n$ whose components are the degrees of freedom of the pump and the labels are $+1$ if x represents a manufacturable pump (this can be understood as a result of the CFD computations) and -1 otherwise.

We consider a set of data arising from the considered application, with 82000 samples and $n = 40$ features. We divide the set in a training set of $m = 50000$ samples and a testing set of $\tilde{m} = 32000$ samples. We start the optimization process with $K_0 = 100$ and we stop the procedure when $\|\nabla f_{\delta_k}(x_k)\| \leq 10^{-4}$. Parameter κ_d is set to 100.

For the testing dataset the exact labels are known, so we can compute the exact testing error. Let $\mathscr{C} \in \mathbb{R}^m$ be the vector with the correct features classification, i.e. $\mathscr{C}(i) = 1$ if the i-th feature $\mathbf{x}_i \in \mathscr{F}$ and $\mathscr{C}(i) = -1$ if $\mathbf{x}_i \in \mathscr{U}$, $i = 1,...,m$, and $P\mathscr{C} \in \mathbb{R}^m$ the result of the classification process, i.e. $P\mathscr{C}(i)$ is the predicted value of $\mathscr{C}(i)$, $i = 1,...,m$. For each feature four different situations can occur:

- TP=true positive: $\mathscr{C}(i) = 1, P\mathscr{C}(i) = 1$ the feature is feasible and is correctly classified,

- FP=false positive: $\mathscr{C}(i) = -1, P\mathscr{C}(i) = 1$ the feature is unfeasible but is misinterpreted and classified as feasible,

- TN=true negative: $\mathscr{C}(i) = -1, P\mathscr{C}(i) = -1$ the feature is unfeasible and is correctly classified,

- FN= false negative: $\mathscr{C}(i) = 1, P\mathscr{C}(i) = -1$ the feature is feasible but is misinterpreted and classified as unfeasible.

In the considered dataset the number of unfeasible features is greater than the number of feasible ones (the ratio is approximatively 3:1). Then, rather than considering the the fraction of features correctly classified $\frac{TP+TN}{TN+TP+FN+FP}$ as a measure of accuracy, we will consider the following parameters:

- $TPR = \frac{TP}{TP+FN}$ *True Positive Rate* or sensitivity or recall, fraction of positive samples correctly classified over all positive samples available in the test,

- $FPR = \frac{FP}{TN+FP}$ *False Positive Rate*, fraction of unfeasible features misinterpreted over all negative samples available in the test.

- $PPV = \frac{TP}{TP+FP}$ *Positive Predictive Value*, the fraction of true positives in the set features classified as positive,

- $FOR = \frac{FN}{TN+FN}$*False Omission Rate*, the fraction of false negatives in the set features classified as negative.

Parameter TPR tells us how many feasible samples, that are those that interest the designer, we have found. PPV also interests the designer, as it gives a measure of the quality of set $\mathscr{F}$, telling how many unfeasible features have been

	Training					Testing			
	it	$\mathbf{CG_{it}}$	$\mathbf{cost_f}$	$\mathbf{cost_p}$	time (m)	TPR	FPR	PPV	FOR
FLM	9	1.2	10.0	58.0	15	76%	20%	36%	5%
LMN	15	3.4	2.7	22.6	5	77%	23%	34%	4%
SAVE					66%			31%	

Table 9.7: *Parametric design test problem. Left: Comparison of FLM and LMN performance for the training phase of the logistic regression classifier (i.e. minimization of* (9.7)*). Right: results of the execution phase of the classifier on the testing set. Meaning of the labels in explained in the text below. SAVE in the last line highlights the saving in computational time for the training (left) and the saving in CFD computations gained by the use of the classification procedure in the parametric design (right).*

mistaken, and that would lead to useless CFD computations and to possible outliers in the optimal solutions set.

The results of the training and testing processes are reported in Table 9.7. The heading `time(m)` denotes the time required for the optimization process in minutes. From the training phase we can see that as in previous sections the use of LMN compared to FLM provides considerable savings both in function evaluations and in products, which lead to a considerable reduction of the training time ($\simeq 66\%$, last line of Table 9.7, left.).

Regarding the testing phase, we can notice that a high percentage ($76\% - 77\%$) of the feasible samples has been correctly classified during the process. This result is important because it influences the savings in building the ANN training set, provided by the procedure in Framework 2, compared to that in Framework 1.

In this case we obtain a saving of 31%, i.e. with the procedure in Framework 2 it is possible to obtain the same training set as employing the procedure in Framework 1, but with 31% CFD computations less (last line of Table 9.7, right). In [J2] even better results are obtained by the use of Support Vector Machine, obtaining $TPR = 78\%$ and $PPV = 50.8\%$ for this dataset, thanks to the possibility of performing a nonlinear classification and of a careful setting of the method's free parameters.

9.3 Chapter conclusion

In this chapter, we have discussed the numerical behaviour of the procedure presented in Chapter 8. It was tested on problems arising in data assimilation and machine learning, one of which arises from a real life application, in the domain of turbomachinery parametric design.

The results show that the implemented procedure (LMN) is able to find a first-order solution of the unperturbed problem of comparable accuracy with respect to that found by other inexact Levenberg-Marquardt methods employing the exact objective function. The LMN approach allows for significant savings, both in function evaluations and in matrix-vector products, thanks to the proposed strategy to control the noise.

The provided numerical results also show that, even though the step is shown to asymptotically converge to a steepest-descent step, overall the procedure benefits from the use of a genuine Levenberg-Marquardt method till the last stage of convergence, gaining a faster convergence rate compared to a pure steepest descent method. Moreover, this is achieved at a modest cost, thanks to the use of iterative methods to solve the arising linear systems. Indeed, a very rough accuracy is imposed, that leads to a low number of linear iterations that is however still enough to achieve convergence.

Acknowledgements We thank the authors of [46] for providing us the Matlab code for the data assimilation test problem.

Conclusion

Main contributions

In this thesis, we have investigated the numerical resolution of particular classes of noisy least squares problems. We designed novel regularizing Levenberg-Marquardt methods that are suitable to handle problems with noisy function and gradient. We have devised specialized updates of the regularization parameters that ensure that the generated sequences approach a solution of the unperturbed original problem. We have then analyzed the theoretical properties of these methods under mild assumptions, such as local and global convergence, regularizing properties, and complexity. We also validated their numerical behaviour in practice, using test problems arising from different fields, such as data assimilation, machine learning, geophysics, and hydrology.

Our attention has been devoted to two classes of problems. We first considered ill-posed problems, and then large scale nonlinear least squares problems with expensive objective function, that can be computed with different levels of accuracy. We considered noisy problems in both classes, but of different nature. For ill-posed problems, the noise is limited to the data and arises from measurements errors, while in the second class it is identified with the accuracy chosen in the function approximations and is thus a dynamically adjusted quantity.

In the field of ill-posed problems, our contribution is twofold. We first considered existing Levenberg-Marquardt procedures for zero-residual problems. We discussed how to implement them in a reliable way and we proposed a more robust improvement of the existing methods. Then, we considered the class of nonzero residual problems. Even though many problems belonging to this class are encountered in numerous applications, whenever the model does not fit the data, they have not been the object of much study in the literature, which is traditionally more focused on the zero residual case. Such problems are usually solved with methods for zero residual problems, incorporating the modelling errors to the data errors. Believing however that methods specially designed for this class of problems are worth to be studied, we proposed a first attempt in this direction. This represents an original contribution in the field of ill-posed inverse problems.

Both methods are characterized by novel Trust-Region radius choices, that provide an automatic setting of the regularization parameters, ensuring a regularizing behaviour of the approaches. This is a highly desirable property for a regularizing method, since an a priori choice is often difficult. We analyzed the local convergence of the proposed methods under mild assumptions, different from those usually used in the literature for Levenberg-Marquardt methods. In our numerical experiments, we showed that the method for zero residual problems is more robust than existing Levenberg-Marquardt methods in the literature, and is less dependent on the choice of free parameters. Both Trust-Region methods are shown to outperform standard trust region methods for the solution of ill-posed problems.

Concerning the the second field of methods, we designed an inexact Levenberg-Marquardt approach that relies on a sequence of problems with noisy functions and gradients approximating the original problem. The method manages to produce a sequence converging to a solution of the unperturbed problem thanks to two key ingredients: a rule to measure the noise level that can be allowed in the objective function while successfully continuing with the optimization process, and a suitable update of the regularization parameters to handle the noise in the function. In our numerical experiments, we showed the method to be cheaper than inexact Levenberg-Marquardt methods employing exact function and gradient, achieving significant savings in function evaluations and matrix-vector products. We also analyzed its performance on a real life machine learning problem arising in the design of turbomachinery components. This method constitutes an improvement of the state-of-the-art in the solution of problems with dynamic noise. Indeed, while unconstrained minimization problems with noisy functions, gradients, and Hessians with varying accuracy have been the object of an intensive study in the last years, we are not aware of other methods designed to solve the considered class of least squares problems.

Perspectives

We briefly discuss possible extensions of the work presented in this thesis that could be the object of further research.

As a first improvement, we could design a variant of the approach presented in Part III specially designed to solve problems with objective functions given by sums of squares over a really large number of terms. For such problems it is not possible to evaluate the exact function, but it possible to devise estimates of the error when the function approximations are built considering only some terms in the sum. However, such estimates are usually probabilistic in nature. Then, a stochastic approach would be more suitable in this context.

A second perspective is to consider the solution of large scale ill-posed problems. This could be tackled in two different ways.

First, we could design of a variant of the elliptical Trust-Region approach presented in Chapter 5, suitable for handling large scale problems. The critical point here is that the method needs the square root of matrix B_k. The action of the square root of B_k on a vector can be approximated by an iterative solver, but this introduces a source of inexactness and consequently the theory should be modified in order to take this into account. Moreover, the linear systems arising in the computation of the step should be approximately solved by employing an iterative solver.

A second possibility is the extension of the method presented in Part III to allow input spaces of increasing dimensions, to include also multilevel strategies. The ideas on which the methods presented in Part II and Part III are based could be coupled to design a method suitable for handling discrete ill-posed problems arising from a discretization of the input space of an infinite dimensional problem, such as parameter identification problems or Fredholm equations. For such problems a mesh size has to be fixed, that affects the ill-posedness, as this worsen with the size of the discretization. A solution method for such problems could start the solution process from a coarse grid and then rely on our noise control strategy to decide when it is necessary to move to a finer grid, so that it is not necessary to make an a-priori decision on the size of the grid. The idea would then be to end up with a grid fine enough to have a good approximation (related to the noise on the data) of the solution of the underlying infinite dimensional problem. This should be coupled with a regularization method to handle the ill-posedness on finer grids, possibly able to handle also large scale problems in case really fine discretizations are needed.

All in all, we hope that the methods designed, developed, and analyzed in this thesis can pave the way towards a broader class of methods aiming at the numerical resolution of large scale, noisy, ill-posed nonlinear least squares problems.

Appendix

Here we report the proofs of some of the results in Chapter 5.

Lemma 5.15 Suppose that Assumptions 5.7 and 5.13 hold. Then, Algorithm 5.1 generates a sequence $\{x_k^\delta\}$ such that, for $\bar{k} \le k < k_*(\delta)$,

(i) $\lambda_k > 0$ and x_k^δ belongs to $B_{2r}(x_{\bar{k}}^\delta) \cap B_r(x^\dagger)$;

(ii) $\|x_{k+1}^\delta - x^\dagger\| < \|x_k^\delta - x^\dagger\|$, $\theta_{k+1} > \theta_k$;

(iii) there exists a constant $\bar{\lambda} > 0$ such that $\lambda_k \le \bar{\lambda}$.

Proof. (i)-(ii) From the choice of the trust-region radius in Algorithm 5.1, (5.5) holds and $\lambda_k \ge \lambda_k^q > 0$. From Lemma 5.14 and Lemma 5.6, condition (5.11) is satisfied for all $\bar{k} \le k < k_*(\delta)$, and this implies $\|x_{k+1}^\delta - x^\dagger\| < \|x_k^\delta - x^\dagger\|$, x_k^δ belongs to $B_{2r}(x_k^\delta) \cap B_r(x^\dagger)$, and $\theta_{k+1} > \theta_k$ for all $\bar{k} \le k < k_*(\delta)$.

(iii) Since $\lambda_k > 0$, the trust-region is active and from (5.17a) we have that

$$\Delta_k = \|z(\lambda_k)\| = \|(B_k^2 + \lambda_k I)^{-1} B_k^{1/2} g_k\| \le \frac{\|B_k^{1/2} g_k\|}{\lambda_k}.$$

Then, if Δ_k chosen at step 1 of Algorithm 5.1 guarantees condition $\rho_k(p_k) \ge \eta$, the thesis follows as

$$\lambda_k \le \frac{\|B_k^{1/2} g_k\|}{\Delta_k} \le \frac{1}{C_{\min}}. \tag{A.1}$$

Otherwise, the trust-region radius is progressively reduced, and we provide a bound for the value of Δ_k at termination of step 2 of Algorithm 5.1. Let us consider the case $f_\delta(x_k^\delta + p_k) > \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2$. Trivially,

$$1 - \rho_k(p_k) = \frac{f_\delta(x_k^\delta + p_k) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2}{f_\delta(x_k^\delta) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2},$$

and

$$\begin{aligned} f_\delta(x_k^\delta + p_k) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2 \quad \le \quad & \frac{1}{2}\|F(x_k^\delta + p_k) - F(x_k^\delta) - J(x_k^\delta)p_k\|^2 \\ & + \|F(x_k^\delta + p_k) - F(x_k^\delta) - J(x_k^\delta)p_k\| \\ & \|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\| \end{aligned}$$

By the Lipschitz continuity of J it holds

$$\|F(x_k^\delta + p_k) - F(x_k^\delta) - J(x_k^\delta)p_k\| \leq \frac{L}{2}\|p_k\|^2.$$

Moreover, using (5.21)

$$\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p(\lambda)\| < \|F(x_k^\delta) - y^\delta\|$$

for any $\lambda \geq 0$. Consequently, as $\|p_k\| \leq \|B_k^{1/2}\|\Delta_k$ and $\Delta_k \leq C_{\max}\|B_k^{1/2}g_k\|$,

$$\begin{aligned} f_\delta(x_k^\delta + p_k) - &\frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2 \leq \\ &\frac{L}{2}K_J^2\Delta_k^2\|F(x_0) - y\|\left(\frac{L}{4}K_J^6C_{\max}^2\|F(x_0) - y\| + 1\right). \end{aligned}$$

Theorem 6.3.1 in [22] shows that

$$f_\delta(x_k^\delta) - \left(\frac{1}{2}z_k^TB_k^2z_k + (B_k^{1/2}g_k)^Tz_k + f_\delta(x_k^\delta)\right) \geq \frac{1}{2}\|B_k^{1/2}g_k\|\min\left\{\Delta_k, \frac{\|B_k^{1/2}g_k\|}{\|B_k^2\|}\right\}.$$

Then, (5.18) yields

$$f_\delta(x_k^\delta) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2 \geq \frac{1}{2}\Delta_k\|B_k^{1/2}g_k\|,$$

whenever $\Delta_k \leq \dfrac{\|B_k^{1/2}g_k\|}{K_J^4}$ and this implies

$$1 - \rho_k(p_k) \leq \frac{LK_J^2\Delta_k\|F(x_0) - y\|(\frac{1}{4}LK_J^6C_{\max}^2\|F(x_0) - y\| + 1)}{\|B_k^{1/2}g_k\|}.$$

Namely, termination of the repeat loop occurs with

$$\Delta_k \leq \omega\|B_k^{1/2}g_k\|,$$

and

$$\omega = \min\left\{\frac{1}{K_J^4}, \frac{1-\eta}{LK_J^2\|F(x_0) - y\|(\frac{1}{4}LK_J^6C_{\max}^2\|F(x_0) - y\| + 1)}\right\}.$$

Taking into account step 1 and the updating rule at step 2.4, we can conclude that, at termination of step 2, the trust-region radius Δ_k satisfies

$$\Delta_k \geq \min\{C_{\min}, \gamma\omega\}\|B_k^{1/2}g_k\|.$$

In fact, in case for a smaller value of Δ_k it happens $f_\delta(x_k^\delta + p_k) \leq \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2$, then $\rho_k(p_k) \geq 1 > \eta$ and the loop at step 2 terminates. Then, it terminates for a trust-region radius greater than or equal to the one estimated above. Then,

$$\lambda_k \leq \frac{\|B_k^{1/2}g_k\|}{\Delta_k} \leq \max\left\{\frac{1}{\gamma\omega}, \frac{1}{C_{\min}}\right\} = \bar{\lambda}.$$

□

Theorem 5.16 Suppose that Assumptions 5.7 and 5.13 hold. Then,

(i) The iterates generated by Algorithm 5.1 satisfy the stopping criterion (5.2) after a finite number $k_*(\delta)$ of iterations.

(ii) Suppose further that the sequence $\{x_k\}$ generated with the exact data y satisfies $\rho_k(x_{k+1}-x_k) \neq \eta$, for all k. Then the sequence $\{x^\delta_{k_*(\delta)}\}$ converges to a point belonging to $\mathscr{S} \cap \bar{B}_r(x^\dagger)$, where $\mathscr{S}$ is defined in (5.32), whenever δ tends to zero.

Proof. (i) Summing up from $\bar{k}$ to $k_*(\delta)-1$, by (5.2), (5.5), (5.11) and Lemma 5.15 it follows

$$(k_*(\delta)-\bar{k})\tau^2\delta^2 \leq \sum_{k=\bar{k}}^{k_*(\delta)-1} \|J(x^\delta_k)^T(F(x^\delta_k))\|^2 \leq \frac{\theta_{\bar{k}}\bar{\lambda}}{2(\theta_{\bar{k}}-1)q^2}\|x^\delta_{\bar{k}}-x^\dagger\|^2.$$

Thus, $k_*(\delta)$ is finite for $\delta > 0$.

(ii) Convergence of $x^\delta_{k_*(\delta)}$ to a solution of (3.1) as δ tends to zero is obtained by adapting the proof of Theorem 4.19. Specifically, let x^* be the limit of the sequence $\{x_k\}$ corresponding to the exact data y and let $\{\delta_n\}$ be a sequence of values of δ converging to zero as $n \to \infty$. Denote by y^{δ_n} a corresponding sequence of perturbed data, and by $k_n = k_*(\delta_n)$ the stopping index determined from the discrepancy principle (5.2) applied with $\delta = \delta_n$. Assume first that $\tilde{k}$ is a finite accumulation point of $\{k_n\}$. Without loss of generality, we can assume that $k_n = \tilde{k}$ for all $n \in \mathbb{N}$. Thus, from the definition of k_n it follows that

$$\|J(x^{\delta_n}_{\tilde{k}})^T(y^{\delta_n}-F(x^{\delta_n}_{\tilde{k}}))\| \leq \tau\delta_n. \tag{A.2}$$

As, by assumption, $\rho_k(x_{k+1}-x_k) \neq \eta$, for all k, it follows that for the fixed index $\tilde{k}$, the iterate $x^\delta_{\tilde{k}}$ depends continuously on δ. Then

$$x^{\delta_n}_{\tilde{k}} \to x_{\tilde{k}}, \qquad J(x^{\delta_n}_{\tilde{k}}) \to J(x_{\tilde{k}}), \qquad F(x^{\delta_n}_{\tilde{k}}) \to F(x_{\tilde{k}}) \qquad \text{as } \delta_n \to 0.$$

Therefore, by (A.2), it follows that $J(x_{\tilde{k}})^T(y-F(x_{\tilde{k}}))=0$, and the $\tilde{k}$-th iterate with exact data y is a solution of (3.1), i.e. $x^* = x_{\tilde{k}}$, and we can conclude that $x^{\delta_n}_{k_n} \to x^*$ as $\delta_n \to 0$.

It remains to consider the case where $k_n \to \infty$ as $n \to \infty$. As $\{x_k\}$ converges to a solution x^* of (3.1) by Theorem 5.12, there exists $\tilde{k} > 0$ such that

$$\|x_k - x^*\| \leq \frac{1}{2}\bar{r} \qquad \text{for all} \qquad k \geq \tilde{k},$$

where $\bar{r} < \min\left\{\dfrac{(q-\sigma)\tau - K_J(\sigma+1)}{c(K_J+\tau)}, r\right\}$. Then, as $x^\delta_{\tilde{k}}$ depends continuously on δ, δ_n tends to zero and $k_*(\delta_n) \to \infty$, there exists δ_n sufficiently small such that $\tilde{k} \leq k_*(\delta_n)$ and

$$\|x^{\delta_n}_{\tilde{k}} - x_{\tilde{k}}\| \leq \frac{1}{2}\bar{r}.$$

Then, for δ_n sufficiently small

$$\|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \|x_{\tilde{k}}^{\delta_n} - x_{\tilde{k}}\| + \|x_{\tilde{k}} - x^*\| \leq \bar{r}. \tag{A.3}$$

Now, from item (i) of Lemma 5.15, it holds $x_{\tilde{k}}^{\delta_n} \in B_{2r}(x_{\tilde{k}}^{\delta_n})$, while from (5.36) and Theorem 5.12 it holds $x^* \in B_{2r}(x_{\tilde{k}}^{\delta_n})$ as

$$\|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \|x_{\tilde{k}}^{\delta_n} - x^\dagger\| + \|x^\dagger - x^*\| \leq 2r.$$

Letting $e_k^* = x^* - x_k^{\delta_n}$, repeating arguments in Lemma 5.14 and using (5.6), (3.2) we get

$$\begin{aligned}
\|M_{\tilde{k}}^g(e_{\tilde{k}}^*)\| &\leq K_J\delta_n + \|J(x_{\tilde{k}}^{\delta_n})^T(y - F(x_{\tilde{k}}^{\delta_n}) + J(x_{\tilde{k}}^{\delta_n})(x^* - x_{\tilde{k}}^{\delta_n}))\| \\
&\leq K_J\delta_n + (c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma)\|J(x_{\tilde{k}}^{\delta_n})^T(y - F(x_{\tilde{k}}^{\delta_n}))\| \\
&\leq (1 + c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma)K_J\delta_n + (c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma)\|J(x_{\tilde{k}}^{\delta_n})^T(y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n}))\|.
\end{aligned}$$

Then, at iteration $\tilde{k}$, conditions (5.2) and (5.5) give

$$\begin{aligned}
\|M_{\tilde{k}}^g(e_{\tilde{k}}^*)\| &\leq \left(K_J\frac{1 + c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma}{\tau} + (c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma)\right)\|J(x_{\tilde{k}}^{\delta_n})^T(y^{\delta_n} - F(x_{\tilde{k}}^{\delta_n})\| \\
&\leq \left(K_J\frac{1 + c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma}{q\tau} + \frac{c\|x^* - x_{\tilde{k}}^{\delta_n}\| + \sigma}{q}\right)\|M_{\tilde{k}}^g(p_{\tilde{k}})\|.
\end{aligned}$$

Thus, by (A.3) and $\bar{r} < \min\left\{\frac{(q-\sigma)\tau - K_J(\sigma+1)}{c(K_J+\tau)}, r\right\}$, it follows that

$$\|M_{\tilde{k}}^g(e_{\tilde{k}}^*)\| \leq \frac{1}{\theta_{\tilde{k}}}\|M_{\tilde{k}}^g(p_{\tilde{k}})\|$$

is satisfied with $\theta_{\tilde{k}} = \dfrac{q\tau}{1 + c(1+\tau)\bar{r} + \sigma(1+\tau)} > 1$. Replacing $x^\dagger$ with x^*, (5.11) gives $\|x_{\tilde{k}+1}^{\delta_n} - x^*\| < \|x_{\tilde{k}}^{\delta_n} - x^*\|$ and repeating the above arguments, by induction we obtain monotonicity of the error $\|x_k^{\delta_n} - x^*\|$ for $\tilde{k} \leq k \leq k_n$. Then

$$\|x_{k_n}^{\delta_n} - x^*\| < \|x_{\tilde{k}}^{\delta_n} - x^*\| \leq \bar{r}.$$

Finally, since the previous arguments can be repeated for any positive $\epsilon \leq \bar{r}$, provided that δ_n is small enough, we obtain that

$$x_{k_n}^{\delta_n} \to x^* \qquad \text{as} \qquad \delta_n \to 0.$$

□

Publications related to the thesis

Some of the work presented in this thesis has been the object of communications to the scientific community, as reported below.

Submitted articles

[S1] S. Bellavia, S. Gratton, and E. Riccietti. "A Levenberg-Marquardt method for large nonlinear least squares problems with dynamic accuracy in functions and gradients". In: *under revision in Numerische Mathematik* (2017).

[S2] S. Bellavia and E. Riccietti. "On an elliptic trust-region procedure for ill-posed nonlinear least squares problems". In: *submitted* (2017).

Journal articles

[J1] S. Bellavia, B. Morini, and E. Riccietti. "On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation". In: *Computational Optimization and Applications* 64.1 (2016), pp. 1–30.

[J2] E. Riccietti, J. Bellucci, M. Checcucci, M. Marconcini, and A. Arnone. "Support Vector Machine classification applied to the parametric design of centrifugal pumps". In: *Engineering Optimization* (2017).

Conference talks

[C1] S. Bellavia, S. Gratton, and E. Riccietti. "A Levenberg-Marquardt method for large scale noisy nonlinear least squares problems". In: *SIAM Conference on Optimization (SIOPT'17)*. Vancouver, Canada, May 2017.

[C2] S. Bellavia, S. Gratton, and E. Riccietti. "Levenberg-Marquardt method for ill-posed large scale nonlinear least squares problems". In: *OIP2016– Optimization Techniques for Inverse Problems III*. Modena, Italy, Sept. 2016.

[C3] S. Bellavia, B. Morini, and E. Riccietti. “A regularization trust-region approach for ill-posed nonlinear systems”. In: *CIMI HPC semester: workshop on optimization and data assimilation*. Toulouse, France, Jan. 2016.

[C4] S. Bellavia, B. Morini, and E. Riccietti. “On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation”. In: *Networking in Numerical Analysis 2015*. Bertinoro, Italy, Nov. 2015.

[C5] S. Bellavia, B. Morini, and E. Riccietti. “On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation”. In: *XX Congresso UMI*. Siena, Italy, Sept. 2015.

[C6] S. Bellavia, B. Morini, and E. Riccietti. “Regularizing trust-region approaches for ill-posed nonlinear systems and nonlinear least squares”. In: *20th Conference of the International Linear Algebra Society (ILAS’16)*. Leuven, Belgium, July 2016.

[C7] S. Bellavia, B. Morini, and E. Riccietti. “Solving ill-posed nonlinear systems with noisy data: a regularizing trust-region approach”. In: *PING – Inverse Problems in Geophysics*. Firenze, Italy, Apr. 2016.

[C8] E. Riccietti, J. Bellucci, M. Checcucci, M. Marconcini, and A. Arnone. “Parametric design of a family of centrifugal pumps: dealing with unbalancedness of geometries dataset”. In: *Summer School on Optimization, Big Data and Applications (OBA)*. Veroli, Italy, July 2017.

[C9] E. Riccietti, J. Bellucci, M. Checcucci, M. Marconcini, and A. Arnone. “Support Vector Machine classification applied to the parametric design of centrifugal pumps”. In: *Congresso Nazionale SIMAI2016*. Milano, Italy, Sept. 2016.

References

[1] G. Allaire. *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*. Oxford University Press, 2007.

[2] T. Anderson and D. Darling. “A test of goodness of fit”. In: *Journal of the American Statistical Association* 49.268 (1954), pp. 765–769.

[3] B. Andreas, H. Engl, A. Neubauer, O. Scherzer, and C. Groetsch. “Weakly closed nonlinear operators and parameter identification in parabolic equations by Tikhonov regularization”. In: *Applicable Analysis* 55.3-4 (1994), pp. 215–234.

[4] D. Arnold. *Noisy optimization with Evolution Strategies*. Springer Science & Business Media, 2002.

[5] A. Bandeira, K. Scheinberg, and L. Vicente. “Convergence of trust-region methods based on probabilistic models”. In: *SIAM Journal on Optimization* 24.3 (2014), pp. 1238–1264.

[6] H. Banks and K. Murphy. “Estimation of coefficients and boundary parameters in hyperbolic systems”. In: *SIAM Journal on Control and Optimization* 24.5 (1986), pp. 926–950.

[7] R. Behling. “The method and the trajectory of Levenberg-Marquardt”. PhD thesis. Instituto Nacional de Matematica Pura e Aplicada (IMPA), Mar. 2011.

[8] R. Behling and A. Fischer. “A unified local convergence analysis of inexact constrained Levenberg-Marquardt methods”. In: *Optimization Letters* 6 (2012), pp. 927–940.

[9] S. Bellavia, C. Cartis, N. Gould, B. Morini, and P. Toint. “Convergence of a regularized Euclidean residual algorithm for nonlinear least-squares”. In: *SIAM Journal on Numerical Analysis* 48.1 (2010), pp. 1–29.

[10] E. Bergou, Y. Diouane, and V. Kungurtsev. “Global and local convergence of a Levenberg-Marquadt algorithm for inverse problems”. In: *Technical Report ISAE-SUPAERO* (2017).

[11] E. Bergou, S. Gratton, and L. Vicente. “Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to Data Assimilation”. In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 924–951.

[12] E. Birgin, N. Krejić, and J. Martínez. “On the employment of inexact restoration for the minimization of functions whose evaluation is subject to errors”. In: *Mathematics of Computation* (2017).

[13] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. “Convergence rate analysis of a stochastic trust region method for nonconvex optimization”. In: *arXiv preprint arXiv:1609.07428* (2016).

[14] E. Blayo, E. Cosme, M. Nodet, and A. Vidard. *Introduction to Data Assimilation*. 2011.

[15] R. Bollapragada, R. Byrd, and J. Nocedal. “Exact and inexact subsampled Newton methods for optimization”. In: *arXiv preprint arXiv:1609.08502* (2016).

[16] L. Bottou, F. Curtis, and J. Nocedal. “Optimization methods for large-scale machine learning”. In: *arXiv preprint arXiv:1606.04838* (2016).

[17] A. Buccini. “Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term”. In: *Applied Numerical Mathematics* (2016).

[18] C. Cartis, N. Gould, and P. Toint. “Trust-region and other regularizations of linear least-squares problems”. In: *BIT Numerical Mathematics* 49 (2009), pp. 21–53.

[19] *Causality workbench team. A marketing dataset.* http://www.causality.inf.ethz.ch/data/CINA.html. 2008.

[20] M. Checcucci, A. Schneider, M. Marconcini, F. Rubechini, A. Arnone, L. De Franco, and M. Coneri. “A novel approach to parametric design of centrifugal pumps for a wide range of specific speeds”. In: *Proceedings of 12th International Symposium on Experimental and Computational Aerothermodynamics of Internal Flows, 13-16 July 2015, Lerici (SP), Italy, ISAIF 12 paper nr.121*. 2015.

[21] F. Colonius and K. Kunisch. “Stability for parameter estimation in two point boundary value problems”. In: *Journal für die Reine und Angewandte Mathematik* (1984).

[22] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. SIAM, 2000.

[23] A. Conn, K. Scheinberg, and L. Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[24] H. Dan, N. Yamashita, and M. Fukushima. “Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions”. In: *Optimization Methods and Software* 17.4 (2002), pp. 605–626.

[25] G. Deidda, C. Fenu, and G. Rodriguez. “Regularized solution of a nonlinear problem in electromagnetic sounding”. In: *Inverse Problems* 30.12 (2014), p. 125014.

[26] R. Dembo, S. Eisenstat, and T. Steihaug. “Inexact Newton methods”. In: *SIAM Journal of Numerical Analysis* 19 (1982), pp. 400–408.

[27] J. Dennis Jr. and R. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Vol. 16. SIAM, 1996.

[28] M. Donatelli and M. Hanke. “Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring”. In: *Inverse Problems* 29.9 (2013), p. 095008.

[29] M. Donatelli, A. Neuman, and L. Reichel. “Square regularization matrices for large linear discrete ill-posed problems”. In: *Numerical Linear Algebra with Applications* 19.6 (2012), pp. 896–913.

[30] A. Ebtehaj, M. Zupanski, G. Lerman, and E. Foufoula-Georgiou. “Variational data assimilation via sparse regularisation”. In: *Tellus A: Dynamic Meteorology and Oceanography* 66.1 (2014), p. 21789.

[31] H. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*. Vol. 375. Kluwer Academic Publishers Group, 1996.

[32] F. Facchinei, A. Fischer, and M. Herrich. “A family of Newton methods for nonsmooth constrained systems with nonisolated solutions”. In: *Mathematical Methods of Operations Research* 77.3 (2013), pp. 433–443.

[33] J. Fan. “A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations”. In: *Journal of Computational Mathematics* (2003), pp. 625–636.

[34] J. Fan. “Convergence rate of the trust-region method for nonlinear equations under local error bound condition”. In: *Computational Optimization and Applications* 34 (2005), pp. 215–227.

[35] J. Fan and J. Pan. “Inexact Levenberg-Marquardt method for nonlinear equations”. In: *Discrete and Continuous Dynamical Systems - Series B* 4.4 (2004), pp. 1223–1232.

[36] J. Fan and J. Pan. “A modified trust-region algorithm for nonlinear equations with new updating rule of trust-region radius”. In: *International Journal of Computer Mathematics* 87 (2010), pp. 3186–3195.

[37] J. Fan and J. Pan. “A note on the Levenberg–Marquardt parameter”. In: *Applied Mathematics Computation* 207 (2009), pp. 351–359.

[38] J. Fan and J. Pan. “An improved trust-region algorithm for nonlinear equations”. In: *Computational Optimization and Applications* 48 (2011), pp. 59–70.

[39] J. Fan and Y. Yuan. “A new trust region algorithm with trust region radius converging to zero”. In: *Proceeding of the 5th International Conference on Optimization: Techiniques and Applications*. ICOTA, Hong Kong. 2001, pp. 786–794.

[40] J. Fan and Y. Yuan. “On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption”. In: *Computing* 74.1 (2005), pp. 23–39.

[41] A. Fischer, P. Shukla, and M. Wang. “On the inexactness level of robust Levenberg–Marquardt methods”. In: *Optimization* 59 (2010), pp. 273–287.

[42] A. Fischer, M. Herrich, A. F. Izmailov, and M. V. Solodov. “Convergence conditions for Newton-type methods applied to complementarity systems with nonisolated solutions”. In: *Computational Optimization and Applications* 63.2 (Mar. 2016), pp. 425–459.

[43] M. Friedlander and M. Schmidt. “Hybrid deterministic-stochastic methods for data fitting”. In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1380–A1405.

[44] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins University, 1996.

[45] S. Gratton, S. Gürol, and P. Toint. “Preconditioning and globalizing Conjugate Gradients in dual space for quadratically penalized nonlinear-least squares problems”. In: *Computational Optimization and Applications* 54.1 (2013), pp. 1–25.

[46] S. Gratton, M. Rincon-Camacho, E. Simon, and P. Toint. “Observation thinning in Data Assimilation computations”. In: *EURO Journal on Computational Optimization* 3.1 (2015), pp. 31–51.

[47] C. W. Groetsch and A. Neubauer. “Regularization of ill- posed problems: optimal parameter choice in finite dimensions”. In: *Journal of Approximation Theory* 58 (1989), pp. 184–200.

[48] C. Groetsch. *The theory of Tikhonov regularization for Fredholm equations*. Boston Pitman Publication, 1984.

[49] W. Hager. “Stabilized quadratic programming”. In: *Computational Optimization and Applications* 12 (1999), pp. 253–273.

[50] M. Hanke. “A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems”. In: *Inverse problems* 13.1 (1997), p. 79.

[51] M. Hanke. *Conjugate Gradient type methods for ill-posed problems*. Vol. 327. CRC Press, 1995.

[52] M. Hanke. “The regularizing Levenberg-Marquardt scheme is of optimal order”. In: *Journal of Integral Equations Applications* 22 (2010), pp. 259–283.

[53] M. Hanke and C. Groetsch. “Nonstationary iterated Tikhonov Regularization”. In: *Journal of Optimization Theory and Applications* 98.1 (1998), pp. 37–53.

[54] M. Hanke, A. Neubauer, and O. Scherzer. “A convergence analysis of the Landweber iteration for nonlinear ill-posed problems”. In: *Numerische Mathematik* 72.1 (1995), pp. 21–37.

[55] P. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

[56] S. Haykin. *Neural Networks: A Comprehensive Foundation. 2nd edition*. Macmillan, New York, 1998.

[57] J. Hendrickx, B. Borchers, D. Corwin, S. Lesch, A. Hilgendorf, and J. Schlue. “Inversion of soil conductivity profiles from electromagnetic induction measurements”. In: *Soil Science Society of America Journal* 66.3 (2002), pp. 673–685.

[58] P. Henrici. “Elements of numerical analysis”. In: *J. Wiley and Sons, Chicester and New York* (1964).

[59] P. Hergt. “Pump research and development: past, present, and future”. In: *Journal of Fluids Engineering* 121.2 (1999), pp. 248–253.

[60] M. Hestenes and E. Stiefel. “Methods of Conjugate Gradients for solving linear systems”. In: *Journal of Research of the National Bureau of Standards* 49.1 (1952).

[61] B. Hofmann. “On the degree of ill-posedness for nonlinear problems”. In: *Journal of Inverse and Ill-Posed Problems* 2.1 (1994), pp. 61–76.

[62] B. Hofmann and O. Scherzer. “Factors influencing the ill-posedness of nonlinear problems”. In: *Inverse Problems* 10.6 (1994), p. 1277.

[63] I. Ipsen, C. Kelley, and S. R. Pope. “Rank-deficient nonlinear least squares problems and subset selection”. In: *SIAM Journal on Numerical Analysis* 49.3 (2011), pp. 1244–1266.

[64] Q. Jin and M. Zhong. “Nonstationary iterated Tikhonov regularization in Banach spaces with uniformly convex penalty terms”. In: *Numerische Mathematik* 127.3 (2014), pp. 485–513.

[65] B. Kaltenbacher. “A Projection-Regularized Newton method for nonlinear ill-posed problems and its application to parameter identification problems with finite element discretization”. In: *SIAM Journal on Numerical Analysis* 37.6 (2000), pp. 1885–1908.

[66] B. Kaltenbacher. “Toward global convergence for strongly nonlinear ill-posed problems via a regularizing multilevel approach”. In: *Numerical Functional Analysis and Optimization* 27.5-6 (2006), pp. 637–665.

[67] B. Kaltenbacher, A. Neubauer, and O. Scherzer. *Iterative regularization methods for nonlinear ill-posed problems*. Vol. 6. Walter de Gruyter, 2008.

[68] C. Kanzow, N. Yamashita, and M. Fukushima. “Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints”. In: *Journal of Computational and Applied Mathematics* 172 (2004), pp. 375–397.

[69] C. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

[70] C. Kelley. *Iterative methods for optimization*. SIAM, 1999.

[71] C. Kelley. *Iterative methods for optimization: Matlab codes*.

[72] J. King and A. Neubauer. “A Variant of Finite-Dimensional Tikhonov Regularization with A-Posteriori Parameter Choice”. In: *Computing* 40 (1988), pp. 91–109.

[73] A. Kravets, M. Shcherbakov, M. Kultsova, and O. Shabalina. “Comparative Analysis of the Numerical Measures for Mining Associative and Causal Relationships in Big Data”. In: *Creativity in Intelligent Technologies and Data Science: First Conference, CIT&DS 2015, Volgograd, Russia, September 15-17, 2015. Proceedings*. Cham: Springer International Publishing, 2015, pp. 571–582.

[74] N. Krejić and J. Martínez. “Inexact Restoration approach for minimization with inexact evaluation of the objective function”. In: *Mathematics of Computation* 85.300 (2016), pp. 1775–1791.

[75] K. Kunisch and L. White. “Parameter estimation, regularity and the penalty method for a class of two point boundary value problems”. In: *SIAM Journal of Control and Optimization* 25 (1 1987).

[76] K. Levenberg. “A method for the solution of certain nonlinear problems in least-squares”. In: *Quarterly Applied Mathematics* 2 (1944), pp. 164–168.

[77] P. Linz and R. Wang. *Exploring Numerical Methods: An Introduction To Scientific Computing Using MATLAB*. Jones & Bartlett Learning, 2002.

[78] D. Marquardt. “An algorithm for least-squares estimation of nonlinear parameters”. In: *SIAM Journal Applied Mathematics* 11 (1963), pp. 431–441.

[79] J. Monedero. “Parametric design: a review and some experiences”. In: *Automation in Construction* 9.4 (2000), pp. 369–377.

[80] J. Moré. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis (Watson, ed). Springer Lecture Notes in Mathematics 630, Berlin*. 1978, pp. 105–116.

[81] V. Morozov. “On the solution of functional equations by the method of regularization”. In: *Soviet Mathematics Doklady* 7 (1996), pp. 414–417.

[82] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Vol. 87. Springer Science & Business Media, 2013.

[83] A. Neubauer. "An a posteriori parameter choice for Tikhonov regularization in the presence of modeling error". In: *Applied Numerical Mathematics* 4.6 (1988), pp. 507–519.

[84] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 1999.

[85] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[86] M. Osborne. "Nonlinear least squares - the Levenberg algorithm revisited". In: *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* 19.03 (1976), pp. 343–357.

[87] C. C. Paige and M. A. Saunders. "LSQR: An algorithm for sparse linear equations and sparse least squares". In: *ACM Transactions on Mathematical Software* 8.1 (1982), pp. 43–71.

[88] M. Piana and M. Bertero. "Projected Landweber method and preconditioning". In: *Inverse Problems* 13.2 (1997), p. 441.

[89] S. Pierret. "Turbomachinery blade design using a Navier-Stokes solver and Artificial Neural Network". In: *ASME Journal of Turbomachinery* 121.3 (1999), pp. 326–332.

[90] M. Powell. "Convergence properties of a class of minimization algorithms". In: *Nonlinear Programming* 2.0 (1975), pp. 1–27.

[91] R. Ramlau. "A modified Landweber method for inverse problems". In: *Numerical Functional Analysis and Optimization* 20.1-2 (1999), pp. 79–98.

[92] A. Rieder. "On the regularization of nonlinear ill-posed problems via inexact Newton iterations". In: *Inverse Problems* 15.1 (1999), p. 309.

[93] F. Rubechini, A. Schneider, A. Arnone, S. Cecchi, and F. A. Malavasi. "A redesign strategy to improve the efficiency of a 17-stage steam turbine". In: *Proceedings of ASME Turbo Expo 2009, 8–12 June 2009, Orlando, Florida.* 2009, pp. 1463–1470.

[94] F. Rubechini, A. Schneider, A. Arnone, F. Daccá, C. Canelli, and P. Garibaldi. "Aerodynamic redesigning of an industrial gas turbine". In: *Proceedings of ASME Turbo Expo 2011, 6-10 June 2011, Vancouver, BC*. 2011, pp. 1387–1394.

[95] M. Saunders. *Systems Optimization Laboratory*. http://web.stanford.edu/group/SOL/software/cgls/.

[96] O. Scherzer. "An iterative multi level algorithm for solving nonlinear ill-posed problems". In: *Numerische Mathematik* 80 (1998), pp. 579–600.

[97] O. Scherzer. "Convergence Criteria of Iterative Methods Based on Landweber Iteration for Solving Nonlinear Problems". In: *Journal of Mathematical Analysis and Applications* 194.3 (1995), pp. 911–933.

[98] O. Scherzer, H. W. Engl, and K. Kunisch. “Optimal a posteriori parameter choice for Tikhonov regularization for solving nonlinear ill-posed problems”. In: *SIAM Journal on Numerical Analysis* 30.6 (1993), pp. 1796–1838.

[99] B. Schölkopf and A. J. Smola. *Learning with kernels: Support Vector Machines, regularization, optimization, and beyond*. MIT Press, 2001.

[100] M. Stephens. “EDF statistics for goodness of fit and some comparisons”. In: *Journal of the American Statistical Association* 69.347 (1974), pp. 730–737.

[101] E. Sturler and M. Kilmer. “A regularized Gauss-Newton trust-region approach to imaging in diffuse optical tomography”. In: *SIAM Journal on Scientific Computing* 34 (2011), pp. 3057–3086.

[102] K. Ueda and N. Yamashita. “Global complexity bound analysis of the Levenberg-Marquardt method for nonsmooth equations and its application to the nonlinear complementarity problem”. In: *Journal of Optimization Theory and Applications* 152.2 (2012), pp. 450–467.

[103] K. Ueda and N. Yamashita. “On a global complexity bound of the Levenberg-Marquardt method”. In: *Journal of Optimization Theory and Applications* 147 (2010), pp. 443–453.

[104] C. Vogel. “A constrained least squares regularization method for nonlinear ill-posed problems”. In: *SIAM Journal on Control and Optimization* 28.1 (1990), pp. 34–49.

[105] C. Vogel. *Computational methods for inverse problems*. SIAM, Frontiers in Applied Mathematics, Providence, 2002.

[106] J. Wait. “Geo-Electromagnetism”. In: (1982).

[107] Y. Wang and Y. Yuan. “Convergence and regularity of trust region methods for nonlinear ill-posed problems”. In: *Inverse Problems* 21 (2005), pp. 821–838.

[108] S. Ward and G. Hohmann. “Electromagnetic theory for geophysical applications”. In: *Electromagnetic Methods in Applied Geophysics*. Vol. 1. 3. 1988, pp. 131–311.

[109] A. Wazwaz. *Linear and nonlinear integral equations*. Vol. 639. Springer, 2011.

[110] N. Yamashita and M. Fukushima. “On the Rate of Convergence of the Levenberg-Marquardt Method”. In: *Computing* 15 (2001), pp. 239–249.

[111] Y. Yuan. “Recent advances in trust region algorithms”. In: *Mathematical Programming* 151.1 (June 2015), pp. 249–281.

[112] J. Zhang. “On the convergence properties of the Levenberg-Marquardt method”. In: *Optimization* 52.6 (2003), pp. 739–756.

[113] J. Zhang and Y. Wang. “A new trust region method for nonlinear equations”. In: *Mathematical Methods of Operations Research* 58 (2003), pp. 283–298.

[114] R. Zhao and J. Fan. “Global complexity bound of the Levenberg–Marquardt method”. In: *Optimization Methods and Software* 31.4 (2016), pp. 805–814.