THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Institut National Polytechnique de Toulouse*
Discipline ou spécialité : *Mathématiques, informatique et télécommunications*

Présentée et soutenue par *Mélodie MOUFFE*
Le *10 février 2009*

Titre : *Multilevel optimization in infinity norm and associated stopping criteria*

JURY
Iain DUFF, Président
François GLINEUR, Rapporteur
Serge GRATTON, Directeur
Michal KOCVARA, Rapporteur
Annick SARTENAER, Membre
Philippe TOINT, Directeur
Xavier VASSEUR, Invité

Ecole doctorale : *Mathématiques, Informatique et télécommunications (MITT)*
Unité de recherche : *CERFACS*
Directeur(s) de Thèse : *Serge GRATTON et Philippe TOINT*
Rapporteurs : *François GLINEUR et Michal Kocvara*

THÈSE

présentée en vue de l'obtention du titre de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE (FRANCE)

Spécialité : Mathématiques, Informatique et Télécommunications

et de

DOCTEUR DES FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX DE NAMUR (BELGIQUE)

Spécialité : Sciences Mathématiques

par

Mélodie MOUFFE

CERFACS

Optimisation multiniveaux en norme infinie et critères d'arrêt associés

Multilevel optimization in infinity norm and associated stopping criteria

Composition du jury :

Iain DUFF	*President*	RAL, UK and CERFACS, France
François GLINEUR	*Referee*	Université Catholique de Louvain, Belgium
Serge GRATTON	*Advisor*	CNES and CERFACS, France
Michal KOCVARA	*Referee*	University of Birmingham, UK
Annick SARTENAER	*Member*	FUNDP University of Namur, Belgium
Philippe TOINT	*Advisor*	FUNDP University of Namur, Belgium
Xavier VASSEUR	*Guest*	CERFACS, France

Je souhaite remercier tout d'abord mes directeurs de thèse, Serge Gratton et Philippe Toint pour le temps et les conseils qu'ils m'ont donnés tout au long de la thèse. Je remercie aussi l'ensemble des membres du jury et en particulier les rapporteurs pour leur avis éclairé sur le manuscript. Je remercie tous mes co-auteurs avec une mention spéciale pour Dimitri avec qui c'est toujours un véritable plaisir de travailler. Je tiens aussi à remercier Iain Duff pour m'avoir donné la chance de réaliser cette thèse et l'ensemble de l'équipe Algo pour leur bonne humeur.

Je remercie bien entendu mes parents pour me soutenir dans tous mes projets ainsi que pour leurs conseils avisés, et Jérôme pour sa présence distrayante et indispensable. Pour terminer, je souhaite remercier de tout mon coeur Jimmy, toujours là pour m'encourager, avec qui une toute autre aventure va commencer.

Abstract of : *Multilevel optimization in infinity norm and associated stopping criteria*

This thesis concerns the study of a multilevel trust-region algorithm in infinity norm, designed for the solution of nonlinear optimization problems of high size, possibly submitted to bound constraints. The study looks at both theoretical and numerical sides.

The multilevel algorithm RMTR_∞ that we study has been developed on the basis of the algorithm created by Gratton, Sartenaer and Toint (2008*b*), which was modified first by replacing the use of the Euclidean norm by the infinity norm and also by adapting it to solve bound-constrained problems.

In a first part, the main features of the new algorithm are exposed and discussed. The algorithm is then proved globally convergent in the sense of Conn, Gould and Toint (2000), which means that it converges to a local minimum when starting from any feasible point. Moreover, it is shown that the active constraints identification property of the trust-region methods based on the use of a Cauchy step can be extended to any internal solver that satisfies a sufficient decrease property. As a consequence, this identification property also holds for a specific variant of our new algorithm.

Later, we study several stopping criteria for nonlinear bound-constrained algorithms, in order to determine their meaning and their advantages from specific points of view, and such that we can choose easily the one that suits best specific situations. In particular, the stopping criteria are examined in terms of backward error analysis, which has to be understood both in the usual meaning (using a product norm) and in a multicriteria optimization framework.

In the end, a practical algorithm is set on, that uses a Gauss-Seidel-like smoothing technique as an internal solver. Numerical tests are run on a FORTRAN 95 version of the algorithm in order to define a set of efficient default parameters for our method, as well as to compare the algorithm with other classical algorithms like the mesh refinement technique and the conjugate gradient method, on both unconstrained and bound-constrained problems. These comparisons seem to give the advantage to the designed multilevel algorithm, particularly on nearly quadratic problems, which is the behavior expected from an algorithm inspired by multigrid techniques.

In conclusion, the multilevel trust-region algorithm presented in this thesis is an improvement of the previous algorithm of this kind because of the use of the infinity norm as well as because of its handling of bound constraints. Its convergence, its behavior concerning the bounds and the definition of its stopping criteria are studied. Moreover, it shows a promising numerical behavior.

Résumé de : *Optimisation multiniveaux en norme infinie et critères d'arrêt associés*

Cette thèse se concentre sur l'étude d'un algorithme multiniveaux de régions de confiance en norme infinie, conçu pour la résolution de problèmes d'optimisation non-linéaires de grande taille pouvant être soumis à des contraintes de bornes. L'étude est réalisée tant sur le plan théorique que numérique.

L'algorithme RMTR_∞ que nous étudions ici a été élaboré à partir de l'algorithme présenté par Gratton, Sartenaer et Toint (2008b), et modifié d'abord en remplaçant l'usage de la norme Euclidienne par une norme infinie, et ensuite en l'adaptant à la résolution de problèmes de minimisation soumis à des contraintes de bornes.

Dans un premier temps, les spécificités du nouvel algorithme sont exposées et discutées. De plus, l'algorithme est démontré globalement convergent au sens de Conn, Gould et Toint (2000), c'est-à-dire convergent vers un minimum local au départ de tout point admissible. D'autre part, il est démontré que la propriété d'identification des contraintes actives des méthodes de régions de confiance basées sur l'utilisation d'un point de Cauchy peut être étendue à tout solveur interne respectant une décroissance suffisante. En conséquence, cette propriété d'identification est aussi respectée par une variante particulière du nouvel algorithme.

Par la suite, nous étudions différents critères d'arrêt pour les algorithmes d'optimisation avec contraintes de bornes afin de déterminer le sens et les avantages de chacun, et ce pour pouvoir choisir aisément celui qui convient le mieux à certaines situations. En particulier, les critères d'arrêts sont analysés en termes d'erreur inverse (backward erreur), tant au sens classique du terme (avec l'usage d'une norme produit) que du point de vue de l'optimisation multicritères.

Enfin, un algorithme pratique est mis en place, utilisant en particulier une technique similaire au lissage de Gauss-Seidel comme solveur interne. Des expérimentations numériques sont réalisées sur une version FORTRAN 95 de l'algorithme. Elles permettent d'une part de définir un panel de paramètres efficaces par défaut et, d'autre part, de comparer le nouvel algorithme à d'autres algorithmes classiques d'optimisation, comme la technique de raffinement de maillage ou la méthode du gradient conjugué, sur des problèmes avec et sans contraintes de bornes. Ces comparaisons numériques semblent donner l'avantage à l'algorithme multiniveaux, en particulier sur les cas peu non-linéaires, comportement attendu de la part d'un algorithme inspiré des techniques multigrilles.

En conclusion, l'algorithme de région de confiance multiniveaux présenté dans cette thèse est une amélioration du précédent algorithme de cette classe d'une part par l'usage de la norme infinie et d'autre part grâce à son traitement de possibles contraintes de bornes. Il est analysé tant sur le plan de la convergence que de son comportement vis-à-vis des bornes, ou encore de la définition de son critère d'arrêt. Il montre en outre un comportement numérique prometteur.

Contents

Chapter 1

Introduction

Nonlinear optimization is a part of applied mathematics that aims at optimizing nonlinear functions. In practice, we look for a minimum of a cost function $f(.)$, called the *objective function*, possibly submitted to some constraints. The traditional optimization problem is written as

$$\min_{x \in \mathcal{F}} f(x), \tag{1.1}$$

where $f(.)$ is a continuous and possibly nonlinear function and where $\mathcal{F}$ is the feasible set. This problem often admits a global solution but possibly also local solutions, that is points minimizing the objective function at least on the intersection of the feasible domain with a (possibly small) open ball. In this work, we look for local solutions of (1.1). Moreover, we are interested in the case where $\mathcal{F}$ is a bound-constrained set, that is if

$$\mathcal{F} = \{x \in \mathbb{R}^n \,|\, l \leq x \leq u\},$$

for some $l, u \in \mathbb{R}^n$ and where the inequality is understood componentwise. In that case, the following sufficient conditions ensure that a vector x^* is an exact local solution of the problem (1.1) :

$$\begin{array}{c} [\nabla_x f(x_*)]_i = 0 \text{ for all } i \notin \mathcal{A}(x_*), \\ \nabla_x^2 f(x_*) \text{ positive definite,} \end{array} \tag{1.2}$$

where $\nabla_x f(.)$ is the gradient of $f(.)$, where $\nabla_x^2 f(.)$ is its Hessian matrix and where

$$\mathcal{A}(\tilde{x}) = \left\{ i \in \{1, ..., n\} \mid \begin{array}{rcl} [\tilde{x}]_i & = & [l]_i \quad \text{and} \quad [\nabla_x f(\tilde{x})]_i > 0 \\ & & \text{or} \\ [\tilde{x}]_i & = & [u]_i \quad \text{and} \quad [\nabla_x f(\tilde{x})]_i > 0 \end{array} \right\}$$

is the set of *binding constraints* at $\tilde{x}$. In practice, we only look for *first-order critical points* of (1.1), that is points that satisfy only the first line of (1.2). To solve this problem, we generally use *iterative methods*. These algorithms produce a sequence of points, called iterates, starting from a given first guess x_0, until the current approximate solution is close enough to a first-order critical solution. In practice, iterative algorithms are stopped when (1.2) is sufficiently close to be satisfied and, for example, when $\mathcal{F} = \mathbb{R}^n$ the stopping criterion is often simply

$$||\nabla_x f(x_*)|| \leq \epsilon,$$

where ϵ is a given threshold. We recommend Kelley (1999) for a discussion about other definitions of the stopping criterion. This stopping criterion has to be adapted to the bound-constrained framework, as will be discussed below. Two main classes of iterative methods are generally used to solve unconstrained and bound-constrained nonlinear optimization problems (see Nocedal and Wright (1999)) : linesearch methods (see Zhu, Byrd, Lu and Nocedal (1997) and Hager and Zhang (2004) among many others) and trust-region methods (Conn, Gould and Toint (1996) or Gould, Orban and Toint (2002), for instance).

At each iteration, linesearch methods select a *descent direction*, defined as a direction along which the cost function can be decreased. A step is then computed from the current iterate along that direction, whose steplength is chosen such that the resulting step leads to a decrease in the objective function. This steplength may be defined as the exact minimizer of $f(.)$ along the chosen descent direction. In that case, the method is called *exact linesearch*, but it is not always efficient in practice. We can also have recourse to an *inexact linesearch* and use the well-known Armijo and Goldstein conditions to try to define a reasonable step (see, for example, Dennis and Schnabel (1983) or Moré and Thuente (1994)). Even if those methods work quite well, our interest in this thesis is focused on trust-region methods, that are less sensitive to ill-conditioned Hessian matrices and to nonconvexity. If the reader wish to have more information about methods to solve general nonlinear optimization problems, we recommend the excellent introduction to the subject by Gould and Leyffer (2003).

Trust-region methods are among the most popular and efficient methods for nonlinear optimization, and they are supported by an extensive theory (see Conn et al., 2000 for a more complete coverage of this subject). Such methods proceed iteratively by minimizing a model of the objective function in a region where the model can be trusted and which is defined in a specific norm. They insist on the fact that each step has to achieve a minimal decrease, known as the Cauchy condition, and they adapt the trust region according to the relative decrease of the objective function in comparison to the decrease of the model. However, as such, these methods do not really exploit problem structure. Our objective is to explore ways to exploit this structure in the frequently occurring situation where the problem at hand can be decomposed into a hierarchy of models with varying degrees of approximation. Indeed, new interest in surface design, data assimilation for weather forecasting (Fisher, 1998) or in optimal control of systems described by partial-differential equations have been the main motivation of this challenging research trend, but other applications such as multi-dimensional scaling (Bronstein, Bronstein, Kimmel and Yavneh, 2005) or quantization schemes (Emilianenko, 2005) also give rise to similar questions. In such problems, one typically considers a (fine) discretization of the infinite-dimensional problem which provides a sufficiently good approximation for the solution. But coarser discretizations are often available that still describe the problem reasonably well, and can therefore be used to improve the efficiency of the numerical solution on the fine discretization.

In the numerical solution of linear systems arising from partial differential equations, techniques have been developed under the name of *multigrid methods* to exploit the case where the problem hierarchy arises from the multilevel discretization of an underlying continuous problem. This field of active research, pioneered by Fedorenko (1964), Bakhvalov (1966) and Brandt (1977), is based on a double observation: to

one side there exist iterative solution methods (called *smoothers*) which are very efficient at reducing the high-frequency, oscillatory components of the error while being possibly very inefficient at reducing their smooth, low-frequency part (the Jacobi and Gauss-Seidel methods are representative examples); on the other hand, the definition of a high frequency component is intrinsically tied to the discretization grid since the finer the grid is, the higher the frequency is representable on this grid. Multigrid methods then proceed by using smoothers to reduce the oscillatory error components on a fine grid, and then consider the remaining smooth components on this fine grid as oscillatory ones on a coarser grid. Broadly speaking, these can again be eliminated using smoothers on the coarser grid, and this technique may be applied recursively. One of the main attractions of well-tuned multigrid methods for linear systems is that their workload increases only linearly with problem size, a crucial feature for the solution of very large instances. We refer the reader to the excellent books by Briggs, Henson and McCormick (2000) and Trottenberg, Oosterlee and Schüller (2001) for a significant coverage of this remarkably efficient class of algorithms.

Exploiting hierarchical problem structure in optimization is much more recent. Several authors have proposed methods that take multilevel hierarchies into account such as Fisher (1998), Nash (2000), Lewis and Nash (2002, 2005), Oh, Milstein, Bouman and Webb (2005), and Hintermüller and Vicente (2005). Kornhuber (1994, 1996, 1997) also developed a method of this type for possibly non-smooth convex bound-constrained problems in the finite-element context. Convergence of this multigrid method is ensured by the successive minimization along coordinate directions generated in Gauss-Seidel-like smoothers, thereby avoiding the need of explicit globalization. On the other hand, Gratton et al. (2008*b*) have proposed a recursive Euclidean-norm trust-region algorithm for general multilevel unconstrained nonconvex minimization. The main attraction of their proposal is to provide the first globally convergent framework for the application of geometric-multigrid-type mechanisms to this class of problems. Moreover, the initial numerical experiments with this algorithm are very promising (see Gratton, Sartenaer and Toint, 2006*a*) and motivate further analysis of methods of this type.

While theoretically satisfying and practically acceptable, the choice of the Euclidean norm for the trust region definition is not without drawbacks. Firstly, and crucially for our concern in this work, Euclidean trust regions do not mix naturally with bound-constrained problems, because the intersection of the trust region (a Euclidean ball) with the feasible domain for bounds (a box) has a more complicated structure than, for example, a simple box. Moreover, the combination of Gauss-Seidel-like smoothing iterations with the Euclidean trust region is unnatural because the smoothing steps consider one variable at a time and are therefore aligned with the coordinate directions. In addition, more technical complications also arise from the fact that, in the proposition of Gratton et al. (2008*b*), the step at a lower level must at the same time be included in the current-level trust region and be such that its prolongation at higher level(s) is included in the higher level(s) trust region(s). As discussed in Gratton et al. (2008*b*), this double requirement implies the use of computationally expensive preconditioners and a special technique for updating the trust region radii which in turn sometimes inefficiently limits the step size.

In order to allow for bound constraints and avoid these technical difficulties, an alternative multilevel algorithm for bound-constrained optimization can be defined

using the infinity- (or max-) norm for the trust region definition. The first purpose of this thesis is to describe this algorithm, which is done at the beginning of Chapter 2. The algorithm, as an added bonus, does not require any imposed preconditioner and is much less restrictive for the lower-level steps than its Euclidean relative for the unconstrained case. Moreover, smoothing iterations which explore directions aligned with the coordinate vectors are well adapted to the box shape of the intersection between the trust region and the set of constraints.

Unfortunately, the convergence theory presented in Gratton et al. (2008*b*, 2006*b*) cannot be applied to this case without significant modifications, not only because of the possible presence of bounds, but also because the algorithm analyzed in these references is itself very dependent on the choice of the Euclidean norm. Our second purpose is thus to prove global convergence of the new algorithm to first-order critical points, that is convergence from arbitrary starting points to limit points satisfying the first-order optimality conditions, which is done in the second part of Chapter 2.

As expected, the algorithm and theory presented here also apply, with minimal adaptations, to the problem of solving sets of nonlinear equations. Indeed, one of the most common techniques in this area is to consider the minimization of some (smooth) norm of the residuals, which can then be viewed as an unconstrained minimization problem, the solution of which yields the desired roots if the residuals converge to zero. As a consequence, the proposed multilevel algorithm also applies to the multilevel solution of nonlinear equations, as does the associated global convergence proof.

We are also interested in the identification of active constraints by the algorithm, that is to determine which inequality constraint will actually be an equality at the exact solution. For convex-constrained problems solved by a trust-region algorithm the internal solver of which is based on the generalized Cauchy step, active constraints identification has been proved by Conn, Gould, Sartenaer and Toint (1993) to happen after a finite number of steps. As a consequence, in the last part of Chapter 2, we show that the identification of active constraints theory presented in that reference can actually be extended without much difficulties to any trust-region method the internal solver of which ensures a sufficient decrease condition, as well as to the use of an infinity-norm trust-region. This result implies that if Gauss-Seidel-like smoothing is used to compute the steps inside a trust-region algorithm for bound-constrained optimization and if this internal solver is shown to satisfy the required sufficient decrease condition, then the algorithm identifies the correct active set after a finite number of iterations.

Moreover, this property also applies to a variant of the multilevel trust-region algorithm, when it is allowed to exploit the multilevel structure at each step only on variables where no constraint is active. Nevertheless, this theoretical result has no clear positive effect on numerical results.

As working with iterative methods in a bound-constrained framework, our next concern is to discuss how to design a good stopping criterion for our algorithm. Many bound-constrained algorithms define their stopping criterion as the norm of the projection of the negative gradient on the constraints (see e.g. Zhu, Byrd, Lu and Nocedal (1994), Lin and Moré (1999), Hager and Zhang (2006) and Xu and Burke (2007)). However another criterion is more often used inside the trust-region community, that was first introduced in Conn et al. (1993), and that has the property of being a first-order approximation of the maximal decrease that can be obtained in

the negative gradient direction. This property is of special interest for trust-region methods because of the importance they give to the decrease achieved at each step.

When they are exactly equal to zero, both stopping criteria are equivalent to the first-order sufficient conditions (first line of (1.2)). However, using them when the data is approximate, which is the case when working on discretized problems, is not straightforward. In addition, a suitable stopping criterion for approximate problems has already been designed in the linear case. As a consequence, in Chapter 3, we to follow a *backward error analysis* approach and see if it leads to already known stopping criteria for our nonlinear bound-constrained optimization context. This technique consists in assuming that the current approximate solution solves exactly a nearby problem, and to measure the distance between the two problems instead of the distance between the two solutions. This technique is well-known and has been intensely studied in linear algebra (see Rigal and Gaches (1967), Cox and Higham (1998), Golub and Van Loan (1983), Chaitin-Chatelin and Fraysse (1996) or Higham (1996)), but it is the first time backward error analysis is used to design stopping criteria for bound-constrained nonlinear optimization. If we decide to stop the algorithm when the backward error is smaller than some threshold, our approach has the advantage that this threshold can be determined as a function of the uncertainties we know on the gradient, the objective function or the bound constraints. Indeed, there is no point trying to reduce the distance between the original and the nearby problem more than these uncertainties. In the end of Chapter 3, we will check that the stopping criterion defined by this analysis satisfies all the properties needed for the convergence of the previously defined multilevel trust-region algorithm.

We finally define in Chapter 4 a practical algorithm where the internal solvers are chosen and several option choices, so far left unspecified in the theoretical algorithm, are described. We apply this particular implementation of the method on a few representable large-scale both unconstrained and bound-constrained test problems. We run a first battery of tests to determine suitable default values for the parameters of the method. We then use this optimal configuration to compare our algorithm with other competing methods in this field and illustrate the strength of multilevel trust-region methods. We finally compare the numerical behavior of different stopping criteria, in particular the classical one for the trust-region methods and the one designed by means of backward error analysis.

Some conclusions and perspectives are finally discussed in Chapter 5.

Chapter 2

A general recursive multilevel infinity-norm trust-region algorithm for bound-constrained optimization

In this chapter, after recalling basic concepts of nonlinear optimization, we introduce the main ideas defining the new multilevel algorithm, prove its convergence from arbitrary starting points, and extend the identification of active constraints theory for trust-region methods to the use of any internal solver that satisfies a sufficient decrease condition.

2.1 Preliminary concepts

We consider the nonlinear optimization problem

$$\min f(x), \tag{2.1}$$

where f is a twice-continuously differentiable objective function which maps $\mathbb{R}^n$ into $\mathbb{R}$ and is bounded below. We are interested in finding a first-order critical solution x_* of (2.1) in the sense $[\nabla_x f(x_*)]_j = 0$ for all j, where $[v]_j$ represents the j^{th} component of a vector v. A very classical way of solving this problem is to apply Newton's method. This is an iterative method in the sense that, given an initial point x_0, it produces a sequence $\{x_k\}$ of iterates, hopefully converging towards the solution of (2.1). At some iteration k, starting from x_k, the method approximates the objective function $f(x)$ around x_k by its second order Taylor expansion. Each step s_k^N produced by Newton's method is the result of the minimization of this Taylor model

$$s_k^N = \min_s f(x_k) + \nabla_x f(x_k)^T s + \frac{1}{2} s^T \nabla_x^2 f(x_k) s,$$

where $\nabla_x f(.)$ is the gradient of $f(.)$, $\nabla_x^2 f(.)$ is its Hessian matrix and v^T denotes the transpose of a vector v. This expression is equivalent to say that $\nabla_x f(x_k)^T + \nabla_x^2 f(x_k) s_k^N = 0$ and $\nabla_x^2 f(x_k) > 0$. In consequence, the next iterate is given by

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

The algorithm is stopped as soon as the gradient is close enough to zero in the sense $||\nabla_x f(x_k)|| < \epsilon$, where the gradient is measured in a suitable norm and where ϵ is a chosen tolerance. Newton's method is locally quadratically convergent under regularity conditions of $f(.)$ at the solution x^*. In other words, this method can be very quick when close to the solution but can fail in finding a solution if the minimization starts too far from it. To overcome this drawback, we can be interested in trust-region methods. They are well-known and very efficient methods to solve nonlinear optimization problems for two main reasons. First they are *globally convergent*, which means that they find a first-order critical point when given any starting point x_0. Their second advantage is that they reduce to Newton's method when close enough to the solution and, consequently, exhibit a *local quadratic convergence*. Let us now look more closely at the way the basic trust-region algorithm works. At each iteration k, the algorithm constructs a model m_k of the objective function around the current iterate x_k, which is generally a quadratic approximation of $f(x)$. It also defines a *trust region* $\mathcal{B}_k$ centered at x_k and defined by its radius $\Delta_k > 0$, in which the model is assumed to be adequate. A step s_k is then computed inside the trust region, that induces a sufficient reduction in the model. The objective function is calculated at the *trial point*, $x_k + s_k$, and this trial point is accepted as the next iterate if and only if ρ_k, the ratio of achieved reduction (in the objective function) to predicted reduction (in its local model), is reasonable (typically larger than a small positive constant η_1). The radius of the trust region is finally updated: it is decreased if the trial point is rejected and left unchanged or increased if ρ_k is sufficiently large. The algorithm is stopped as soon as the norm of the gradient is smaller than a chosen tolerance, that is $||\nabla_x f(x_k)|| < \epsilon$. The introduction of the trust region $\mathcal{B}_k$ ensures that the algorithm is globally convergent, while the definition of the model implies that when approaching the solution, m_k becomes very similar to the objective function and, therefore, the trust-region radius Δ_k tends to infinity, such that the trust-region method finally reduces to Newton's method. We refer the reader to Conn et al. (2000) for a comprehensive coverage of this subject.

We now consider the bound-constrained optimization problem

$$\min_{x \in \mathcal{F}} f(x), \tag{2.2}$$

where $\mathcal{F} = \{x \in \mathbb{R}^n | l \leq x \leq u\}$ is a set of bound constraints and where $l, u \in \overline{\mathbb{R}}^n$ and are possibly infinite. In this case, finding a first-order critical solution x_* of (2.2) means finding x_* such that

$$[\nabla_x f(x_*)]_j = 0 \quad \text{for all } j \notin \mathcal{A}(x_*), \tag{2.3}$$

where $\mathcal{A}(x) = \mathcal{A}^-(x) \cup \mathcal{A}^+(x)$ is the set of *binding constraints* with

$$\begin{aligned} \mathcal{A}^-(x) &= \{j \in \{1, ..., n\} \mid [x]_j = [l]_j \quad \text{and} \quad [\nabla_x f(x)]_j > 0\} \\ \mathcal{A}^+(x) &= \{j \in \{1, ..., n\} \mid [x]_j = [u]_j \quad \text{and} \quad [\nabla_x f(x)]_j < 0\}. \end{aligned}$$

In that context, the unconstrained trust-region algorithm can be easily adapted to become Algorithm 2.1.1 below. Nevertheless, a few comments are necessary. We first define a *criticality measure* $\chi_k = \chi(x_k)$ that has to be equal to zero when evaluated at the exact solution x_* and which is used as a stopping criteria designed for bound-constrained optimization. Usual criticality measures are, for example,

$\chi_k^{out,2} = \|\text{Proj}_{\mathcal{F}}(x_k - \nabla_x f(x_k)) - x_k\|_2$ where $\text{Proj}_{\mathcal{F}}$ is the orthogonal projection onto the box $\mathcal{F}$, or $\chi_k^{tr} = |\min_{x_k+d\in\mathcal{F},\|d\|_\infty\leq 1}\langle\nabla_x f(x_k), d\rangle|$ (see e.g. Conn et al. (2000)). The choice of the most suitable definition to use is not obvious and will be discussed in detail in Chapter 3. A second point to specify is that we have chosen to define the trust-region constraint in infinity-norm to make it easier to intersect with the original set of bound constraints $\mathcal{B}_k = \{x_k + s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_k\}$. Finally, the chosen model here is

$$m_k(x_{k+1}) = f(x_k) + g_k^T(x_{k+1-x_k}) + \frac{1}{2}(x_{k+1-x_k})^T H_k(x_{k+1-x_k}), \tag{2.4}$$

where $g_k = \nabla_x f(x_k)$ and where H_k is a symmetric $n \times n$ approximation of $\nabla_x^2 f(x_k)$. In what follows, we express the model as a function of the step s_k we are looking for by replacing x_{k+1} by $x_k + s_k$. The condition about the sufficient decrease, known as the modified Cauchy condition, is given by

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{red}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k, 1\right], \tag{2.5}$$

where $\kappa_{red} \in (0, \frac{1}{2})$ and where $\beta_k = 1 + \|H_k\|$. Despite its apparently technical character, this requirement is not overly restrictive and can be guaranteed in practical algorithms, as described for instance in Section 12.2.1 of Conn et al. (2000), or in the following Section 4.1.1 and Appendix B.1.

Algorithm 2.1.1: BTR(x_0, g_0, ϵ)

Step 0: Initialization. Compute $f(x_0)$, define $\mathcal{B}_0 = \{x_0 + s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_0\}$ and set $k = 0$.

Step 1: Step computation. Compute a step $s_k \in \mathcal{B}_k$ that sufficiently reduces the model m_k defined by (2.4) in the sense of (2.5). Set $\delta_k = m_k(x_k) - m_k(x_k + s_k)$.

Step 2: Acceptance of the trial point. Compute $f(x_k + s_k)$ and $\rho_k = [f(x_k) - f(x_k + s_k)]/\delta_k$. If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise, define $x_{k+1} = x_k$.

Step 3: Termination. Compute g_{k+1} and χ_{k+1}. If $\chi_{k+1} \leq \epsilon$, then return with the approximate solution $x_* = x_{k+1}$.

Step 4: Trust-Region Update. Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, +\infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2\Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1, \end{cases}$$

where $0 < \eta_1 < \eta_2 < 1$ and $0 < \gamma_1 < \gamma_2 < 1$. Define $\mathcal{B}_{k+1} = \{x_{k+1} + s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_{k+1}\}$, increment k by one and go to Step 1.

2.2 The problem and algorithm

If a hierarchy of descriptions for problem (2.2) is known, we consider exploiting the knowledge of this hierarchy, as proposed in Gratton et al. (2008*b*). To be more specific, suppose that a collection of functions $\{f_i\}_{i=0}^r$ is available, each f_i being a twice-continuously differentiable function from $\mathbb{R}^{n_i}$ to $\mathbb{R}$ (with $n_i \geq n_{i-1}$). We assume that $n_r = n$ and $f_r(x) = f(x)$ for all $x \in \mathbb{R}^n$, giving back our original problem. We also make the assumption that f_i is "more costly" to minimize than f_{i-1} for each $i = 1, \ldots, r$. This may be the case if the f_i represent increasingly finer discretizations of the same infinite-dimensional objective. To fix terminology, we will refer to a particular i as a *level*. We use the first subscript i in all subsequent subscripted symbols to denote a quantity corresponding to the i-th level, ranging from coarsest ($i = 0$) to finest ($i = r$) (meaning in particular, if applied to a vector, that this vector belongs to $\mathbb{R}^{n_i}$). Some relation must of course exist between the variables of two successive functions of the collection set $\{f_i\}_{i=0}^r$. We thus assume that, for each $i = 1, \ldots, r$, there exist a full-rank linear operator R_i from $\mathbb{R}^{n_i}$ into $\mathbb{R}^{n_{i-1}}$ (the restriction) and another full-rank operator P_i from $\mathbb{R}^{n_{i-1}}$ into $\mathbb{R}^{n_i}$ (the prolongation) such that

$$\sigma_i P_i = R_i^T, \tag{2.6}$$

for some known constant $\sigma_i > 0$, where P_i and R_i are interpreted as restriction and prolongation between a fine and a coarse grid. These assumptions are common to a number of multilevel approaches in optimization (Fisher, 1998, Nash, 2000, Gratton et al., 2008*b*) or in the solution of nonlinear systems of equations (see Briggs et al., 2000 and the references therein). For simplicity of notations, and because this is often the case in practice, we assume, without loss of generality, that $\|R_i\|_\infty = 1$ for all i (as we can choose $\sigma_i = 1/\|P_i^T\|_\infty$).

When the problem has two levels (r and $r-1$), the main idea is to use f_{r-1} as a model for $f_r = f$ in the neighborhood of the current iterate $x_{r,k}$, which is cheaper than using Taylor's quadratic model at level r. We will use the word model from now on both to designate Taylor's model and f_{r-1}, since the lower representation of the objective function is now seen as a possible model of f_r. We then minimize the (potentially nonquadratic) model f_{r-1} *using a trust-region algorithm* at level $r-1$, whose iteration ℓ therefore features its own box-shaped trust-region $\mathcal{B}_{r-1,\ell}$ of radius $\Delta_{r-1,\ell}$. This minimization is carried under a set of constraints inherited from level r and from the initial point $x_{r-1,0} = R_r x_{r,k}$, until some approximate constrained minimizer $x_{r-1,*}$ is found. The resulting step is then prolongated to level r by computing

$$s_{r,k} = P_r(x_{r-1,*} - x_{r-1,0}).$$

The main difficulty is to specify the form of the constraints inherited from the upper level. First of all, the resulting feasible set (at the lower level) must be a box in order to preserve the coherence and efficiency of the algorithm across levels. We also wish to guarantee the feasibility *at the upper level* of the prolongated trial point $x_{r,k} + s_{r,k}$ with respect to the bound constraints. Finally, we would like to ensure that this trial step lies within the upper-level trust region $\mathcal{B}_{r,k}$. Unfortunately, *the prolongation of the restriction of a box at level r back to level r is in general* not *included in the original box*, as shown in Figure 2.1.

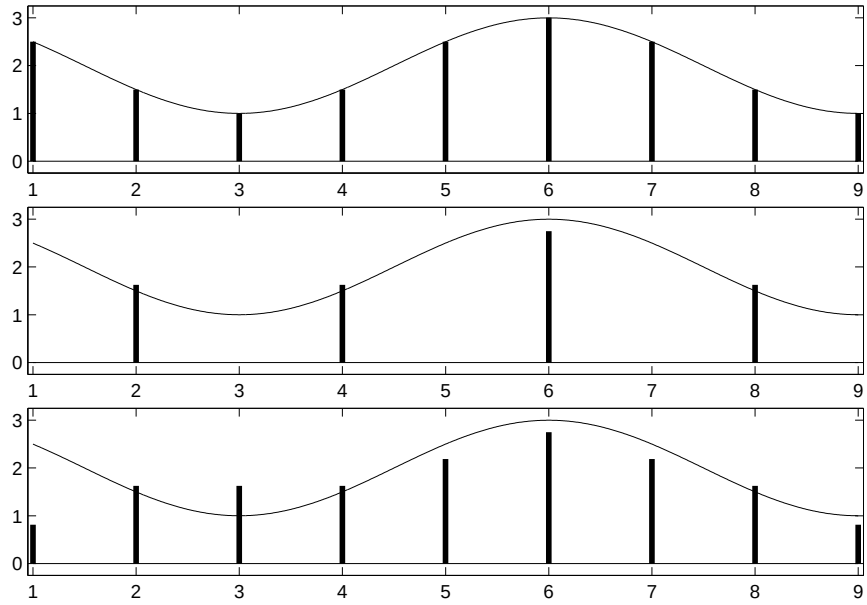


Figure 2.1: Prolongation and restriction of bounds. In this figure, one considers the set of continuous functions $\phi(t)$ for $t \in [1, 9]$ with a zero lower bound and an upper bound given by $2 + \cos(\pi t/3)$. The vertical bars in the upper graph show the possible ranges for the values $\phi(1), \ldots, \phi(9)$ for such functions, considered here as problem variables. The vertical bars in the middle graph show the ranges obtained by applying the restriction operator (corresponding to the normalized transpose of the linear interpolation for a coarser grid of 4 discretization points) to the set of bounds obtained in the upper graph. The vertical bars in the lower graph finally correspond to applying the prolongation (linear interpolation) to the bounds obtained in the middle graph. One notices that these latter ranges are *not* always included in the original ranges of the upper graph.

We are thus forced to alter our technique for representing an upper-level box at the lower level if we insist that its prolongation satisfies the constraints represented by the upper-level box. This is highly desirable for the upper-level box $\mathcal{F}_r$ defining the original bound constraints of the problem, because we wish to preserve feasibility at all levels. On the other hand, we might accept some flexibility for the lower-level box corresponding to the upper-level trust region $\mathcal{B}_{r,k}$, because one expects that a step whose norm is proportional to the trust-region size would be enough to ensure convergence (even if strict inclusion does not hold) without being unduly restrictive. Thus we are lead to a two-pronged strategy, where we separately represent, on one hand, the bound constraints at the lower level in a way guaranteeing feasibility of the prolongated step, and, on the other hand, the upper trust region, possibly more loosely. If $\mathcal{F}_{r-1}$ is the representation of the bound constraints at the lower-level and $\mathcal{A}_{r-1}$ that of the upper trust region, then the step at iteration ℓ of the lower-level minimization must be included in the box

$$\mathcal{W}_{r-1,\ell} \stackrel{\text{def}}{=} \mathcal{F}_{r-1} \cap \mathcal{A}_{r-1} \cap \mathcal{B}_{r-1,\ell}. \tag{2.7}$$

We discuss below how $\mathcal{F}_{r-1}$ and $\mathcal{A}_{r-1}$ are computed.

If more than two levels are available ($r > 1$), the same technique can be applied recursively, the process stopping at level 0, where there is no coarser model, and thus Taylor's model is always used. Let us consider the details of this process in this more general situation. Consider iteration k at level i, and assume that $x_{i,k}$ is an iterate

in the minimization of f_i inside an iteration q at level $i+1$ where f_i has been chosen as a model for f_{i+1} (i.e. a *recursive iteration*).

We start by considering the representation of the problem's bounds at lower levels. At level i, we define

$$\mathcal{F}_i \stackrel{\text{def}}{=} \{x \mid l_i \leq x \leq u_i\} \tag{2.8}$$

the "restricted" feasible domain, where

$$[l_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \frac{1}{\|P_{i+1}\|_\infty} \max_{t=1,\ldots,n_{i+1}} \left\{ \begin{array}{lll} [l_{i+1} - x_{i+1,q}]_t & \text{when} & [P_{i+1}]_{tj} > 0 \\ [x_{i+1,q} - u_{i+1}]_t & \text{when} & [P_{i+1}]_{tj} < 0 \end{array} \right\} \tag{2.9}$$

and

$$[u_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \frac{1}{\|P_{i+1}\|_\infty} \min_{t=1,\ldots,n_{i+1}} \left\{ \begin{array}{lll} [u_{i+1} - x_{i+1,q}]_t & \text{when} & [P_{i+1}]_{tj} > 0 \\ [x_{i+1,q} - l_{i+1}]_t & \text{when} & [P_{i+1}]_{tj} < 0 \end{array} \right\} \tag{2.10}$$

for $j = 1, \ldots, n_i$. The idea behind this generalization of the definition by Gelman and Mandel (1990), originally stated for more specific prolongation operators[(1)], is to use the structure of P_{i+1} to compute a coarse set of bounds $\mathcal{F}_i$ *in order to guarantee that its prolongation is feasible for the fine level*, that is

$$l_{i+1} \leq x_{i+1} + P_{i+1}(l_i - x_i) \leq x_{i+1} + P_{i+1}(u_i - x_i) \leq u_{i+1}$$

for all $x_{i+1} \in \mathcal{F}_{i+1}$, for all $x_i \in \mathcal{F}_i$. This property is proved in Lemma 2.3.2 below. Figure 2.2 on the following page shows the application of the (generalized) Gelman-Mandel's coarse bounds and their prolongation on the example of Figure 2.1.

We now turn to the representation of the upper trust region at the lower level. At level i we also define

$$\mathcal{A}_i = \{x \mid v_i \leq x \leq w_i\}, \tag{2.11}$$

the restriction of the trust-region constraints inherited from levels r to $i+1$ through $x_{i+1,q}$, computed using the restriction operator R_{i+1}. The j-th components of v_i and w_i are

$$\begin{aligned} [v_i]_j \quad = \quad & \sum_{u=1,[R_{i+1}]_{ju}>0}^{n_{i+1}} [R_{i+1}]_{ju} [\max(v_{i+1}, x_{i+1,q} - \Delta_{i+1,q} e)]_u \\ & + \sum_{u=1,[R_{i+1}]_{ju}<0}^{n_{i+1}} [R_{i+1}]_{ju} [\min(w_{i+1}, x_{i+1,q} + \Delta_{i+1,q} e)]_u \end{aligned} \tag{2.12}$$

---

(1) The original formulation is restricted to the case where $||P_{i+1}||_\infty \leq 1$ and $P_{i+1} > 0$, and is given by

$$[l_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \max_{t=1,\ldots,n_{i+1}:[P_{i+1}]_{tj}>0} [l_{i+1} - x_{i+1,q}]_t,$$

$$[u_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \max_{t=1,\ldots,n_{i+1}:[P_{i+1}]_{tj}>0} [x_{i+1,q} - u_{i+1}]_t.$$

We extend this definition to cover prolongation operators with $||P_{i+1}||_\infty > 1$ and also to handle negative elements in P_{i+1} (as in cubic interpolation, for instance), which imposes taking both upper and lower bounds at the upper level into account for the definition of the upper and lower bounds at the coarse level.

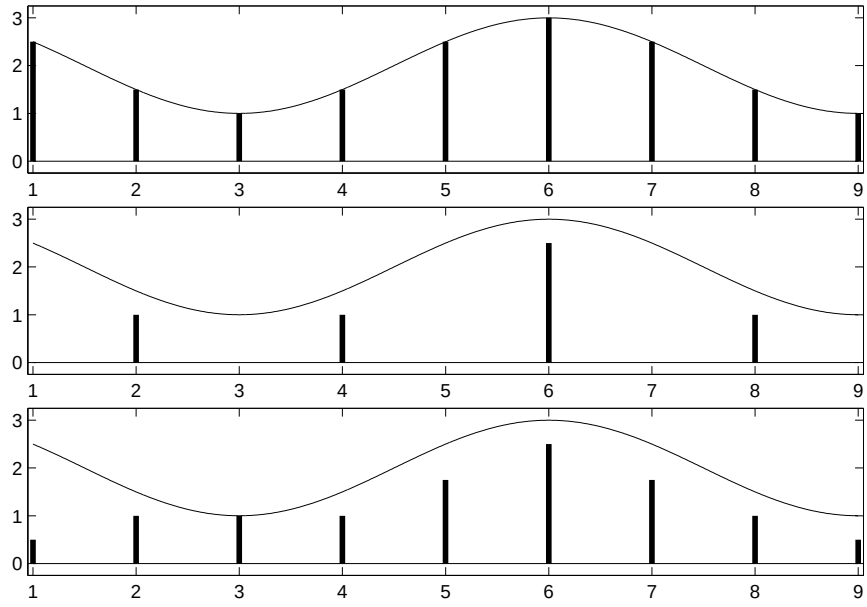


Figure 2.2: Prolongation of Gelman and Mandel's bounds for the same example as in Figure 2.1. As in this figure, the vertical bars in the upper graph show the possible ranges for the values $\phi(1), \ldots, \phi(9)$. The vertical bars in the middle graph now show the ranges obtained by deriving the generalized Gelman and Mandel's bounds from the set of bounds obtained in the upper graph, and the vertical bars in the lower graphs finally correspond to applying the prolongation (linear interpolation) to the bounds obtained in the middle graph.

and

$$\begin{aligned}[w_i]_j \quad = \quad & \sum_{u=1,[R_{i+1}]_{ju}>0}^{n_{i+1}} [R_{i+1}]_{ju}[\min(w_{i+1}, x_{i+1,q} + \Delta_{i+1,q}e)]_u \\ & + \sum_{u=1,[R_{i+1}]_{ju}<0}^{n_{i+1}} [R_{i+1}]_{ju}[\max(v_{i+1}, x_{i+1,q} - \Delta_{i+1,q}e)]_u,\end{aligned} \tag{2.13}$$

where $e \in \mathbb{R}^n$ is a vector whose components are all equal to 1 (and where we define $v_r = -\infty$ and $w_r = +\infty$ for consistency). Notice that, as allowed in our above discussion, the choice of using R_i to restrict these bounds implies that recursive iterates at level i are not necessarily included in the level i trust region anymore but cannot be very far from it. Indeed, recalling that $||R_i||_\infty = 1$ for $i = 1, ..., r$, we have that

$$\|x_{i,k+1} - x_{i,k}\|_\infty \leq \|P_i\|_\infty \|x_{i-1,*} - x_{i-1,0}\|_\infty \leq 2\|P_i\|_\infty \Delta_{i,k}, \tag{2.14}$$

where the last inequality is proved in Lemma 2.3.3 below.

If the trust region at level i around iterate $x_{i,k}$ is defined by

$$\mathcal{B}_{i,k} = \{x_{i,k} + s \in \mathbb{R}^{n_i} \mid \|s\|_\infty \leq \Delta_{i,k}\},$$

we then have to find a step $s_{i,k}$ which sufficiently reduces a model of f_i in the region

$$\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{A}_i \cap \mathcal{B}_{i,k}. \tag{2.15}$$

Observe that the set $\mathcal{W}_{i,k}$ can either be viewed both as $\mathcal{W}_{i,k} = \mathcal{L}_i \cap \mathcal{B}_{i,k}$, the intersection of a level dependent domain $\mathcal{L}_i \stackrel{\text{def}}{=} \mathcal{F}_i \cap \mathcal{A}_i$ with an iteration dependent

trust-region $\mathcal{B}_{i,k}$, or as $\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{S}_{i,k}$, the intersection of $\mathcal{F}_i$, the feasible set for hard constraints, with $\mathcal{S}_{i,k} \stackrel{\text{def}}{=} \mathcal{A}_i \cap \mathcal{B}_{i,k}$, the feasible set for soft ones. This last set can be interpreted as a "composite" trust region which includes all constraints imposed by trust regions at level i and higher. Note that all the involved sets are boxes, which makes their representation and intersection computationally easy.

Figure 2.3 on the next page illustrates the process to compute a recursive step in the example already used in Figures 2.1 and 2.2. In this figure, the values of the variables at successive iterates are shown by horizontally barred circles and the steps by arrows. Trust-region bounds on each variable are shown with vertical brackets, the sets $\mathcal{S}_{r,k}$ and $\mathcal{A}_{r-1}$ by thin vertical boxes, the set $\mathcal{L}_{r-1}$ by fatter vertical boxes and the sets $\mathcal{F}_r$ and $\mathcal{F}_{r-1}$ by thick lines. At stage 3, $\mathcal{W}_{r-1,0}$ is given by the intersection of the fat boxes representing $\mathcal{L}_{r-1}$ with the brackets representing $\mathcal{B}_{r-1,0}$.

Once $\mathcal{W}_{i,k}$ is known, according to the situation, we then choose a model for f_{i+1} as one of f_i or

$$m_{i+1,q}(x_{i+1,q} + s_{i+1}) = f_{i+1}(x_{i+1,q}) + \langle g_{i+1,q}, s_{i+1}\rangle + \tfrac{1}{2}\langle s_{i+1}, H_{i+1,q} s_{i+1}\rangle, \qquad (2.16)$$

the usual truncated Taylor series for f_{i+1} (with $g_{i+1,q} = \nabla_x f_{i+1}(x_{i+1,q})$ and $H_{i+1,q}$ being a general symmetric approximation of $\nabla_x^2 f_{i+1}(x_{i+1,q})$). As it will be discussed in Chapter 4, this freedom of choice is crucial for the application of multigrid-type techniques in our context. In the latter case, we assume that f_{i+1} and its coarse model, the lower-level function f_i, are *first order coherent*, that is $g_{i,0} = R_{i+1} g_{i+1,q}$. This assumption is not restrictive, as we can always choose a first order coherent coarse model of f_{i+1} by adding a gradient correction term to f_i as in

$$f_i(x_{i,0} + s_i) + \langle R_{i+1} g_{i+1,q} - \nabla_x f_i(x_{i,0}), s_i\rangle.$$

If one chooses the function f_i as a model for f_{i+1} (which is only possible if $i > 0$), the determination of the step then consists in (approximately) solving the lower-level bound-constrained problem

$$\min_{x_{i,0}+\tilde{s}_i \in \mathcal{L}_i} f_i(x_{i,0} + \tilde{s}_i). \qquad (2.17)$$

This minimization produces a step s_i such that $f_i(x_{i,0} + s_i) < f_i(x_{i,0})$ which must be then brought back to level $i+1$ by the prolongation P_{i+1}, i.e. $s_{i+1} = P_{i+1} s_i$. Note that

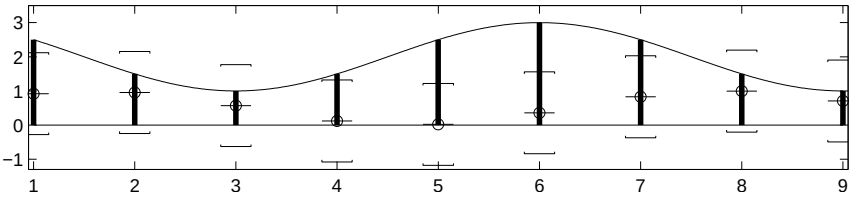
$$\langle g_{i+1,q}, s_{i+1}\rangle = \langle g_{i+1,q}, P_{i+1} s_i\rangle = \frac{1}{\sigma_{i+1}} \langle R_{i+1} g_{i+1,q}, s_i\rangle. \qquad (2.18)$$

As the decrease of f_i achieved by s_i can be approximated to first-order by $f_i(x_{i,0}) - f_i(x_{i,0} + s_i) \approx \langle g_{i,0}, s_i\rangle = \langle R_{i+1} g_{i+1,q}, s_i\rangle$, the decrease of the model at level $i+1$ when computing steps at level i is computed, using (2.18), as $[f_i(x_{i,0}) - f_i(x_{i,0} + s_i)]/\sigma_{i+1}$.

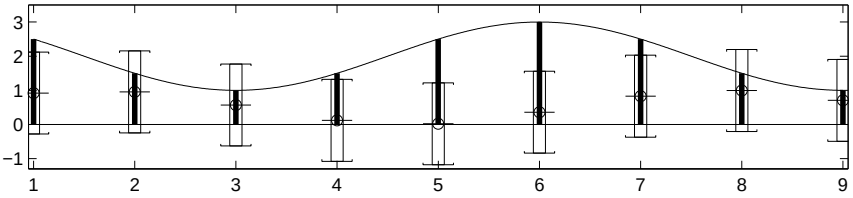
But does it always make sense to use the lower level model? The answer obviously depends on the benefit expected from the solution of (2.17). In Gratton et al. (2008*b*), it sufficed to test if $\|g_{i,0}\|_2 = \|R_{i+1} g_{i+1,q}\|_2$ was large enough compared to $\|g_{i+1,q}\|_2$. However, this criticality measure is inadequate in our context because (2.17) is now a bound-constrained problem. In the sequel of this chapter we assume that we use a criticality measure $\chi_{i+1,q}$ designed for bound-constrained optimization[(2)] for each

[(2)] such as $\mu_{i+1,q} = \|\mathrm{Proj}_{i+1,q}(x_{i+1,q} - g_{i+1,q}) - x_{i+1,q}\|_2$ where $\mathrm{Proj}_{i+1,q}$ is the orthogonal projection onto the box $\mathcal{L}_{i+1}$ or $\chi_{i+1,q} \stackrel{\text{def}}{=} \chi(x_{i+1,q}) = |\min_{\substack{x_{i+1,q}+d \in \mathcal{L}_{i+1} \\ \|d\|_\infty \le 1}} \langle g_{i+1,q}, d\rangle|$.

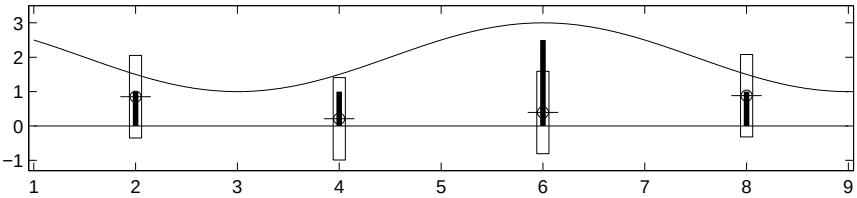
The iterate $x_{r,k}$, and the sets $\mathcal{F}_r$ (thick lines), $\mathcal{A}_r = \mathbb{R}^9$ and $\mathcal{B}_{r,k}$ (brackets) are given at level r:



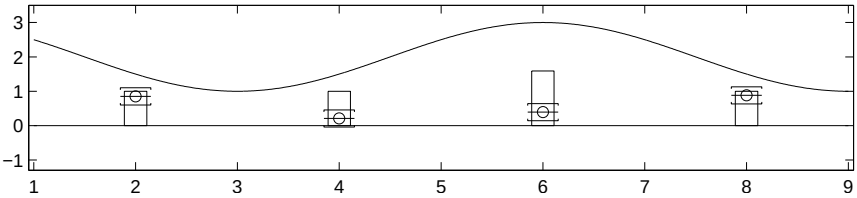
1) compute $S_{r,k} = \mathcal{A}_r \cap \mathcal{B}_{r,k}$ (thin boxes) at level r:



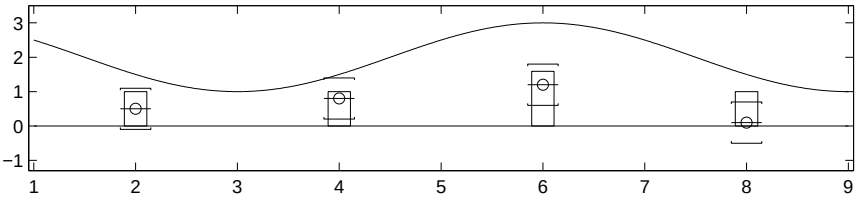
2) restrict the problem: compute $x_{r-1,0} = Rx_{r,k}$, $\mathcal{F}_{r-1}$ (thick lines) and $\mathcal{A}_{r-1}$ (thin boxes) at level $r-1$:



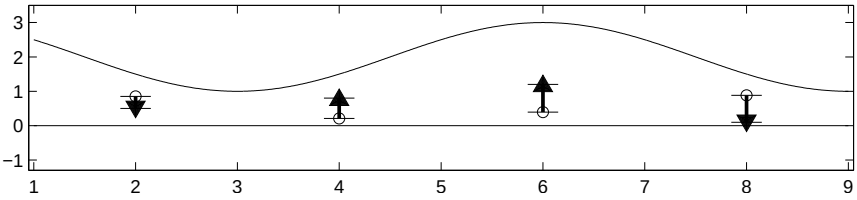
3) compute $\mathcal{L}_{r-1} = \mathcal{F}_{r-1} \cap \mathcal{A}_{r-1}$ (fat boxes) and add $\mathcal{B}_{r-1,0}$ (brackets) at level $r-1$:



4) perform some iterations at level $r-1$, yielding $x_{r-1,*}$ (circle) and $\mathcal{B}_{r-1,*}$(new brackets):



6) compute $x_{r-1,*} - x_{r-1,0}$ (arrows-horizontal line), the total step at level $r-1$:



7) prolongate the step (arrows-horizontal line) and compute the level-r trial point $x_{r,k} + P(x_{r-1,*} - x_{r-1,0})$.

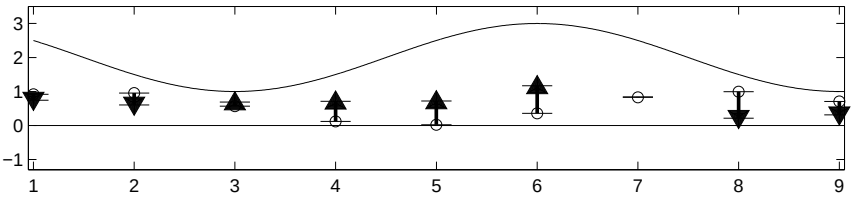


Figure 2.3: The definition of the various sets and the step computation for the example of Fig. 2.1.

$x_{i+1,q} \in \mathcal{L}_{i+1}$.Practical choices for this measure will be discussed in Chapter 3. Then if the restriction of the problem from the non-critical iterate $x_{i+1,q}$ at level $i+1$ to level i is not already first-order critical, that is if

$$\chi_{i,0} \geq \kappa_\chi \chi_{i+1,q}, \tag{2.19}$$

for some constant $\kappa_\chi \in (0, \max\{1, \sigma_i\})$, we may proceed at this lower level. Otherwise, the recursion is useless and we should use (2.16) instead.

Once we have decided to approximately solve (2.17), we must also decide what we mean by "approximately". We choose to terminate the minimization at level r if $\chi_{r,k} \leq \epsilon_r$ for some $\epsilon_r > 0$ and, in the spirit of (2.19), to terminate the lower level minimization at iterate (i,p) as soon as the inequality

$$\chi_{i,p} < \epsilon_i \stackrel{\text{def}}{=} \kappa_\chi \epsilon_{i+1}, \tag{2.20}$$

holds. We then define $x_{i,*} = x_{i,p}$, $s_i = x_{i,*} - x_{i,0}$ and $s_{i+1,q} = P_{i+1} s_i$.

If, on the other hand, we decide at iteration $(i+1,q)$ to use Taylor's model $m_{i+1,q}$ given by (2.16), a step $s_{i+1,q}$ is then computed that produces a sufficient decrease in the value of this model in its usual meaning for trust-region methods with convex constraints (defined here by the set $\mathcal{L}_{i+1}$), that is, $s_{i+1,q}$ is such that it satisfies

$$m_{i+1,q}(x_{i+1,q}) - m_{i+1,q}(x_{i+1,q} + s_{i+1,q}) \geq \kappa_{\text{red}} \chi_{i+1,q} \min\left[\frac{\chi_{i+1,q}}{\beta_{i+1,q}}, \Delta_{i+1,q}, 1\right], \tag{2.21}$$

for some constant $\kappa_{\text{red}} \in (0, \frac{1}{2})$ and $\beta_{i+1,q} \stackrel{\text{def}}{=} 1 + \|H_{i+1,q}\|_{\infty,1}$ where $\|M\|_{\infty,1} \stackrel{def}{=} \max_{x \neq 0} \left\{ \frac{\|Mx\|_1}{\|x\|_\infty} \right\}$ for all matrices M. Despite its apparently technical character, this requirement, known as the modified Cauchy condition, is not overly restrictive and can be guaranteed in practical algorithms, as described for instance in Section 12.2.1 of Conn et al. (2000).

We now specify our algorithm formally, as Algorithm RMTR$_\infty$ on the following page. It uses the constants $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_1 \leq \gamma_2 < 1$ and Δ_i^{s} $(i = 0, \ldots, r)$.

Algorithm 2.2.1: RMTR$_\infty$ $(i, x_{i,0}, g_{i,0}, \chi_{i,0}, \mathcal{F}_i, \mathcal{A}_i, \epsilon_i)$

Step 0: Initialization. Compute $f_i(x_{i,0})$. Set $k = 0$ and

$$\mathcal{L}_i = \mathcal{F}_i \cap \mathcal{A}_i \quad \text{and} \quad \mathcal{W}_{i,0} = \mathcal{L}_i \cap \mathcal{B}_{i,0},$$

where $\mathcal{B}_{i,0} = \{x_{i,0} + s \in \mathbb{R}^{n_i} \,|\, \|s\|_\infty \leq \Delta_{i,0} = \Delta_i^{\mathrm{s}}\}$.

Step 1: Model choice. If $i = 0$, go to Step 3. Else, compute $\mathcal{L}_{i-1}$ and $\chi_{i-1,0}$. If (2.19) fails, go to Step 3. Otherwise, choose to go to Step 2 or to Step 3.

Step 2: Recursive step computation. Call Algorithm

$$\mathbf{RMTR}_\infty(i-1, R_i x_{i,k}, R_i g_{i,k}, \chi_{i-1,0}, \mathcal{F}_{i-1}, \mathcal{A}_{i-1}, \kappa_\chi \epsilon_i),$$

yielding an approximate solution $x_{i-1,*}$ of (2.17). Then define $s_{i,k} = P_i(x_{i-1,*} - R_i x_{i,k})$, set $\delta_{i,k} = \frac{1}{\sigma_i}\big[f_{i-1}(R_i x_{i,k}) - f_{i-1}(x_{i-1,*})\big]$ and go to Step 4.

Step 3: Taylor step computation. Choose $H_{i,k}$ and compute a step $s_{i,k} \in \mathbb{R}^{n_i}$ that sufficiently reduces the model $m_{i,k}$ given by (2.16) in the sense of (2.21) and such that $x_{i,k} + s_{i,k} \in \mathcal{W}_{i,k}$. Set $\delta_{i,k} = m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k})$.

Step 4: Acceptance of the trial point. Compute $f_i(x_{i,k} + s_{i,k})$ and

$$\rho_{i,k} = \big[f_i(x_{i,k}) - f_i(x_{i,k} + s_{i,k})\big]/\delta_{i,k}. \tag{2.22}$$

If $\rho_{i,k} \geq \eta_1$, then define $x_{i,k+1} = x_{i,k} + s_{i,k}$; otherwise, define $x_{i,k+1} = x_{i,k}$.

Step 5: Termination. Compute $g_{i,k+1}$ and $\chi_{i,k+1}$. If $\chi_{i,k+1} \leq \epsilon_i$ or $x_{i,k+1} \notin \mathcal{A}_i$, then return with the approximate solution $x_{i,*} = x_{i,k+1}$.

Step 6: Trust-Region Update. Set

$$\Delta_{i,k+1} \in \begin{cases} [\Delta_{i,k}, +\infty) & \text{if } \rho_{i,k} \geq \eta_2, \\ [\gamma_2 \Delta_{i,k}, \Delta_{i,k}] & \text{if } \rho_{i,k} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}] & \text{if } \rho_{i,k} < \eta_1, \end{cases} \tag{2.23}$$

and $\mathcal{W}_{i,k+1} = \mathcal{L}_i \cap \mathcal{B}_{i,k+1}$ where

$$\mathcal{B}_{i,k+1} = \{x_{i,k+1} + s \in \mathbb{R}^{n_i} \,|\, \|s\|_\infty \leq \Delta_{i,k+1}\}.$$

Increment k by one and go to Step 1.

Some comments are now necessary for a full understanding of this algorithm.

1. In the initialization step, Δ_i^s is the initial radius of the local trust-region and depends only on the level.

2. The test for the value of i at the beginning of Step 1 is designed to identify the lowest level, at which no further recursion is possible. In this case, a Taylor's iteration is the only choice left.

3. As a result of the discussion preceding (2.14), $x_{i,k+1}$ may not belong to the composite trust region $\mathcal{A}_i$ when the step $s_{i,k}$ is computed by a recursive iteration. However, as indicated above, we wish to limit the length of the step at level $i+1$ to a multiple of the trust-region size. Because of (2.14) and the definition of $\mathcal{A}_i$, we may achieve this objective by stopping our iteration at level i as soon as the iterates leave the composite trust-region $\mathcal{A}_i$. This explains the second termination test in Step 5 of the algorithm and is discussed in detail in Lemma 2.3.3.

4. The difference between the "restriction formulae" (2.8)-(2.10) for the hard bounds and (2.11)-(2.13) for the soft ones makes it necessary to pass both $\mathcal{A}_i$ and $\mathcal{F}_i$ to the algorithm at level i, as it is necessary to compute $\mathcal{L}_i$ at each level independently.

5. The original problem (2.2) is solved by calling RMTR_∞ from a virtual $(r+1)$-rst level at which we assume the trust region to be infinite.

6. If there is only one level ($r = 0$), then RMTR_∞ reduces to the basic trust-region algorithm 2.1.1 described in the previous section.

As usual in trust-region algorithms, iterations at which $\rho_{i,k} \geq \eta_1$ are called *successful* and even *very successful* if $\rho_{i,k} \geq \eta_2$. Otherwise, the step is called unsuccessful. At such iterations, the trial point $x_{i,k} + s_{i,k}$ is accepted as the new iterate and the radius of the corresponding trust region is possibly enlarged. If the iteration is unsuccessful, the trial point is rejected and the radius is reduced.

2.3 Convergence theory

Having motivated our interest in the new method, both as an efficient solver for bound-constrained problems and as an improvement on the existing RMTR algorithm for the unconstrained case, we are now interested in obtaining a theoretical guarantee that RMTR_∞ converges to a first-order critical point of the problem from any starting point. The theory proposed in this section differs significantly from the proof for the RMTR algorithm in Gratton et al. (2008*b*), mostly because of the new criticality measure (imposed by the bounds and the choice of the infinity norm) and because the new algorithm allows for potentially very asymmetric trust regions.

We start by making our assumptions more formal. First, we assume that the Hessians of each f_i and their approximations are bounded above by the constant $\kappa_{\text{H}} \geq 1$, so that, for $i = 0, \ldots, r$,

$$1 + \|\nabla_x^2 f_i(x_i)\|_{\infty,1} \leq \kappa_{\text{H}} \tag{2.24}$$

for all $x_i \in \mathcal{F}_i$ and

$$\beta_{i,k} \leq \kappa_{\rm H} \tag{2.25}$$

for all k, where $\beta_{i,k}$ is as in (2.21). We also assume that all gradients at all levels remain uniformly bounded, which is to say that there exists $\kappa_{\rm g} \geq 1$ such that

$$\|\nabla_x f_i(x_i)\|_1 \leq \kappa_{\rm g} \quad \text{for all} \quad i = 0, \ldots, r, \quad \text{and all} \quad x_i \in \mathcal{F}_i. \tag{2.26}$$

This assumption is not overly restrictive (since $\kappa_{\rm g}$ may depend on n_r) and, for instance, automatically holds by continuity on the feasible set if all iterates $x_{j,\ell}$ remain in a bounded domain, which is the case if both l and u are finite in (2.2). We next assume that the criticality measure $\chi(\cdot)$ satisfies the following level-independent property

$$|\chi(x) - \chi(y)| \leq \kappa_{\rm L} \|x - y\|_\infty \tag{2.27}$$

for all $x, y \in \mathcal{F}$ and also that it satisfies, for all iterations $(i-1, \ell)$ inside a recursive iteration (i, k), the following condition

$$\chi_{i-1,0\ell} = \chi(x_{i-1,0}) \leq 2\kappa_{\rm g} \Delta_{i,k} \qquad \text{for all } \; k, \;\; \text{for all } \; i = 1, \ldots, r. \tag{2.28}$$

These two requirements for the criticality measure are reasonable and satisfied by its most classical definitions, as will be proved in Section 3.4. Notice that this is always true for $i - 1 = r$ since we have assumed Δ_{r+1} is infinite at the virtual level $r + 1$. We now define some additional notation and concepts. We first choose the constant $\kappa_{\rm P} \geq 1$ such that

$$\|P_i\|_\infty \leq \kappa_{\rm P} \qquad \text{for all } \; i = 1, \ldots, r. \tag{2.29}$$

If we choose to go to Step 2 (i.e. we choose to use the function f_{i-1} as a model at iteration (i, k)), we say that this iteration initiates a *minimization sequence* at level $i - 1$, which consists of all successive iterations *at this level* (starting from the point $x_{i-1,0} = R_i x_{i,k}$) until a return is made to level i within iteration (i, k). In this case, we say that iteration (i, k) is the *predecessor* of the minimization sequence at level $i - 1$. If $(i - 1, \ell)$ belongs to this minimization sequence, this is written as $(i, k) = \pi(i-1, \ell)$. We also denote by p_{i-1} the index of the penultimate iterate in the minimization sequence $\{x_{i-1,0}, \ldots, x_{i-1,p_{i-1}}, x_{i-1,*}\}$. Note that (2.15) implies that $\mathcal{W}_{i,k} \subseteq \mathcal{B}_{i,k}$. To each iteration (i, k) at level i, we now associate the set

$$\mathcal{R}(i,k) \stackrel{\rm def}{=} \{(j,\ell) \mid \text{iteration } (j,\ell) \text{ occurs within iteration } (i,k)\}.$$

This set always contains the pair (i, k) and contains only that pair if a Taylor step is used at iteration (i, k). If we choose a recursive step, then it also contains the pairs of level and iteration number of all iterations that occur in the potential recursion started in Step 2 and terminating on return within iteration (i, k), but it does not contain the pairs of indices corresponding to the terminating iterates $(j, *)$ of its internal minimization sequences. It is easy to verify that $j \leq i$ for every j such that $(j, \ell) \in \mathcal{R}(i, k)$ for some non-negative k and ℓ. Note also that $\mathcal{R}(i, k)$ contains at most one minimization sequence at level $i-1$, but may contain more than one at level $i-2$ and below, since each iteration at level $i - 1$ may generate its own. Associated with $\mathcal{R}(i, k)$, we also define

$$\mathcal{T}(i,k) \stackrel{\rm def}{=} \{(j,\ell) \in \mathcal{R}(i,k) \mid (j,\ell) \text{ is a Taylor iteration}\}.$$

The algorithm also ensures the following technical lemma.

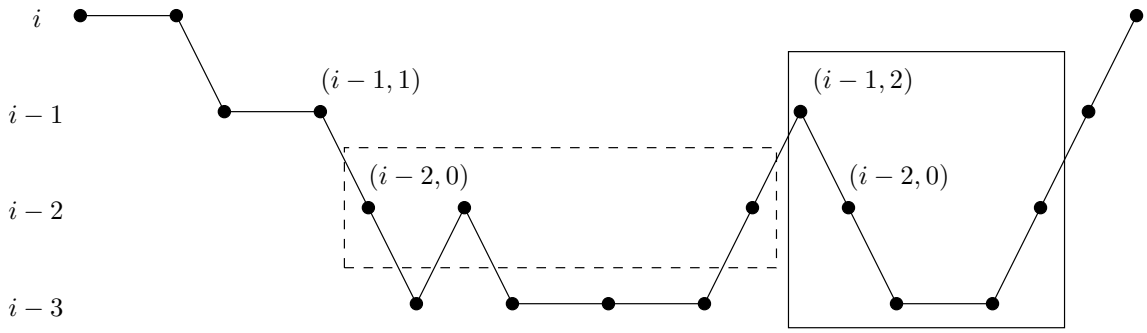


Figure 2.4: Illustration of some multilevel notations. The dashed rectangle area contains a minimization sequence at level $i-2$ initiated at iteration $(i-1,1)$ and the solid line rectangle contains $\mathcal{R}(i-1,2)$.

Lemma 2.3.1 *There exists an* $\epsilon_{\min} \in (0,1]$ *such that, for each iteration* $(i,k) \neq (i,*)$ *(i.e., for all iterates at level* i *but the last one),*

$$\chi_{i,k} \geq \epsilon_{\min}. \tag{2.30}$$

Proof. The inequality (2.20), which is the stopping criteria for minimization at level j, in Step 5 of the algorithm, implies that for all (i,k) and all $(j,\ell) \in \mathcal{R}(i,k)$,

$$\chi_{j,\ell} \geq \epsilon_j = \kappa_\chi \chi_{\pi(j,\ell)} \geq \kappa_\chi \epsilon_{j+1} = \kappa_\chi^2 \chi_{\pi^2(j,\ell)} \geq \cdots \geq \kappa_\chi^{i-j} \chi_{i,k} \geq \cdots \geq \kappa_\chi^r \epsilon_r.$$

This proves (2.30) with $\epsilon_{\min} = \min[1, \kappa_\chi^r \epsilon_r]$. □

We now prove the general version of the Gelman and Mandel's result stating that "bound constraints are preserved" by the prolongation operator.

Lemma 2.3.2 *The definitions (2.9)–(2.10) enforce the inclusion*

$$x_{i,k} + P_i(x_{i-1} - x_{i-1,0}) \in \mathcal{F}_i \qquad \textit{for all } x_{i-1} \in \mathcal{F}_{i-1} \tag{2.31}$$

for $i = 1, \ldots, r$*. As a consequence* $x_{i,k} \in \mathcal{F}_i$ *for all* $i = 0, \ldots, r$ *and all* $k \geq 0$*.*

Proof. For $t = 1, \ldots, n_i$, define $\phi_{i,t} = \sum_{j=1}^{n_{i-1}} |[P_i]_{t,j}|$ and observe that $\phi_{i,t} \leq \|P_i\|_\infty$ for all t. Consider now any $x_{i-1} \in \mathcal{F}_{i-1}$ and the corresponding lower level step

$s_{i-1} = x_{i-1} - x_{i-1,0}$. Then (2.9) and (2.10) imply that

$$\begin{aligned}
&[x_{i,k}]_t + \sum_{j=1}^{n_{i-1}} [P_i]_{tj}[s_{i-1}]_j \\
&\quad = [x_{i,k}]_t + \sum_{j=1,[P_i]_{tj}<0}^{n_{i-1}} |[P_i]_{tj}|(-[s_{i-1}]_j) + \sum_{j=1,[P_i]_{tj}>0}^{n_{i-1}} |[P_i]_{tj}|[s_{i-1}]_j \\
&\quad \geq [x_{i,k}]_t + \sum_{j=1,[P_i]_{tj}<0}^{n_{i-1}} |[P_i]_{tj}| \frac{(-\min_t[x_{i,k} - l_i]_t)}{\|P_i\|_\infty} + \sum_{j=1,[P_i]_{tj}>0}^{n_{i-1}} |[P_i]_{tj}| \frac{\max_t[l_i - x_{i,k}]_t}{\|P_i\|_\infty} \\
&\quad \geq [x_{i,k}]_t + \sum_{j=1,[P_i]_{tj}<0}^{n_{i-1}} |[P_i]_{tj}| \frac{[l_i - x_{i,k}]_t}{\|P_i\|_\infty} + \sum_{j=1,[P_i]_{tj}>0}^{n_{i-1}} |[P_i]_{tj}| \frac{[l_i - x_{i,k}]_t}{\|P_i\|_\infty} \\
&\quad \geq [x_{i,k}]_t + \phi_{i,t} \frac{[l_i - x_{i,k}]_t}{\|P_i\|_\infty} \\
&\quad = \frac{\phi_{i,t}}{\|P_i\|_\infty}[l_i]_t + \left(1 - \frac{\phi_{i,t}}{\|P_i\|_\infty}\right)[x_{i,k}]_t \\
&\quad \geq [l_i]_t
\end{aligned}$$

where the last inequality results from the fact that $[x_{i,k}]_t \geq [l_i]_t$. A similar reasoning gives that

$$[x_{i,k}]_t + \sum_{j=1}^{n_{i-1}} [P_i]_{t,j}[s_{i-1}]_j \leq [u_i]_t$$

for all t, thereby concluding the proof of (2.31). The feasibility of every iterate with respect to the level-dependent bound constraints then results from the fact that all trial points at level i belong to $\mathcal{F}_i$ by construction. □

We next show that the distance from all iterates in a single minimization sequence at level i to the starting point of that sequence is bounded above by a multiple of the trust-region radius at the predecessor's level.

Lemma 2.3.3 *The definitions (2.12)-(2.13) imply that, for* $0 \leq j < r$,

$$\|x - x_{j,0}\|_\infty \leq 2\Delta_{\pi(j,0)} \tag{2.32}$$

for all $x \in \mathcal{L}_j$.

Proof. Consider an $x \in \mathcal{L}_j \subseteq \mathcal{A}_j$. If we now denote the bounds defining the set $\mathcal{S}_{\pi(j,0)} = \mathcal{A}_{j+1} \cap \mathcal{B}_{\pi(j,0)}$ by

$$\overline{v}_{j+1} \stackrel{\text{def}}{=} \max\left[v_{j+1}, x_{\pi(j,0)} - \Delta_{\pi(j,0)}e\right] \quad \text{and} \quad \overline{w}_{j+1} \stackrel{\text{def}}{=} \min\left[w_{j+1}, x_{\pi(j,0)} + \Delta_{\pi(j,0)}e\right],$$

we then verify that

$$\begin{aligned}
[w_j - v_j]_t &= \sum_{u=1,[R_{j+1}]_{tu}>0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{w}_{j+1}]_u + \sum_{u=1,[R_{j+1}]_{tu}<0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{v}_{j+1}]_u \\
&\quad - \sum_{u=1,[R_{j+1}]_{tu}>0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{v}_{j+1}]_u - \sum_{u=1,[R_{j+1}]_{tu}<0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{w}_{j+1}]_u \\
&= \sum_{u=1,[R_{j+1}]_{tu}>0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{w}_{j+1} - \overline{v}_{j+1}]_u + \sum_{u=1,[R_{j+1}]_{tu}<0}^{n_{j+1}} [R_{j+1}]_{tu}[\overline{v}_{j+1} - \overline{w}_{j+1}]_u \\
&\stackrel{\text{def}}{=} [R_{j+1}z(t)]_t,
\end{aligned}$$

where we have used (2.12) and (2.13), and where, for $t = 1, \ldots, n_{j+1}$,

$$[z(t)]_u = \text{sign}([R_{j+1}]_{tu})[\overline{w}_{j+1} - \overline{v}_{j+1}]_u.$$

This last definition implies that $\|z(t)\|_\infty = \|\overline{w}_{j+1} - \overline{v}_{j+1}\|_\infty$ for $t = 1, \ldots, n_{j+1}$. Taking norms and using the identity $\|R_{j+1}\|_\infty = 1$, we therefore obtain that

$$\begin{aligned}
\|w_j - v_j\|_\infty &= \max_t |[R_{j+1}z(t)]_t| \\
&\leq \max_t \|R_{j+1}z(t)\|_\infty \\
&\leq \max_t \|z(t)\|_\infty \\
&= \|\overline{w}_{j+1} - \overline{v}_{j+1}\|_\infty.
\end{aligned} \tag{2.33}$$

Remembering now the definition of $\overline{w}_{j+1}$ and $\overline{v}_{j+1}$, we see that

$$\begin{aligned}
\|\overline{w}_{j+1} - \overline{v}_{j+1}\|_\infty &= \|\min\left[w_{j+1}, x_{\pi(j,0)} + \Delta_{\pi(j,0)}e\right] - \max\left[v_{j+1}, x_{\pi(j,0)} - \Delta_{\pi(j,0)}e\right]\|_\infty \\
&\leq \|\min\left[w_{j+1}, x_{\pi(j,0)} + \Delta_{\pi(j,0)}e\right] - x_{\pi(j,0)}\|_\infty \\
&\quad + \|x_{\pi(j,0)} - \max\left[v_{j+1}, x_{\pi(j,0)} - \Delta_{\pi(j,0)}e\right]\|_\infty \\
&\leq 2\Delta_{\pi(j,0)}.
\end{aligned}$$

Combining now this bound with (2.33) and our assumption that $x \in \mathcal{A}_j$, we obtain that

$$\|x - x_{j,0}\|_\infty \leq \|w_j - v_j\|_\infty \leq 2\Delta_{\pi(j,0)}.$$

□

Our next proposition indicates that, if $\Delta_{i,k}$ becomes too small, then the method reduces, at level i, to the standard trust-region method using Taylor's iterations only.

Lemma 2.3.4 *Assume that, for some iteration (i,k),*

$$\Delta_{i,k} \leq \frac{1}{2}\min\left[1, \frac{\epsilon_{\min}}{2\kappa_g}, \Delta^{\text{s}}_{\min}\right] \stackrel{\text{def}}{=} \kappa_2 \in (0,1), \tag{2.34}$$

where $\Delta^{\text{s}}_{\min} \stackrel{\text{def}}{=} \min_{i=0,\ldots,r} \Delta^{\text{s}}_i$ Then no recursion occurs in iteration (i,k) and $\mathcal{R}(i,k) = \mathcal{T}(i,k) = \{(i,k)\}$.

Proof. Assume that iteration (i,k) is recursive and that iteration $(i-1,0)$ exists. Using (2.34), (2.30) and (2.28) successively, we conclude that

$$\Delta_{i,k} \leq \frac{\epsilon_{\min}}{4\kappa_{\rm g}} \leq \frac{\chi_{i-1,0}}{4\kappa_{\rm g}} \leq \tfrac{1}{2}\Delta_{i,k}$$

which is impossible. Hence our initial assumption that iteration (i,k) is recursive cannot hold and the proof is complete. □

This lemma essentially states that when the trust-region becomes too small compared to the current criticality level, then too little can be gained from lower level iterations to allow recursion. This has the following important consequence.

Lemma 2.3.5 *Consider an iteration (i,k) for which $\chi_{i,k} > 0$ and*

$$\Delta_{i,k} \leq \min\left[\kappa_2, \kappa_3\chi_{i,k}\right], \tag{2.35}$$

where κ_2 is defined in (2.34) and $\kappa_3 \in (0,1)$ is given by

$$\kappa_3 = \min\left[1, \frac{\kappa_{red}(1-\eta_2)}{\kappa_H}\right].$$

Then iteration (i,k) is very successful and $\Delta_{i,k+1} \geq \Delta_{i,k}$.

Proof. Because of (2.34) and Lemma 2.3.4, we know that iteration (i,k) is a Taylor iteration. Thus, using (2.21), and the definition of $\delta_{i,k}$ in Step 3 of the algorithm,

$$\delta_{i,k} \geq \kappa_{\rm red}\chi_{i,k}\min\left[1, \frac{\chi_{i,k}}{\beta_{i,k}}, \Delta_{i,k}\right].$$

But, because $\kappa_{\rm red} \in (0, \frac{1}{2})$ and thus $\kappa_{\rm red}(1-\eta_2) \leq 1$, and also because of (2.25), (2.35) implies that $\Delta_{i,k} \leq \min\left[1, \frac{\chi_{i,k}}{\beta_{i,k}}\right]$ and hence that

$$\delta_{i,k} \geq \kappa_{\rm red}\chi_{i,k}\Delta_{i,k}. \tag{2.36}$$

We now observe that the mean-value theorem, (2.16) and the definition of $g_{i,k}$ ensure that

$$f_i(x_{i,k}+s_{i,k}) - m_{i,k}(x_{i,k}+s_{i,k}) = \tfrac{1}{2}\langle s_{i,k}, [\nabla_x^2 f_i(\xi_{i,k}) - H_{i,k}]s_{i,k}\rangle$$

for some $\xi_{i,k} \in [x_{i,k}, x_{i,k}+s_{i,k}]$, and thus using (2.24), (2.25), the inequality $|\langle u,v\rangle| \leq \|u\|_1\|v\|_\infty$ and the bound $\|s_{i,k}\|_\infty \leq \Delta_{i,k}$, we obtain that

$$\begin{aligned}
|f_i(x_{i,k}+s_{i,k}) - m_{i,k}(x_{i,k}+s_{i,k})| &\leq \frac{1}{2}\|(\nabla_x^2 f_i(\xi_{i,k}) - H_{i,k})s_{i,k}\|_1\|s_{i,k}\|_\infty \\
&\leq \frac{1}{2}\|(\nabla_x^2 f_i(\xi_{i,k}) - H_{i,k})\|_{\infty,1}\|s_{i,k}\|_\infty^2 \\
&\leq \frac{1}{2}[\,\|\nabla_x^2 f_i(\xi_{i,k})\|_{\infty,1} + \|H_{i,k}\|_{\infty,1}\,]\|s_{i,k}\|_\infty^2 \\
&\leq \kappa_{\rm H}\Delta_{i,k}^2.
\end{aligned}$$

Combining now (2.35), (2.36) and this last inequality, we verify that

$$|\rho_{i,k} - 1| \leq \left|\frac{f_i(x_{i,k}+s_{i,k}) - m_{i,k}(x_{i,k}+s_{i,k})}{\delta_{i,k}}\right| \leq \frac{\kappa_{\rm H}}{\kappa_{\rm red}\chi_{i,k}}\Delta_{i,k} \leq 1-\eta_2.$$

Thus iteration (i,k) must be very successful and, because of (2.23), the trust-region radius cannot decrease. □

This last result implies the following useful consequence.

Lemma 2.3.6 *Each minimization sequence contains at least one successful iteration.*

Proof. This follows from the fact that unsuccessful iterations cause the trust-region radius to decrease, until (2.35) is eventually satisfied and a (very) successful iteration occurs because of Lemma 2.3.5. □

The attentive reader will have noticed that the term in $\Delta^{\rm s}_{\rm min}$ in the minimum defining κ_2 in (2.34) has not been used in Lemma 2.3.4. This term is however crucial in the following further consequence of (2.34).

Lemma 2.3.7 *For every iteration (j,ℓ), with $j = 0,\ldots,r$ and $\ell > 0$, we have that*

$$\Delta_{j,\ell} \geq \Delta_{\rm min} \stackrel{\rm def}{=} \gamma_1 \min[\kappa_2, \kappa_3 \epsilon_{\rm min}]. \tag{2.37}$$

Proof. Suppose that (j,ℓ) is the first iteration such that

$$\Delta_{j,\ell} < \gamma_1 \min[\kappa_2, \kappa_3 \epsilon_{\rm min}]. \tag{2.38}$$

Since $\gamma_1 < 1$ and $\kappa_2 \leq \Delta^{\rm s}_{\rm min}$, we then obtain that

$$\Delta_{j,0} = \Delta^{\rm s}_j \geq \Delta^{\rm s}_{\rm min} > \gamma_1 \Delta^{\rm s}_{\rm min} \geq \gamma_1 \min[\kappa_2, \kappa_3 \epsilon_{\rm min}],$$

and, because of (2.38), we have that $\ell > 0$. This in turn implies that $\Delta_{j,\ell}$ is computed using Step 6 of the algorithm. But, the mechanism of the algorithm imposes that $\Delta_{j,\ell} \geq \gamma_1 \Delta_{j,\ell-1}$ an thus (2.38) also yields that

$$\Delta_{j,\ell-1} < \min[\kappa_2, \kappa_3 \epsilon_{\rm min}] \leq \min[\kappa_2, \kappa_3 \chi_{j,\ell-1}],$$

where we have used Lemma 2.3.1 and the fact that $(j,\ell-1) \neq (j,*)$ to derive the last inequality. Hence, we may apply Lemma 2.3.5 to conclude that iteration $(j,\ell-1)$ is very successful and that $\Delta_{j,\ell} \geq \Delta_{j,\ell-1}$. Thus, iteration (j,ℓ) cannot be the first such that (2.38) holds. This implies that (2.38) is impossible, which completes the proof. □

We next show the crucial result that the algorithm is well defined, and that all the recursions are finite.

Theorem 2.3.8 *The number of iterations in each level is finite. Moreover, there exists $\kappa_h \in (0,1)$ such that, for every minimization sequence at level $i = 0,\ldots,r$ and every $t \geq 0$,*

$$f_i(x_{i,0}) - f_i(x_{i,t+1}) \geq \tau_{i,t} \mu^{i+1} \kappa_h,$$

where $\tau_{i,t}$ is the total number of successful Taylor iterations in $\displaystyle\bigcup_{\ell=0}^{t} \mathcal{R}(i,\ell)$ and $\mu = \eta_1/\sigma_{\rm max}$ with $\sigma_{\rm max} = \max[1, \max_{i=1,\ldots,r} \sigma_i]$.

Proof. We will show this by induction on the levels, starting from level 0. First, let us define $\omega_{i,t}$ as the number of successful Taylor iterations in $\mathcal{R}(i,t)$. Thus,

$$\tau_{i,t} = \sum_{\ell=0}^{t} \omega_{i,\ell}.$$

Note that, if iteration (i,ℓ) is successful, then $\omega_{i,\ell} \geq 1$.

Consider first a minimization sequence started at level 0, and assume without loss of generality, that it belongs to $\mathcal{R}(r,k)$ for some $k \geq 0$. Every iteration in this minimization sequence has to be a Taylor iteration, which implies that the sufficient decrease condition (2.21) is satisfied, and in particular, for all successful iterations,

$$\begin{aligned} f_0(x_{0,\ell}) - f_0(x_{0,\ell+1}) &\geq \eta_1 \delta_{0,\ell} \geq \eta_1 \kappa_{\rm red} \chi_{0,\ell} \min\left[1, \frac{\chi_{0,\ell}}{\beta_{0,\ell}}, \Delta_{0,\ell}\right] \\ &\geq \omega_{0,\ell} \eta_1 \kappa_{\rm red} \epsilon_{\min} \min\left[1, \frac{\epsilon_{\min}}{\kappa_{\rm H}}, \Delta_{\min}\right] \end{aligned} \tag{2.39}$$

where we used Lemma 2.3.7, (2.25), (2.30) and the fact that $\omega_{0,\ell} = 1$ for every successful iteration $(0,\ell)$, since $\mathcal{R}(0,\ell) = \{(0,\ell)\}$. Since we know from Lemma 2.3.6 that every minimization sequence has at least one successful iteration, we can sum up the reductions obtained at level 0, which gives us

$$f_0(x_{0,0}) - f_0(x_{0,t+1}) = \sum_{\ell=0}^{t}{}^{(\rm S)} [f_0(x_{0,\ell}) - f_0(x_{0,\ell+1})] \geq \tau_{0,t} \eta_1 \kappa_{\rm h} \geq \tau_{0,t} \mu \kappa_{\rm h} \tag{2.40}$$

where the superscript (S) indicates that the sum is restricted to successful iterations and where

$$\kappa_{\rm h} \stackrel{\rm def}{=} \kappa_{\rm red} \epsilon_{\min} \min\left[1, \frac{\epsilon_{\min}}{\kappa_{\rm H}}, \Delta_{\min}\right] = \kappa_{\rm red} \epsilon_{\min} \min\left[\frac{\epsilon_{\min}}{\kappa_{\rm H}}, \Delta_{\min}\right], \tag{2.41}$$

where the last equality results from the inequalities $\epsilon_{\min} \leq 1$ and $\kappa_{\rm H} \geq 1$. If $r = 0$, since $f_0 = f$ is bounded below by assumption, then (2.40) implies that $\tau_{0,t}$ is finite. If $r > 0$, f_0 is continuous, and thus it is bounded below on the set $\{x \in \mathbb{R}^{n_0} | \|x - x_{0,0}\|_\infty \leq 2\Delta_{r,k}\}$, and again, $\tau_{0,t}$ has to be finite. Since $\tau_{0,t}$ accounts for all successful iterations in the minimization sequence, we obtain that there must be a last finite successful iteration $(0,p_0)$. For the purpose of obtaining a contradiction, let us assume that the sequence is infinite. Then, all iterations $(0,\ell)$ would be unsuccessful for $\ell > p_0$, causing $\Delta_{0,\ell}$ to converge to zero, which is impossible in view of Lemma 2.3.7. Hence, the minimization sequence is finite. The same reasoning may be applied to every such sequence at level 0.

Now, consider an arbitrary minimization sequence at level i within $\mathcal{R}(r,k)$ for some $k > 0$, and assume that each minimization sequence at level $i-1$ is finite and also that each successful iteration $(i-1,u)$ in every minimization sequence at this lower level satisfies

$$f_{i-1}(x_{i-1,u}) - f_{i-1}(x_{i-1,u+1}) \geq \omega_{i-1,u} \mu^i \kappa_{\rm h}. \tag{2.42}$$

Consider a successful iteration (i,ℓ), whose existence is ensured by Lemma 2.3.6. If it is a Taylor iteration, we obtain that

$$f_i(x_{i,\ell}) - f_i(x_{i,\ell+1}) \geq \eta_1 \kappa_{\rm h} \geq \mu^{i+1} \kappa_{\rm h} = \omega_{i,\ell} \mu^{i+1} \kappa_{\rm h}, \tag{2.43}$$

since $\eta_1 \in (0,1)$, $\sigma_{\max} > 1$ and $\omega_{i,\ell} = 1$ for every successful Taylor iteration (i,ℓ). If, on the other hand, iteration (i,ℓ) uses Step 2, then we obtain that

$$\begin{aligned} f_i(x_{i,\ell}) - f_i(x_{i,\ell+1}) &\geq \frac{\eta_1}{\sigma_i}\left[f_{i-1}(x_{i-1,0}) - f_{i-1}(x_{i-1,*})\right] \\ &\geq \mu \sum_{u=0}^{p_{i-1}^{(\mathrm{S})}} \left[f_{i-1}(x_{i-1,u}) - f_{i-1}(x_{i-1,u+1})\right]. \end{aligned}$$

Since $\omega_{i,\ell} = \tau_{i-1,p_{i-1}}$, the definition of $\tau_{i-1,t}$ and (2.42) give that

$$f_i(x_{i,\ell}) - f_i(x_{i,\ell+1}) \geq \mu^{i+1}\kappa_{\mathrm{h}} \sum_{u=0}^{p_{i-1}} \omega_{i-1,u} = \tau_{i-1,p_{i-1}}\mu^{i+1}\kappa_{\mathrm{h}} = \omega_{i,\ell}\mu^{i+1}\kappa_{\mathrm{h}}. \tag{2.44}$$

Combining (2.43) and (2.44), we see that (2.42) again holds at level i instead of $i-1$. Moreover, as above,

$$f_i(x_{i,0}) - f_i(x_{i,t+1}) = \sum_{\ell=0}^{t^{(\mathrm{S})}} \left[f_i(x_{i,\ell}) - f_i(x_{i,\ell+1})\right] \geq \tau_{i,t}\mu^{i+1}\kappa_{\mathrm{h}}, \tag{2.45}$$

for the minimization sequence including iteration (i,ℓ). If $i = r$, $f_i = f$ is bounded below by assumption and (2.45) imposes that the number of successful iterations in this sequence must again be finite. The same conclusion holds if $i < r$, since f_i is continuous and hence bounded below on the set $\{x \in \mathbb{R}^{n_i} | \|x - x_{i,0}\|_\infty \leq 2\Delta_{r,k}\}$ which contains $x_{i,t+1}$ because of Lemma 2.3.3. As for level 0, we may then conclude that the number of iterations (both successful and unsuccessful) in the minimization sequence is finite. Moreover, the same reasoning holds for every minimization sequence at level i, and the induction is complete. □

Corollary 2.3.9 *Assume that f is bounded below by f_{low}. Then Algorithm RMTR$_\infty$ needs at most*

$$\left\lceil \frac{f(x_{r,0}) - f_{\mathrm{low}}}{\theta(\epsilon_{\min})} \right\rceil \tag{2.46}$$

successful Taylor iterations at any level to obtain an iterate $x_{r,k}$ such that $\chi_{r,k} < \epsilon_r$, where

$$\theta(\epsilon) = \mu^{r+1}\kappa_{red}\epsilon \min\left[\frac{\epsilon}{\kappa_H}, \gamma_1 \min\left[\kappa_2, \kappa_3\epsilon\right]\right]. \tag{2.47}$$

Proof. The desired bound directly follows from Theorem 2.3.8, (2.41), (2.37) and the definition of $\epsilon_{\min}$. □

This complexity result for general nonconvex problems is similar to Corollary 3.8 in Gratton et al. (2008*b*), and may also be very pessimistic. It is of the same order ϵ^2 as the corresponding bound for the pure gradient method (see (Nesterov 2004), page 29). This is not surprising given that it is based on the Cauchy condition, which itself results from a step in the steepest-descent direction. Note that the bound is in terms of iteration numbers, and only implicitly accounts for the cost of computing a Taylor step satisfying (2.21). As was the case for the Euclidean norm, this suggests several comments.

1. The bound (2.46) is expressed in terms of the number of successful Taylor iterations, that is successful iterations where the trial step is computed without resorting to further recursion. This provides an adequate measure of the linear algebra effort for all successful iterations, since successful iterations using the recursion of Step 2 cost little beyond the evaluation of the level-dependent objective function and its gradient. Moreover, the number of such iterations is, by construction, at most equal to r times that of Taylor iterations (in the worst case where each iteration at level r includes a full recursion to level 0 with a single successful iteration at each level $j > 0$). Hence the result shows that the number of necessary successful iterations, all levels included, is of order $1/\epsilon^2$ for small values of ϵ. This order is not qualitatively altered by the inclusion of unsuccessful iterations either, provided we replace the very successful trust-region radius update (top case in (2.23)) by

$$\Delta_{i,k}^{+} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}] \quad \text{if} \quad \rho_{i,k} \geq \eta_2,$$

   for some $\gamma_3 > 1$. Indeed, Lemma 2.3.7 imposes that the decrease in radius caused by unsuccessful iterations must asymptotically be compensated by an increase at successful ones. This is to say that, if α is the average number of unsuccessful iterations per successful one at any level, then one must have that $\gamma_3 \gamma_2^{\alpha} \geq 1$, and therefore that $\alpha \leq -\log(\gamma_3)/\log(\gamma_2)$. Thus the complexity bound in $1/\epsilon^2$ for small ϵ is only modified by a constant factor if all iterations (successful and unsuccessful) are considered. This therefore also gives a worst case upper bound on the number of function and gradient evaluations.

2. Moreover, (2.46) involves the number of successful Taylor iterations *summed up on all levels* (as a result of Theorem 2.3.8). Thus such successful iterations at cheap low levels decrease the number of necessary expensive ones at higher levels, and the multilevel algorithm requires (at least in the theoretical worst case) fewer Taylor iterations at the upper level than the single-level variant. This provides theoretical backing for the practical observation that the structure of multilevel bound-constrained optimization problems can be used to advantage.

3. The definition of $\theta(\epsilon)$ in (2.47) is interesting in that it does not depend on the problem dimension, but rather on the properties of the problem or of the algorithm itself. Thus, if we consider the case where different levels correspond to different discretization meshes and make the mild assumption that r, κ_{H} and κ_{g} are uniformly bounded above (i.e., by a constant independant of the dimension n_r), we deduce that our complexity bound is mesh-independent. Nevertheless, notice that this hypothesis is definetly more restrictive than those needed to prove that the algorithm is globally convergent.

A second important consequence of Theorem 2.3.8 is that the algorithm is globally convergent, in the sense that, if ϵ_r is "driven to zero", it generates a subsequence of iterates that are asymptotically first-order critical. More specifically, we examine the sequence of iterates $\{x_{r,k}\}$ generated as follows. We consider, at level r, a sequence of tolerances $\{\epsilon_{r,j}\} \in (0,1)$ monotonically converging to zero, start the algorithm with $\epsilon_r = \epsilon_{r,0}$ and alter slightly the mechanism of Step 5 (at level r only) to reduce ϵ_r from $\epsilon_{r,j}$ to $\epsilon_{r,j+1}$ as soon as $\chi_{r,k+1} \leq \epsilon_{r,j}$. The calculation is then continued with this more stringent threshold until it is also attained, ϵ_r^{g} is then again reduced and so on.

Theorem 2.3.10 *Assume that ϵ_r is "driven to zero" in Algorithm RMTR$_\infty$. Then*

$$\liminf_{k\to\infty} \chi_{r,k} = 0. \tag{2.48}$$

Proof. Since $\Delta_{r+1,0} = \infty$ ensures that $\mathcal{L}_r = \mathcal{F}_r$, Lemma 2.3.2 implies that each successive minimization at level r can only stop at iteration k if

$$\chi_{r,k+1} \leq \epsilon_{r,j}. \tag{2.49}$$

Theorem 2.3.8 then implies that there are only finitely many successful iterations between two reductions of ϵ_r. We therefore obtain that for each $\epsilon_{r,j}$ there is an arbitrarily large k such that (2.49) holds. The desired result then follows immediately from our assumption that $\{\epsilon_{r,j}\}$ converges to zero. □

Of course, the interest of this result is mostly theoretical, since most practical applications of Algorithm RMTR$_\infty$ consider a nonzero gradient tolerance ϵ_r.

Observe that our definition of ϵ_i in (2.20) implies that, if ϵ_r is driven to zero, then so is $\epsilon_i = \kappa_\chi^{r-i}\epsilon_r$. As for the Euclidean case, and assuming the trust region becomes asymptotically inactive at every level (as is most often the case in practice), each minimization sequence in the algorithm becomes infinite (as if it were initiated with a zero gradient threshold and an infinite initial radius). Recursion to lower levels then remains possible for arbitrarily small gradients, and may therefore occur arbitrarily far in the sequence of iterates. Moreover, we may still apply Theorem 2.3.10 at each level and deduce that, if the trust region becomes asymptotically inactive,

$$\liminf_{k\to\infty} \chi_{i,k} = 0 \tag{2.50}$$

for all $i = 0, \ldots, r$.

As is the case for single-level trust-region algorithms, we now would like to prove that the limit inferior in (2.48) and (2.50) can be replaced by a true limit. This requires the notion of a *recursively successful iteration*. We say that iteration $(j,\ell) \in \mathcal{R}(i,k)$ is *recursively successful for* (i,k) whenever iterations $(j,\ell), \pi(j,0)$, $\pi^2(j,0), \ldots, \pi^{i-j}(j,0) = (i,k)$ are all successful. This is to say that the decrease in the objective function obtained at iteration (j,ℓ) effectively contributes to the reduction obtained at iteration (i,k). We start by stating a result on the relative sizes of the objective function decreases in the course of a recursive iteration.

Lemma 2.3.11 *Assume that some iteration $(j,\ell) \in \mathcal{R}(i,k)$ is recursively successful for (i,k). Then*

$$f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) \leq f_j(x_{j,0}) - f_j(x_{j,*}) \leq \mu^{j-i}\,[\, f_i(x_{i,k}) - f_i(x_{i,k+1})\,]. \tag{2.51}$$

Proof. The first inequality immediately results from the monotonicity of the sequence of objective function values in a minimization sequence. To prove the second inequality, consider iteration $(j+1,q) = \pi(j,0)$. Then

$$f_j(x_{j,0}) - f_j(x_{j,*}) = \sigma_{j+1}\delta_{j+1,q} \leq \eta_1^{-1}\sigma_{\max}\,[\, f_{j+1}(x_{j+1,q}) - f_{j+1}(x_{j+1,q+1})\,]$$

where we used the definition of $\delta_{j+1,q}$, the definition of $\sigma_{\max}$ and the fact that iteration $(j+1,q)$ must be successful since (j,ℓ) is recursively successful for (i,k). But this

argument may now be repeated at level $j+2, \ldots, i$, yielding the desired bound, given that $\mu = \eta_1/\sigma_{\max} < 1$. □

This lemma then allows us to express a simple relation between the size of Taylor steps at recursively successful iterations and the associated objective decrease.

Lemma 2.3.12 *Assume that the Taylor iteration $(j,\ell) \in \mathcal{R}(i,k)$ is recursively successful for (i,k) and that, for some $\epsilon \in (0,1)$,*

$$\chi_{j,\ell} \geq \epsilon \tag{2.52}$$

and

$$f_i(x_{i,k}) - f_i(x_{i,k+1}) < \frac{\mu^r \eta_1 \kappa_{red} \epsilon^2}{\kappa_H}. \tag{2.53}$$

Then

$$\|x_{j,\ell} - x_{j,\ell+1}\|_\infty \leq \frac{1}{\kappa_{red}\eta_1\epsilon} [\, f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) \,]. \tag{2.54}$$

Proof. We know from (2.21), (2.25), (2.52) and the successful nature of iteration (j,ℓ) that

$$\begin{aligned} f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) &\geq \eta_1\kappa_{\rm red}\chi_{j,\ell} \min\left[\tfrac{\chi_{j,\ell}}{\kappa_{\rm H}}, \Delta_{j,\ell}, 1\right] \\ &\geq \eta_1\kappa_{\rm red}\epsilon \min\left[\tfrac{\epsilon}{\kappa_{\rm H}}, \Delta_{j,\ell}, 1\right] \\ &= \eta_1\kappa_{\rm red}\epsilon \min\left[\tfrac{\epsilon}{\kappa_{\rm H}}, \Delta_{j,\ell}\right] \end{aligned} \tag{2.55}$$

where we used (2.25) and the inequality $\epsilon < 1$ to deduce the last equality. But Lemma 2.3.11 gives that

$$\begin{aligned} f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) &\leq \mu^{j-i} [\, f_i(x_{i,k}) - f_i(x_{i,k+1}) \,] \\ &\leq \mu^{-r} [\, f_i(x_{i,k}) - f_i(x_{i,k+1}) \,] \\ &\leq \frac{\eta_1\kappa_{\rm red}\epsilon^2}{\kappa_{\rm H}}, \end{aligned}$$

where we used (2.53) to deduce the last inequality. Hence we see that only the second term in the last minimum of (2.55) can be active, which gives that

$$f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) \geq \eta_1\kappa_{\rm red}\epsilon\Delta_{j,\ell}.$$

We then obtain (2.54) from the observation that $x_{j,\ell+1} = x_{j,\ell} + s_{j,\ell} \in \mathcal{W}_{j,\ell} \subseteq \mathcal{B}_{j,\ell}$. □

We next prove the following useful technical lemma.

Lemma 2.3.13 *Assume that a minimization sequence at level j $(0 \leq j \leq r)$ is such that*

$$\chi_{j,0} \geq \epsilon_{\rm ncr} \tag{2.56}$$

for some $\epsilon_{\rm ncr} \in (0,1)$, but also that

$$\|s_{j,\ell}\|_\infty \leq \kappa_{\rm ncr} [\, f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) \,] \tag{2.57}$$

for some $\kappa_{\rm ncr} > 0$ as long as iteration (j,ℓ) is successful and $\chi_{j,\ell} \geq \frac{1}{2}\epsilon_{\rm ncr}$. Assume finally that

$$f_j(x_{j,0}) - f_j(x_{j,*}) \leq \frac{\epsilon_{\rm ncr}}{2\kappa_{\rm ncr}\kappa_{\rm L}}. \tag{2.58}$$

Then $\chi_{j,\ell} \geq \frac{1}{2}\epsilon_{\rm ncr}$ and (2.57) holds for all $\ell \geq 0$.

Proof. Assume that there exists a (first) successful iteration (j, s) such that

$$\chi_{j,s} < \tfrac{1}{2}\epsilon_{\rm ncr}, \tag{2.59}$$

which implies that $\chi_{j,\ell} \geq \frac{1}{2}\epsilon_{\rm ncr}$ for all $0 \leq \ell < s$. We now use (2.57) and the triangle inequality, and sum on all successful iterations (at level j) from 0 to $s-1$, yielding

$$\|x_{j,0} - x_{j,s}\|_\infty \leq \sum_{\ell=0}^{s-1}{}^{(S)} \|x_{j,\ell} - x_{j,\ell+1}\|_\infty \leq \kappa_{\rm ncr}\,[\,f_j(x_{j,0}) - f_j(x_{j,s})\,]. \tag{2.60}$$

Applying now 2.27, the monotonicity of f_j within the minimization sequence, and (2.58), we obtain from (2.60) that

$$\begin{aligned} |\chi_{j,0} - \chi_{j,s}| &\leq \kappa_{\rm ncr}\kappa_{\rm L}\,[\,f_j(x_{j,0}) - f_j(x_{j,s})\,] \\ &\leq \kappa_{\rm ncr}\kappa_{\rm L}\,[\,f_j(x_{j,0}) - f_j(x_{j,*})\,] \\ &\leq \tfrac{1}{2}\epsilon_{\rm ncr}. \end{aligned}$$

But this last inequality is impossible since we know from (2.56) and (2.59) that $\chi_{j,0} - \chi_{j,s} > \frac{1}{2}\epsilon_{\rm ncr}$. Hence our assumption (2.59) is itself impossible and we obtain that, for all $\ell \geq 0$, $\chi_{j,\ell} \geq \frac{1}{2}\epsilon_{\rm ncr}$. This and the lemma's assumptions then ensure that (2.57) also holds for all $j \geq 0$. □

We now consider the case of recursive iterations.

Lemma 2.3.14 *Assume that, for some recursive successful iteration (i, k),*

$$\chi_{i,k} \geq \epsilon_{\rm rsi} \tag{2.61}$$

and

$$f_i(x_{i,k}) - f_i(x_{i,k+1}) \leq \frac{\kappa_\chi \epsilon_{\rm rsi}}{2\kappa_{\rm rsi}\kappa_{\rm L}} \tag{2.62}$$

for some $\epsilon_{\rm rsi} \in (0,1)$ and some $\kappa_{\rm rsi} > 0$. Assume also that

$$\|s_{i-1,\ell}\|_\infty \leq \kappa_{\rm rsi}\,[\,f_{i-1}(x_{i-1,\ell}) - f_{i-1}(x_{i-1,\ell+1})\,] \tag{2.63}$$

for all (recursively) successful iterations in the minimization sequence initiated at level $i-1$ by iteration (i, k) as long as

$$\chi_{i-1,\ell} \geq \tfrac{1}{2}\kappa_\chi \epsilon_{\rm rsi}. \tag{2.64}$$

Then

$$\|s_{i,k}\|_\infty \leq \mu^{-1}\kappa_P\kappa_{\rm rsi}\,[\,f_i(x_{i,k}) - f_i(x_{i,k+1})\,]. \tag{2.65}$$

Proof. Consider the minimization sequence initiated at level $i-1$ by iteration (i, k). Because of (2.19) and (2.61), we have that $\chi_{i-1,0} \geq \kappa_\chi\epsilon_{\rm rsi}$. We may now apply Lemma 2.3.13 with $\epsilon_{\rm ncr} = \kappa_\chi\epsilon_{\rm rsi}$ and $\kappa_{\rm ncr} = \kappa_{\rm rsi}$, given that (2.62) ensures (2.58). As a result, we know that $\chi_{i-1,\ell} \geq \frac{1}{2}\kappa_\chi\epsilon_{\rm rsi}$ and (2.63) hold for all successful iterations $(i-1,\ell)$ $(\ell \geq 0)$. Using the triangle inequality and summing on all successful iterations at level $i-1$, we find that

$$\|x_{i-1,0} - x_{i-1,*}\|_\infty \leq \sum_{\ell=0}^{p_{i-1}}{}^{(S)} \|x_{i-1,\ell} - x_{i-1,\ell+1}\|_\infty \leq \kappa_{\rm rsi}\,[\,f_{i-1}(x_{i-1,0}) - f_{i-1}(x_{i-1,*})\,].$$

This inequality, the definition of $s_{i,k}$, (2.29) and Lemma 2.3.11 in turn imply that

$$\begin{aligned} \|s_{i,k}\|_\infty &\leq \|P_i\|_\infty \|x_{i-1,0} - x_{i-1,*}\|_\infty \\ &\leq \kappa_{\rm P}\kappa_{\rm rsi}\,[\,f_{i-1}(x_{i-1,0}) - f_{i-1}(x_{i-1,*})\,] \\ &\leq \mu^{-1}\kappa_{\rm P}\kappa_{\rm rsi}\,[\,f_i(x_{i,k}) - f_i(x_{i,k+1})\,]. \end{aligned}$$

□

Our next step is to consider the cumulative effect of all the complete recursion for an iteration at the finest level.

Lemma 2.3.15 *Assume that, for some successful iteration (r,k) $(k \geq 0)$,*

$$\chi_{r,k} \geq \epsilon \tag{2.66}$$

and

$$f(x_{r,k}) - f(x_{r,k+1}) < \frac{\eta_1\kappa_{red}(\frac{1}{2}\kappa_\chi)^{2r}\epsilon^2}{2\kappa_{\rm L}} \tag{2.67}$$

for some $\epsilon \in (0,1)$. Then

$$\|s_{r,k}\|_\infty \leq \kappa_{\rm acc}\,[\,f(x_{r,k}) - f(x_{r,k+1})\,], \tag{2.68}$$

where

$$\kappa_{\rm acc} \stackrel{\rm def}{=} \left(\frac{\kappa_P}{\mu}\right)^r \frac{1}{\kappa_{red}\eta_1(\frac{1}{2}\kappa_\chi)^r\epsilon}. \tag{2.69}$$

Proof. Assume that (2.66) and (2.67) hold at the successful iteration (r,k) and consider the subset of iterations given by $\mathcal{R}(r,k)$. If (r,k) is a Taylor iteration, then $\mathcal{R}(r,k) = \{(r,k)\}$ and the desired result follows from Lemma 2.3.12 and the inequality

$$\frac{1}{\kappa_{\rm red}\eta_1\epsilon} \leq \kappa_{\rm acc}. \tag{2.70}$$

If iteration (r,k) is recursive, consider a minimization sequence containing a recursively successful iteration for (r,k) at the deepest possible level in $\mathcal{R}(r,k)$. Let the index of this deepest level be d and note that every successful iteration in this minimization sequence must be recursively successful for (r,k). We will now prove the result by induction on the levels, from $d+1$ up to r. First, let $(d+1,q) = \pi(d,0)$ and assume that

$$\chi_{d+1,q} \geq (\tfrac{1}{2}\kappa_\chi)^{r-d-1}\epsilon, \tag{2.71}$$

which gives, in view of (2.19), that $\chi_{d,0} \geq (\frac{1}{2})^{r-d-1}\kappa_\chi^{r-d}\epsilon$. Each (recursively) successful iteration of our deepest minimization sequence must thus be a Taylor iteration. Because of Lemma 2.3.12, we then obtain that, as long as $\chi_{d,\ell} \geq (\frac{1}{2}\kappa_\chi)^{r-d}\epsilon$ and iteration (d,ℓ) is successful, we have that

$$\|s_{d,\ell}\|_\infty = \|x_{d,\ell} - x_{d,\ell+1}\|_\infty \leq \frac{1}{\kappa_{\rm red}\eta_1(\frac{1}{2}\kappa_\chi)^{r-d}\epsilon}\,[\,f_d(x_{d,\ell}) - f_d(x_{d,\ell+1})\,], \tag{2.72}$$

We could then apply Lemma 2.3.14 for iteration $(d+1,q) = \pi(d,0)$ with

$$\epsilon_{\rm rsi} = (\tfrac{1}{2}\kappa_\chi)^{r-d-1}\epsilon \quad \text{and} \quad \kappa_{\rm rsi} = \frac{1}{\kappa_{\rm red}\eta_1(\frac{1}{2}\kappa_\chi)^{r-d}\epsilon},$$

if (2.62) holds. But note that Lemma 2.3.11 implies that

$$f_{d+1}(x_{d+1,q}) - f_{d+1}(x_{d+1,q+1}) \leq \mu^{d+1-r} \left[f(x_{r,k}) - f(x_{r,k+1}) \right]$$

which in turn gives (2.62) in view of (2.67), as desired. As a result of Lemma 2.3.14, we then deduce that

$$\begin{aligned} \|s_{d+1,q}\|_\infty &\leq \mu^{-1} \kappa_{\rm P} \kappa_{\rm rsi} \left[f_{d+1}(x_{d+1,q}) - f_{d+1}(x_{d+1,q+1}) \right] \\ &= \left(\frac{\kappa_{\rm P}}{\mu} \right) \frac{1}{\kappa_{\rm red} \eta_1 (\frac{1}{2}\kappa_\chi)^{r-d} \epsilon} \left[f_{d+1}(x_{d+1,q}) - f_{d+1}(x_{d+1,q+1}) \right]. \end{aligned} \tag{2.73}$$

Consider now a minimization sequence at level j such that $d < j < r$, and such that this minimization sequence belongs to $\mathcal{R}(r,k)$. Then define $(j+1,t) = \pi(j,0)$ and assume, in line with (2.71), that $\chi_{j+1,t} \geq (\frac{1}{2}\kappa_\chi)^{j-1}\epsilon$ which yields in particular that $\chi_{j,0} \geq (\frac{1}{2})^{j-1}\kappa_\chi^j \epsilon$. Assume now that

$$\chi_{j,\ell} \geq (\tfrac{1}{2}\kappa_\chi)^j \epsilon, \tag{2.74}$$

that iteration (j,ℓ) is (recursively) successful, and that

$$\|s_{j,\ell}\|_\infty \leq \left(\frac{\kappa_{\rm P}}{\mu} \right)^j \frac{1}{\kappa_{\rm red} \eta_1 (\frac{1}{2}\kappa_\chi)^j \epsilon} \left[f_j(x_{j,\ell}) - f_j(x_{j,\ell+1}) \right].$$

Applying Lemma 2.3.11 and using (2.67), we may then apply Lemma 2.3.14 for iteration $(j+1,t)$, with

$$\epsilon_{\rm rsi} = (\tfrac{1}{2}\kappa_\chi)^{j-1}\epsilon \quad \text{and} \quad \kappa_{\rm rsi} = \left(\frac{\kappa_{\rm P}}{\mu} \right)^j \frac{1}{\kappa_{\rm red} \eta_1 (\frac{1}{2}\kappa_\chi)^j \epsilon}.$$

This ensures that

$$\begin{aligned} \|s_{j+1,t}\|_\infty &\leq \mu^{-1} \kappa_{\rm P} \kappa_{\rm rsi} \left[f_{j+1}(x_{j+1,t}) - f_{j+1}(x_{j+1,t+1}) \right] \\ &= \left(\frac{\kappa_{\rm P}}{\mu} \right)^{j+1} \frac{1}{\kappa_{\rm red} \eta_1 (\frac{1}{2}\kappa_\chi)^j \epsilon} \left[f_{j+1}(x_{j+1,t}) - f_{j+1}(x_{j+1,t+1}) \right]. \end{aligned} \tag{2.75}$$

The induction is then completed, and the desired result follows since $d < j < r$. □

We finally prove the main result.

Theorem 2.3.16 *Assume that ϵ_r is "driven to zero" in Algorithm RMTR$_\infty$. Then*

$$\lim_{k \to \infty} \chi_{r,k} = 0. \tag{2.76}$$

Proof. As in Theorem 2.3.10, we identify our sequence of iterates with that generated by considering a sequence of tolerances $\{\epsilon_{r,j}\} \in (0,1)$ monotonically converging to zero. We start our proof by observing that the monotonic nature of the sequence $\{f(x_{r,\ell})\}_{\ell \geq 0}$ and the fact that $f(x)$ is bounded below impose that

$$f(x_{r,k}) - f(x_{r,k+1}) \to 0 \tag{2.77}$$

for all successful iterations (r,k). Assume now, for the purpose of deriving a contradiction, that

$$\limsup_{k \to \infty} \chi_{r,k} \geq 3\epsilon > 0 \tag{2.78}$$

for some $\epsilon \in (0,1)$ and consider a $k_0 > 0$ such that $\chi_{r,k_0} \geq 2\epsilon$ and such that both (2.67) and

$$f(x_{r,k}) - f(x_{r,k+1}) \leq \frac{\epsilon}{\kappa_{\rm acc}\kappa_{\rm L}} \tag{2.79}$$

hold for all $k \geq k_0$. Without loss of generality, we may assume that the minimization sequence at level r starts at iteration k_0. But Lemma 2.3.15 ensures that (2.68) holds for each successful iteration (r,k) $(k \geq k_0)$ as long as (2.66) holds. We may therefore apply Lemma 2.3.13 with

$$\epsilon_{\rm ncr} = 2\epsilon \quad \text{and} \quad \kappa_{\rm ncr} = \kappa_{\rm acc}$$

to the (truncated) minimization sequence at level r and deduce that (2.79) implies (2.58) and that (2.66) holds for all $k \geq k_0$, which is impossible in view of Theorem 2.3.10. Hence (2.78) is impossible and our proof complete. □

Theorem 2.3.16 implies, in particular, that any limit point of the infinite sequence $\{x_{r,k}\}$ is first-order critical for problem (2.2). But we may draw stronger conclusions: if we additionally assume that the trust region becomes asymptotically inactive at all levels, then, as explained above, each minimization sequence in the algorithm becomes infinite, and we may apply Theorem 2.3.16 to each of them, concluding that

$$\lim_{k\to\infty} \chi_{i,k} = 0$$

for every level $i = 0, \ldots, r$. The behavior of Algorithm RMTR$_\infty$ is therefore truly coherent with its multilevel formulation, since the same convergence results hold for each level.

Notice that the algorithm for which the convergence has been proved is a general algorithm that allows for many practical algorithms. Some algorithmic options covered by this general framework will be described in Chapter 4. The only requirements needed are the following. The transfer operators have to satisfy (2.6). The coarse representation $\mathcal{F}_{i-1}$ of the bound constraints has to be defined using (2.9)-(2.10) or by another formulae ensuring $s_i = P_i s_{i-1} \in \mathcal{F}_i$ for all $s_{i-1} \in \mathcal{F}_{i-1}$. The coarse model has to be chosen first-order coherent with the function it represents. There has to be a level-dependant stopping criterion based on the criticality measure but ϵ_i is not necessarily defined like in (2.20). The descent condition (2.19) is needed but the constant κ_χ can be different. The method used to compute the step at Taylor iterations has to satisfy the sufficient decrease condition (2.21). Finally, conditions (2.25) to (2.29) have to be satisfied, that is the Hessian matrix, its approximation, the gradient and the prolongation operator have to be bounded, and the criticality measure has to be Lipschitz continuous and to satisfy the property (2.28).

The convergence results at the upper level are unaffected if minimization sequences at lower levels are "prematurely" terminated, provided each such sequence contains at least one successful iteration. Indeed, none of the proofs depends on the actual stopping criterion used. Thus, one might think of stopping a minimization sequence after a preset number of successful iterations: in combination with the freedom left at Step 1 to choose the model whenever (2.19) holds, this strategy allows a straightforward implementation of fixed lower-iterations patterns, like the V- or W-cycles in multigrid methods. This will be explain in Chapter 4.

Our theory also remains essentially unchanged if we merely insist on first-order coherence (i.e., (2.18)) to hold only for small enough trust-region radii $\Delta_{i,k}$, or only up to a perturbation of the order of $\Delta_{i,k}$ (because it would imply that when $\Delta_{i,k}$ is getting small because steps are refused, the models become first-order coherent and therefore the descent condition prevent to make recursive steps, as required in the convergence proof). Other generalizations may be possible. Similarly, although we have assumed for motivation purposes that each f_i is "more costly" to minimize than f_{i-1}, we have not used this feature in the theory presented above, nor have we used the form of the lower levels' objective functions. In particular, to define f_i as identically zero for $i = 0, \ldots, r-1$ satisfies all our assumptions. Nonconstant prolongation and restriction operators of the form $P_i(x_{i,k})$ and $R_i(x_{i,k})$ may also be considered, provided the singular values of these operators remain uniformly bounded. We refer the reader to (Gratton et al. 2006*b*) for a discussion of convergence properties of multilevel trust-region methods to second-order critical points.

2.4 Identification of active constraints

2.4.1 Introduction

Trust-region methods for which step is computed by a method based on the generalized Cauchy step (Conn et al. 1993) have been proved to identify the correct active set after a finite number of iterations, first in a bound-constrained framework in Conn, Gould and Toint, 1988, then in Conn et al., 1993, for the convex-constrained case.

The aim of this section is to show that this identification theory remains valid when relying on steps that respect only some sufficient decrease property (known as the Cauchy condition) instead of using the generalized Cauchy step itself. This will allow us to deduce that RMTR_∞ identifies the active constraints in a finite number of steps if we force the algorithm to avoid the deactivation of active constraints when doing recursive steps. Note that the reasoning of this section has been made in a unilevel context such that the first subscript i is skipped in all quantities.

2.4.2 Active constraints identification

In the context of constrained optimization, the active constraints identification plays an important role in projection methods because ever since the complete active set has been identified it is possible to consider the problem as unconstrained and look for the solution only on the optimal face determined by the active set. A result has been obtained by Conn et al., 1993, for trust-region methods in Euclidean-norm on convex-constrained problems, which specifies that the entire set of active constraints at the exact solution is identified after a finite number of iterations and remains active until the exact solution is reached. We show that the proof obtained for a generalized Cauchy step, is still valid for an infinity-norm trust-region algorithm and when computing other kinds of step, provided that the step satisfies a sufficient decrease condition. Finally, the result of Conn et al., 1993, has been proved with a specific criticality measure and we will extend it to the use of other measures in the next chapter. We will base our proof on Section 12.3 of Conn et al., 2000, which summarizes the active constraints identification part of Conn et al., 1993. We include

all the lemmas and theorems that constitute the proof, such that the entire reasoning can be followed easily, but we will only demonstrate those where the proof changes either because of the use of a different step than the generalized Cauchy step because the infinite norm is used inside the trust-region algorithm or because of the choice of a possibly different criticality measure.

Before starting, let us recall the properties already asked for the convergence of RMTR_∞ that are also needed for the identification of active constraints theory. We again consider that we minimize a function f that maps $\mathbb{R}^n$ into $\mathbb{R}$, that is twice-continuously differentiable, bounded below and whose Hessian matrix is bounded above in $(\infty, 1)$-norm by a constant $\kappa_H - 1$, where $\kappa_H \geq 1$, like in (2.24). The model chosen in the trust-region algorithm has to be equal to the objective function when both evaluated at all the iterates x_k generated by the algorithm and so are their gradients, too. Finally, the Hessian Matrix of the model has also to be bounded above by the constant $\kappa_H - 1$, that is $\beta_k \leq \kappa_H$, like in (2.25). We add some assumptions concerning the set of constraints because we have chosen to prove the result in the more general framework of convex-constrained optimization. These are the same as those imposed in Conn et al., 2000 for the convergence of convex-constrained trust-region methods. We assume $\mathcal{F}$ is a nonempty closed convex set of constraints defined by $\mathcal{F} = \cap_{i=1}^m [\mathcal{F}]_i$ where $[\mathcal{F}]_i = \{x \in \mathbb{R}^n | c_i(x) \geq 0\}$, where each $c_i : \mathbb{R}^n \to \mathbb{R}$ is twice-continuously differentiable on $\mathbb{R}^n$. We define x_* to be a limit point of the sequence $\{x_k\}$ generated by the infinity-norm trust-region algorithm and we denote the set of all limit points by L_*. For all $x \in \mathcal{F}$ we define $\mathcal{A}(x) = \{i : c_i(x) = 0\}$, the set of active constraints at x. In addition, $\mathcal{N}(x)$ represents the normal cone of $\mathcal{F}$ at $x \in \mathcal{F}$ and, if X is a convex set, then the relative interior of X, denoted $\text{ri}\{X\}$, is its interior when X is regarded as a subset of its affine hull, that is the affine subspace with lowest dimensionality that contains X (see Conn et al., 1993 for more details). We also need some specific assumptions in the context of active constraints identification. The majority of these assumptions are also the same as those imposed in Conn et al., 2000 : we assume that for all $x_* \in \text{L}_*$, the vectors $\{\nabla_x c_i(x)\}_{i \in \mathcal{A}(x_*)}$ are linearly independent and $-\nabla_x f(x_*) \in \text{ri}\{\mathcal{N}(x_*)\}$. We also assume that the sequence of iterates $\{x_k\}$ lies in a closed, bounded domain Ω. But we need to generalize the last two assumptions made in Conn et al., 2000 to the use of any step s^{SD} that ensures the sufficient decrease condition. The first one is simply another way of writing the sufficient decrease condition (2.21)

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{all} \pi_k^{SD} \min \left[\frac{\pi_k^{SD}}{\beta_k}, \Delta_k \right], \tag{2.80}$$

for all k, where $\kappa_{all} \in (0, \frac{1}{2})$, where $\pi_k^{SD} = \min\{1, \chi(x_k^{SD})\}$ and $\chi(x_k^{SD})$ is a specific criticality measure depending on the current active set. For the sake of generality, we left the discussion about exact specifications of this measure for the next chapter (see Section 3.4.3). The only result that is based on the definition of the criticality measure will thus be proved in Chapter 3. We finally ask that the new iterate generated by the algorithm does not deactivate any constraint that was active at x_k^{SD} where the sufficient decrease holds

$$\mathcal{A}(x_k^{SD} = x_k + s_k^{SD}) \subseteq \mathcal{A}(x_k + s_k = x_k + s_k^{SD} + s_k^+). \tag{2.81}$$

Note that this condition means that the theory is also valid for the methods that perform additional steps s_k^+ on the face selected by a step x^{SD} in order to improve

convergence. In consequence, the theory of the identification of active constraints is valid for RMTR_∞ in the bound-constrained case if Taylor steps satisfy the sufficient decrease condition (2.80) (which will be proved in Section 4.1.1), if recursive steps are applied alternatively and if the constraints that are active at the Taylor step are forced to remain active after the recursive step. We will see in Chapter 4 that the practical algorithm we use corresponds to this description.

We now begin the first part of the proof, mainly about the geometry of the active set. The first lemma says that each connected set of limit points $\text{L} \subseteq \text{L}_*$ spreads onto a single face of the feasible region. This means that we may associate a specific active set $\mathcal{A}(\text{L})$ with each connected set of limit points. We can deduce from this first result that all connected sets of limit points are well separated when all limit points are finite, which is formally written in Lemma 2.4.2. Theorem 2.4.3 concludes that, for k sufficiently large, every iterate x_k lies in the neighborhood of a well-defined connected set L_{*k} of limit points and additionally that all the constraints that are not in $\mathcal{A}(\text{L}_{*k})$ are also inactive at x_k. Note that the proof of these results is exactly the same as in Conn et al., 2000 because the assumptions used in this proof are independent of the method and of the choice of the criticality measure.

Lemma 2.4.1 *For each connected set of limit points* $\text{L}(x_*) \subseteq \text{L}_*$*, there exists a set* $\mathcal{A}(\text{L}(x_*)) \subseteq \{1, ..., m\}$ *for which*

$$\mathcal{A}(x_*) = \mathcal{A}(\text{L}(x_*))$$

for all $x_* \in \text{L}(x_*)$.

Lemma 2.4.2 *There exists a* $\phi \in (0, 1)$ *such that*

$$dist(x_*, \text{L}') \geq \phi,$$

for every $x_* \in \text{L}_*$ *and each compact connected set of limit points* L' *such that* $\mathcal{A}(\text{L}') \neq \mathcal{A}(x_*)$.

Theorem 2.4.3 *There exists constants* $\delta \in (0, \frac{1}{4}\phi)$*,* $\phi \in (0, 1)$*, and an index* $k_1 \geq 0$ *such that, for* $k \geq k_1$*, there is a compact connected set of limit points* L_{*k} *such that*

$$x_k \in \mathcal{V}(\text{L}_{*k}, \delta), \tag{2.82}$$

where

$$\mathcal{V}(\varepsilon, \delta) \stackrel{def}{=} \{x \in I\!R^n | dist(x, \varepsilon) \leq \delta\},$$

and that for all $x \in \mathcal{V}(\text{L}_{*k}, \delta)$

$$\mathcal{A}(x) \subseteq \mathcal{A}(\text{L}_{*k}).$$

The following theorem, which ends the first part of the analysis about the geometry of the limit set, determines that if an iterate x_k is sufficiently close to its associated connected set of limit points, but x_k^{SD} has not found the whole set of active constraints, then x_k is bounded away from criticality in the sense that π_k^{SD} is bounded away from zero by a small constant independent of k. As the next chapter is dedicated to the choice of the criticality measure and because this theorem relies closely on its definition, this result will be proven in Section 3.4.3 of Chapter 3

Theorem 2.4.4 *There exists $k_2 \geq k_1$ (where k_1 is defined in Theorem 2.4.3) such that, if there is a $j \in \{1, ..., m\}$ with*

$$j \in \mathcal{A}(\mathrm{L}_{*k}) \text{ and } j \notin \mathcal{A}(x_k^{SD}) \tag{2.83}$$

for some $k \geq k_2$, then

$$\pi_k^{SD} \geq \epsilon_* \tag{2.84}$$

for some $\epsilon_* \in (0, 1)$ independent of k and j.

The second part of the generalization concerns the identification of the active constraints by the method. We begin by showing that, when k is sufficiently large, if the trust region radius is small and the final active set has not been identified by the step, then the iteration is very successful.

Lemma 2.4.5 *Suppose that*

$$\Delta_k \leq \frac{(1-\eta_2)\kappa_{all}\pi_k^{SD}}{\kappa_H} \tag{2.85}$$

for some $k \geq k_2$ and the final active set has not been identified by the step, that is (2.83) holds. Then iteration k is very successful and $\Delta_{k+1} \geq \Delta_k$.

Proof. The proof is similar to the proof of Theorem 12.3.8 of Conn et al., 2000 where we replace the sufficient decrease result and adapted some constants to deal with the infinite norm trust-region.

Applying the mean-value theorem to f and to m, we first obtain

$$\begin{aligned} |f(x_k + s_k) - m_k(x_k + s_k)| &= \Big| f(x_k) + \nabla f(x_k)^T s_k + \tfrac{s_k^T \nabla^2 f(\xi_f) s_k}{2} \\ &\quad - m_k(x_k) - \nabla m_k(x_k)^T s_k - \tfrac{s_k^T \nabla^2 m_k(\xi_m) s_k}{2} \Big|, \end{aligned}$$

where $\xi_f, \xi_m \in (x_k, x_k + s_k)$. Because of the assumption on the model and the function, and using the Cauchy-Schwarz inequality, this result becomes

$$\begin{aligned} |f(x_k + s_k) - m_k(x_k + s_k)| &\leq \frac{||[\nabla^2 f(\xi_f) - \nabla^2 m_k(\xi_m)] s_k||_1}{2} ||s_k||_\infty \\ &\leq \frac{||\nabla^2 f(\xi_f) - \nabla^2 m_k(\xi_m)||_{\infty,1}}{2} ||s_k||_\infty^2 \\ &\leq \kappa_H ||s_k||_\infty^2 \end{aligned} \tag{2.86}$$

On the other hand, (2.80) and (2.85) together imply that

$$\begin{aligned} m_k(x_k) - m_k(x_k + s_k) &\geq \kappa_{all}\pi_k^{SD} \min\left[\frac{\pi_k^{SD}}{\beta_k}, \Delta_k\right], \\ &\geq \kappa_{all}\pi_k^{SD}\Delta_k, \end{aligned}$$

This last result combined with (2.86) gives

$$\begin{aligned} |1 - \rho_k| &= \left| \frac{f(x_k + s_k) - m(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\ &\leq \frac{\kappa_H ||s_k||_\infty^2}{\kappa_{all}\pi_k^{SD}\Delta_k}. \end{aligned}$$

The trust-region principle gives us

$$|1 - \rho_k| \leq \frac{\kappa_H \Delta_k}{\kappa_{all} \pi_k^{SD}}.$$

Finally, in regard of (2.85), we conclude

$$|1 - \rho_k| \leq 1 - \eta_2,$$

which implies that iteration k is very successful. □

We next consider a maximal active set $\mathcal{A}_*^{max}$, that is $\mathcal{A}_*^{max} = \mathcal{A}(x_*)$ for some $x_* \in \mathrm{L}_*^{max} \subseteq \mathrm{L}_*$ and

$$\mathcal{A}_*^{max} \not\subset \mathcal{A}(u_*) \tag{2.87}$$

for any $u_* \in \mathrm{L}'_* \neq \mathrm{L}_*^{max}$. This allows us to prove the important Lemma 2.4.7 which states that the correct active set is identified at least on a subsequence of successful iterations. It requires to have the following result beforehand, that says that the projection of the gradient onto the tangent cone $\mathcal{T}(u)$ of $\mathcal{F}$ at a point u having the correct active set tends to zero as both this point and the iterates approach a connected set of limit points. The proof of Lemma 2.4.6 is exactly the same as in Conn et al., 2000.

Lemma 2.4.6 *Let $\mathcal{K}$ be the index set of an infinite subsequence such that*

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} dist(x_k, \mathrm{L}) = \lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} ||u_k - x_k||_\infty = 0$$

for some connected set of limit points $\mathrm{L} \subseteq \mathrm{L}_*$ and some sequence $\{u_k\}_{k \in \mathcal{K}}$ such that $u_k \in \mathcal{F}$ and $\mathcal{A}(u_k) = \mathcal{A}(\mathrm{L})$ for all $k \in \mathcal{K}$. Then one has that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} \mathrm{Proj}_{\mathcal{T}(u_k)}(-g_k) = 0.$$

Lemma 2.4.7 *There exists a subsequence $\{k_i\}$ of successful iterations such that, for i sufficiently large,*

$$\mathcal{A}(x_{k_i}) = \mathcal{A}_*^{max}. \tag{2.88}$$

Proof. The proof follows the same reasoning as the proof of Lemma 12.3.7 of Conn et al., 2000 where the sufficient decrease result is replaced by (2.80), where we use the new bound (2.85) of Lemma (2.4.5) on Δ_k and with some adaptation of the constants to deal with the infinite norm trust region.

We define the subsequence $\{k_j\}$ as the subsequence of successful iterates whose iterates approach limit points $x_* \in \mathrm{L}_{*k}$ with their active set equal to $\mathcal{A}_*^{max}$; that is,

$$\{k_j\} \stackrel{def}{=} \{k \in S | \mathcal{A}(\mathrm{L}_{*k}) = \mathcal{A}_*^{max}\},$$

and assume, for the purpose of obtaining a contradiction, that

$$\mathcal{A}(x_{k_j+1}) \neq \mathcal{A}_*^{max} \tag{2.89}$$

for j large enough. Assume now, again for the purpose of contradiction, that

$$\mathcal{A}_*^{max} \subseteq \mathcal{A}(x_{k_j}^{SD}) \tag{2.90}$$

for such a j. Using successively (2.81), (2.89), and Theorem 2.4.3, we obtain that, for j sufficiently large,

$$\mathcal{A}_*^{max} \subset \mathcal{A}(x_{k_j+1}) = \mathcal{A}(\mathrm{L}_{*k_j+1}),$$

which is impossible because of (2.87) and because Lemma 2.4.1 implies $\mathrm{L}_{*k_j+1} \neq \mathrm{L}_{*k}$. Hence (2.90) cannot hold, and there must be a $p_j \in \mathcal{A}_*^{max} = \mathcal{A}(\mathrm{L}_{*k_j})$ such that $p_j \notin \mathcal{A}(x_{k_j}^{SD})$ for j large enough. From Theorem 2.4.4, we then deduce that (2.84) holds for j sufficiently large. But the fact that iteration k_j is successful, together with (2.80) and the boundedness of the Hessian matrix of the model, implies that

$$\begin{aligned} f(x_{k_j}) - f(x_{k_j+1}) &\geq \eta_1 \kappa_{all} \epsilon_* \min\left[\frac{\epsilon_*}{\beta_{k_j}}, \Delta_{k_j}\right] \\ &\geq \eta_1 \kappa_{all} \epsilon_* \min\left[\frac{\epsilon_*}{\kappa_H}, \Delta_{k_j}\right] \end{aligned}$$

for j large enough, and thus that

$$\lim_{j \to \infty} \Delta_{k_j} = 0 \tag{2.91}$$

because f is bounded below by assumption. We therefore obtain that

$$||s_{k_j}||_\infty \leq \Delta_{k_j} \leq \frac{1}{2}\delta < \frac{1}{8}\psi$$

for j larger than j_1, say. But this last inequality and Theorems 2.4.2 and 2.4.3 imply that x_{k_j+1} cannot jump to the vicinity of any other connected set of limit points with a different active set, and hence x_{k_j+1} belongs to $\mathcal{V}(\mathrm{L}_{*k}, \delta)$ again and $\mathcal{A}(\mathrm{L}_{*k}) = \mathcal{A}_*^{max}$. Therefore, the subsequence $\{k_j\}$ is identical to the complete sequence of successful iterations with $k \geq k_1$. Hence we may deduce from (2.91) that

$$\lim_{\substack{k \to \infty \\ k \in S}} \Delta_k = 0. \tag{2.92}$$

But the mechanism of the Algorithm 2.1.1 (Δ_k is decreased after an unsuccessful iteration) and (2.92) also give the limit

$$\lim_{k \to \infty} \Delta_k = 0. \tag{2.93}$$

In particular, we have that

$$\Delta_k \leq \frac{\gamma_1^2 \kappa_{all} \epsilon_* (1 - \eta_2)}{\kappa_H} \tag{2.94}$$

for $k \in S$ sufficiently large. As a consequence we note that, for k large enough, x_k, x_k^{SD}, and $x_k + s_k$ all belong to $\mathcal{V}(\mathrm{L}, \delta)$ for a single connected set of limit points L. We also note that Lemma 2.4.5, the fact that (2.84) now holds for $k \in S$, and (2.92) together give that

$$k \in S \Rightarrow \Delta_{k+1} \geq \Delta_k \tag{2.95}$$

for k large enough. We can therefore deduce the desired contradiction from (2.93) and (2.95) if we can prove that all iterations are eventually successful.

Suppose therefore that this is not the case. It is then possible to find a subsequence K of sufficiently large k such that

$$k \notin S \text{ and } k+1 \in S. \tag{2.96}$$

Note that, because of the mechanism of Step 4 of the Algorithm 2.1.1, one has that

$$\Delta_k \leq \frac{\Delta_{k+1}}{\gamma_1} \leq \frac{\gamma_1 \kappa_{all} \epsilon_* (1-\eta_2)}{\kappa_H}, \tag{2.97}$$

where we used (2.94) to deduce the last inequality. Now, if one has that

$$\mathcal{A}(x_k^{SD}) \subset \mathcal{A}(\mathrm{L}) = \mathcal{A}_*^{max}, \tag{2.98}$$

then Theorem 2.4.4 and Lemma 2.4.5 together with (2.97) imply that $k \in S$, which contradicts (2.96). Hence (2.98) cannot hold, and (2.81) with Theorem 2.4.4 give that

$$\mathcal{A}(x_k + s_k) = \mathcal{A}(x_k^{SD}) = \mathcal{A}(\mathrm{L}) \tag{2.99}$$

for k sufficiently large. Observe now that, since $k \notin S$, one has that $x_{k+1} = x_k$, and hence that

$$\begin{aligned}
m_{k+1}(x_{k+1}+s_{k+1}) - m_k(x_k+s_k) & \\
&= m_{k+1}(x_k+s_{k+1}) - m_k(x_k+s_k) \\
&= \langle -g_k, s_k - s_{k+1}\rangle + \frac{1}{2}\langle s_{k+1}, \nabla_x^2 m_{k+1}(\xi_{k+1}) s_{k+1}\rangle \\
&\quad -\frac{1}{2}\langle s_k, \nabla_x^2 m_k(\xi_k) s_k\rangle \\
&\geq \langle -g_k, s_k - s_{k+1}\rangle - \frac{1}{2}\beta_{k+1}\Delta_{k+1}^2 - \frac{1}{2}\beta_k \Delta_k^2 \\
&\geq \langle -g_k, s_k - s_{k+1}\rangle - \frac{1}{2}\kappa_H (1+\frac{1}{\gamma_1^2})\Delta_{k+1}^2,
\end{aligned}$$

where we have successively used $\xi_k \in [x_k, x_k+s_k]$, $\xi_{k+1} \in [x_{k+1}, x_{k+1}+s_{k+1}]$, the Cauchy-Schwarz inequality, the boundedness of the Hessian matrix of the model, the fact that the infinite norm of the steps is bounded by the trust region radius, the norm equivalence and the mechanism of Step 4 of the Algorithm 2.1.1. But

$$\begin{aligned}
\langle -g_k, s_k - s_{k+1}\rangle &= \langle \mathrm{Proj}_{\mathcal{T}(x_k+s_k)}(-g_k), s_k - s_{k+1}\rangle \\
&\quad + \langle \mathrm{Proj}_{\mathcal{N}(x_k+s_k)}(-g_k), s_k - s_{k+1}\rangle \\
&\geq -||\mathrm{Proj}_{T(x_k+s_k)}(-g_k)||_2 \, ||s_k - s_{k+1}||_2 \\
&\quad + \langle \mathrm{Proj}_{N(x_k+s_k)}(-g_k), \mathrm{Proj}_{T(x_k+s_k)}(s_k - s_{k+1})\rangle \\
&\geq -||\mathrm{Proj}_{T(x_k+s_k)}(-g_k)||_2 \sqrt{n}(||s_k||_\infty + ||s_{k+1}||_\infty) \\
&\geq -\left(1+\frac{1}{\gamma_1^2}\right)\sqrt{n}\Delta_{k+1}||\mathrm{Proj}_{T(x_k+s_k)}(-g_k)||_2
\end{aligned}$$

for all $k \in K$, where we have used the Moreau decomposition (p36 of Conn et al., 2000) of $-g_k$, the fact that $s_{k+1} - s_k \in T(x_k+s_k)$, which is the polar of $N(x_k+s_k)$,

and, as above, the fact that the steps are bounded by the trust region radius, the fact that $\gamma_1 < 1$, and the mechanism of Step 4 of Algorithm 2.1.1. Combining the last two chains of inequalities, we obtain that

$$\begin{aligned} & m_{k+1}(x_{k+1}+s_{k+1}) - m_k(x_k+s_k) \\ & \geq -\left(1+\tfrac{1}{\gamma_1^2}\right)\Delta_{k+1}\left[\sqrt{n}||\mathrm{Proj}_{T(x_k+s_k)}(-g_k)||_2 + \tfrac{1}{2}\kappa_H\Delta_{k+1}\right] \end{aligned}$$

for such k. We now recall (2.93) and, because of the equality (2.99), apply Lemma 2.4.6 (with $u_k = x_k + s_k$) and therefore deduce from this last inequality that

$$m_{k+1}(x_{k+1}+s_{k+1}) - m_k(x_k+s_k) \geq -\frac{1}{2}\kappa_{all}\epsilon_*\Delta_{k+1}$$

for k large enough in K. On the other hand, (2.80) and $\beta_k \leq \kappa_{umh}$ imply that

$$f(x_{k+1}) - m_{k+1}(x_{k+1}+s_{k+1}) \geq \kappa_{all}\epsilon_*\Delta_{k+1}.$$

Hence, recalling that $k \notin S$, we obtain that

$$\begin{aligned} f(x_k) - m_k(x_k+s_k) &= f(x_{k+1}) - m_{k+1}(x_{k+1}+s_{k+1}) \\ &\quad + m_{k+1}(x_{k+1}+s_{k+1}) - m_k(x_k+s_k) \\ &\geq \frac{1}{2}\kappa_{all}\epsilon_*\Delta_{k+1} \\ &\geq \frac{1}{2}\kappa_{all}\gamma_1\epsilon_*\Delta_{k+1} \end{aligned}$$

for all $k \in K$ sufficiently large. But then, using the definition of ρ_k, Theorem 6.41 (p 133 of Conn et al., 2000) with ($\nu_k^s = 1$), and (2.97), we obtain that

$$|\rho_k - 1| \leq \frac{2\kappa_H}{\kappa_{all}\gamma_1\epsilon_*}\Delta_k \leq 1 - \eta_2$$

and hence that $\rho_k \geq \eta_2$ for all $k \in K$ large enough, which contradicts (2.96). The condition (2.89) is thus impossible for k sufficiently large. All iterations are eventually very successful, which produces the desired contradiction. As a consequence, (2.89) cannot hold for all j, and we obtain that there exists a subsequence $\{k_p\} \subseteq \{k_j\}$ such that, for all p,

$$\mathcal{A}_*^{max} = \mathcal{A}(x_{k_p+1}) = \mathcal{A}(x_{k_p+q(k_p)}),$$

where $x_{k_p+q(k_p)}$ is the first successful iteration after iteration k_p. The lemma is thus proved if we choose $\{k_i\} = \{k_p + q(k+p)\}$. □

We finish by showing that, once found, the maximal active set $\mathcal{A}_*^{max}$ cannot be abandoned for sufficiently large k. This allows us to reformulate the convergence to first-order critical points in terms of the projected gradient in Corollary 2.4.9

Theorem 2.4.8 *One has that*

$$\mathcal{A}(x_*) = \mathcal{A}_*^{max} \tag{2.100}$$

for all $x_* \in \mathrm{L}_*$, *and*

$$\mathcal{A}(x_k) = \mathcal{A}(x_k^{SD}) = \mathcal{A}_*^{max} \tag{2.101}$$

for all k *sufficiently large.*

Proof. The proof is similar to the proof of Theorem 12.3.8 of Conn et al., 2000 where we replace the sufficient decrease result by (2.80) and adapted some constants to deal with the infinite norm trust-region.

Consider $\{k_i\}$, the subsequence of successful iterates such that (2.88) holds, as given by Lemma 2.4.7. Suppose furthermore that this subsequence is restricted to sufficiently large indices, that is, $k_i \geq k_2$ for all i. Suppose finally that there exists a subsequence of $\{k_i\}$, say $\{k_p\}$, such that, for each p, there is a j_p such that

$$j_p \in \mathcal{A}(x_{k_p}) \text{ and } j_p \notin \mathcal{A}(x_{k_p+1}).$$

Now Theorem 2.4.3 and (2.88) give that $\mathcal{A}(\mathrm{L}_{*k+p}) = \mathcal{A}_*^{max}$. Using this observation and (2.81), we obtain that

$$j_p \in \mathcal{A}(\mathrm{L}_{*k+p}) \text{ and } j_p \notin \mathcal{A}(x_{k_p}^{SD})$$

for all p. But Theorem 2.4.4 then ensures that

$$\pi_{k_p}^{SD} \geq \epsilon_* \tag{2.102}$$

for all p. Combining this with (2.80) and the boundedness of the Hessian matrix of the model, we obtain that, for all p,

$$f(x_{k_p}) - f(x_{k_p+1}) \geq \eta_1 \kappa_{all} \epsilon_* \min\left[\frac{\epsilon_*}{\kappa_H}, \Delta_{k_p}\right],$$

which implies that

$$\lim_{k\to\infty} \Delta_{k_p} = 0. \tag{2.103}$$

Applying (2.80) and the boundedness of the Hessian matrix of the model, and using (2.102), we obtain that

$$f(x_{k_p}) - m_{k_p}(x_{k_p} + s_{k_p}) \geq \kappa_{all} \epsilon_* \Delta_{k_p} \tag{2.104}$$

for all p sufficiently large. On the other hand, we have that, for all k,

$$\begin{aligned} |\langle -g_k, s_k\rangle| &= |\langle \mathrm{Proj}_{\mathcal{T}(x_k)}(-g_k), s_k\rangle + \langle \mathrm{Proj}_{\mathcal{N}(x_k)}(-g_k), \mathrm{Proj}_{\mathcal{T}(x_k)}(s_k)\rangle| \\ &\leq |\langle \mathrm{Proj}_{\mathcal{T}(x_k)}(-g_k), s_k\rangle| \\ &\leq ||\mathrm{Proj}_{\mathcal{T}(x_k)}(-g_k)||_1 ||s_k||_\infty \end{aligned}$$

where we have used the Moreau decomposition, the fact that $s_k \in \mathcal{T}(x_k)$ and the Cauchy-Schwarz inequality. This implies

$$\begin{aligned} f(x_k) - m_k(x_k + s_k) &\leq |\langle -g_k, s_k\rangle| + \kappa_H ||s_k||_\infty^2 \\ &\leq \tfrac{|\langle -g_k, s_k\rangle|}{||s_k||_\infty} \Delta_k + \kappa_H ||s_k||_\infty^2, \\ &\leq ||\mathrm{Proj}_{\mathcal{T}(x_k)}(-g_k)||_1 \Delta_k + \kappa_H \Delta_k^2, \end{aligned} \tag{2.105}$$

Combining (2.104), (2.105) taken at $k = k_p$, and dividing both sides by Δ_k, we obtain that

$$\kappa_{all} \epsilon_* \leq ||\mathrm{Proj}_{T(x_{k_p})}(-g_k)||_1 + \kappa_H \Delta_{k_p}. \tag{2.106}$$

Assuming that the sequence $\{x_{k_p}\}$ converges to some x_* in some L (or taking a subsequence if necessary), using (2.103) and Lemma 2.4.6 (with $K = \{k_p\}$, $u_k = x_k$

and $\mathcal{A}(\mathrm{L}) = \mathcal{A}_*^{max}$), we deduce that (2.106) is impossible for p large enough. As a consequence, no such subsequence $\{k_p\}$ exists, and we have that, for large i,

$$\mathcal{A}_*^{max} \subseteq \mathcal{A}(x_{k_i}^{SD}) \subseteq \mathcal{A}(x_{k_i+1}) \subseteq \mathcal{A}(\mathrm{L}_{*k_i+1}),$$

where the last inclusion follows from Theorem 2.4.3. But the maximality of $\mathcal{A}_*^{max}$ then imposes that

$$\mathcal{A}_*^{max} = \mathcal{A}(x_{k_i}^{SD}) = \mathcal{A}(x_{k_i+1}) = \mathcal{A}(\mathrm{L}_{*k_i+1})$$

for i sufficiently large. Hence we obtain that, for i large enough, $\mathcal{A}(x_{k_i+q}) = \mathcal{A}_*^{max}$, where $k_i + q$ is the index of the first successful iteration after iteration k_i. Hence $k_i + q \in \{k_i\}$. We can therefore repeatedly apply this reasoning and deduce that

$$\{k_i\} = \{k \in S | k \text{ is sufficiently large}\}$$

and also that $\mathcal{A}(x_k) = \mathcal{A}_*^{max}$ for all $k \in S$ large enough, hence providing (2.101). Moreover, $\mathcal{A}_*^{max}$ is then the only possible active set for the limit points, which proves (2.100). □

Corollary 2.4.9 *We have*

$$\lim_{k\to\infty} ||\mathrm{Proj}_{\mathcal{T}(x_k)}(-g_k)||_2 = 0.$$

Proof. This immediately results from Lemma 2.4.6 (with $K = \{1, 2, ...\}$ and $u_k = x_k$) and Theorem 2.4.8. □

We have proved that the active constraints identification theory for trust-region methods in Euclidean norm with convex constraints developed by Conn et al., 1993, can be generalized to the use of any step respecting a sufficient decrease condition and to the use of the infinity norm in the trust-region constraint without significant modification.

2.5 Conclusion

In this Chapter, we have presented the general multilevel trust-region algorithm in infinity-norm RMTR_∞ designed for both unconstrained and bound-constrained nonlinear optimization. We have proved it is globally convergent to first-order critical points and it identifies the correct active set in a finite number of iterations. Sections 2.2 and 2.3 correspond to the main part of Gratton, Mouffe, Toint and Weber-Mendonça, 2008*a*.

Chapter 3

Stopping criteria for bound-constrained optimization

In this chapter, we get a closer look into the definition of a suitable stopping criterion for nonlinear bound-constrained optimization. More precisely, we are interested into relating the criticality measures used as stopping criteria in bound-constrained optimization to backward error analysis coming from linear algebra. This will lead us to consider also the backward error problem from the point of view of multicriteria optimization. We finally show that the different measures discussed in this chapter satisfy the requirements for the convergence of RMTR_∞, stated in Chapter 2.

3.1 Backward error analysis

3.1.1 Introduction to backward error analysis for optimization

We are still interested in solving a problem of the type (2.2)

$$\min_{\mathcal{F}} f(x),$$

where $\mathcal{F} = \{x \in \mathbb{R}^n | l \leq x \leq u\}$ is a set of bound constraints and $l, u \in \mathbb{R}^n$. We define the *active set of binding constraints*, for all $x \in \mathcal{F}$, as $\mathcal{A}(x) = \mathcal{A}^-(x) \cup \mathcal{A}^+(x)$ with

$$\begin{array}{rcl} \mathcal{A}^-(x) = \{j \in \{1,...,n\} \mid [x]_j & = & [l]_j \quad \text{and} \quad [\nabla_x f(x)]_j > 0\} \\ \mathcal{A}^+(x) = \{j \in \{1,...,n\} \mid [x]_j & = & [u]_j \quad \text{and} \quad [\nabla_x f(x)]_j < 0\}. \end{array}$$

In that context, if $[\nabla_x f(x_*)]_j = 0$ for all $j \notin \mathcal{A}(x_*)$, then we say that x_* is a first-order critical point of (2.2).

We consider iterative optimization methods that produce a sequence of iterates x_k which converges to a first-order solution x_* of the problem to solve. But this sequence can be infinite. We are thus interested in determining when to stop the algorithm in order to achieve a reasonable reliability of the approximate solution. A first way of expressing this problem is to stop the iterations when the current iterate x_k is such that

$$||x_k - x_*|| < \epsilon,$$

where ϵ is the tolerance we accept on the distance between the approximate and the first-order solution. But we generally do not know the exact solution as we are

precisely looking for it. As a consequence, we prefer to consider the backward error, which replaces the question *How far from the solution is the current iterate x_k ?* by *If there exists a minimization problem (P) such that x_k is one of its first-order solutions, how far from the original problem (2.2) is (P) ?* In the backward error analysis context we stop the algorithm at iteration k when x_k is a first-order critical point of a perturbed version of the original problem (2.2):

$$\min_{l+\Delta l \leq x \leq u+\Delta u} f(x) + \Delta f + \Delta g^T x,$$

and when the perturbations $\Delta l, \Delta u, \Delta f, \Delta g \in \mathbb{R}^n$ are sufficiently small. The first-order sufficient condition of optimality implies $[\nabla_x f(x_k)+\Delta g]_j = 0$ for all $j \notin \mathcal{A}_\Delta(x_k)$, where $\mathcal{A}_\Delta(x) = \mathcal{A}^-_\Delta(x) \cup \mathcal{A}^+_\Delta(x)$, with

$$\begin{array}{l} \mathcal{A}^-_\Delta(x) = \{j \in \{1,...,n\} \mid [x]_j = [l]_j + [\Delta l]_j \text{ and } [\nabla_x f(x) + \Delta g]_j > 0\} \\ \mathcal{A}^+_\Delta(x) = \{j \in \{1,...,n\} \mid [x]_j = [u]_j + [\Delta u]_j \text{ and } [\nabla_x f(x) + \Delta g]_j < 0\}. \end{array}$$

The value of Δf does not appear in this sufficient condition such that we can set $\Delta f = 0$ without loss of generality. Finally, we are looking for

$$y \stackrel{def}{=} (\Delta g\,;\, \Delta l\,;\, \Delta u) \in \mathcal{Y}_k$$

with

$$\mathcal{Y}_k \stackrel{def}{=} \{(\Delta g\,;\, \Delta l\,;\, \Delta u) \in \mathbb{R}^{3n} : [\nabla_x f(x_k) + \Delta g]_j = 0 \text{ for all } j \notin \mathcal{A}_\Delta(x_k)\}$$

and where y is a vector composed by the three perturbation vectors $\Delta g, \Delta l$ and Δu stucked into a long vector $y = (\Delta g\,;\, \Delta l\,;\, \Delta u)$, and, finally, where a product norm has to be defined for vectors $(\Delta g\,;\, \Delta l\,;\, \Delta u)$ in the space of the perturbations. The algorithm is then stopped if

$$\inf_{y \in \mathcal{Y}_k} \parallel y = (\Delta g\,;\, \Delta l\,;\, \Delta u) \parallel < \epsilon(\epsilon_g, \epsilon_l, \epsilon_u),$$

where $\epsilon_g, \epsilon_l, \epsilon_u \in \mathbb{R}$ are chosen tolerances which represent in most of the cases an order of magnitude corresponding to the accuracy of the computation of g, l and u. Moreover, notice that this infimum is actually a minimum. Indeed, looking at $\mathcal{Y}_k$, we see that it is equal to the direct product on all $j = 1, ..., n$ of the sets $[\mathcal{Y}_k]_j$ containing all the possible solutions for $\{y\}_j \stackrel{not.}{=} ([\Delta g]_j\,;\, [\Delta l]_j\,;\, [\Delta u]_j) \in \mathbb{R}^3$. The sets $[\mathcal{Y}_k]_j$ are each composed by the union of two direct products of three elements : we necessarily have, for all j,

$$[\Delta g]_j = [-\nabla_x f(x_k)]_j \text{ or } \left\{ \begin{array}{ll} [\Delta_l]_j = [x_k - l]_j & \text{if } [\nabla_x f(x_k) + \Delta g]_j > 0 \\ [\Delta_u]_j = [x_k - u]_j & \text{if } [\nabla_x f(x_k) + \Delta g]_j < 0 \end{array} \right.$$

while the other components of $\{y\}_j$ can take any value in $\mathbb{R}^n$ on the condition that

$l + \Delta l \leq x_k \leq u + \Delta u$ and, therefore,

$$[\mathcal{Y}_k]_j = \left\{ \begin{array}{l} \left\{ ([-\nabla_x f(x_k)]_j; [\Delta l]_j; [\Delta u]_j) \ : \ \begin{array}{l} [\Delta l]_j, [\Delta u]_j \in \mathbb{R} \\ [l + \Delta l]_j \leq [x_k]_j \leq [u + \Delta u]_j \end{array} \right\} \\ \qquad \cup \\ \left\{ ([\Delta g]_j; [x_k - l]_j; [\Delta u]_j) \ : \ \begin{array}{l} [\Delta g]_j, [\Delta u]_j \in \mathbb{R} \\ [x_k]_j \leq [u + \Delta u]_j \\ [\nabla_x f(x_k) + \Delta g]_j > 0 \end{array} \right\} \\ \qquad \cup \\ \left\{ ([\Delta g]_j, [\Delta l]_j, [x_k - u]_j) \ : \ \begin{array}{l} [\Delta g]_j, [\Delta l]_j \in \mathbb{R} \\ [l + \Delta l]_j \leq [x_k]_j \\ [\nabla_x f(x_k) + \Delta g]_j < 0 \end{array} \right\}. \end{array} \right.$$

Finally, we deduce that $\mathcal{Y}_k$ is the union of a finite number of direct products between closed sets and is thus itself a closed set. As a consequence, we have for some $y_0 \in \mathcal{Y}_k$ (which exists because $\mathcal{Y}_k$ is obviously not empty)

$$\begin{aligned} \inf_{y \in \mathcal{Y}_k} \| \, (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| &= \inf_{\substack{y \, \in \, \mathcal{Y}_k \\ ||y|| \, \leq \, ||y_0||}} \| \, (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| \\ &= \min_{\substack{y \, \in \, \mathcal{Y}_k \\ ||y|| \, \leq \, ||y_0||}} \| \, (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| \\ &= \min_{y \in \mathcal{Y}_k} \| \, (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| \end{aligned}$$

where the first and the third equalities are coming from the fact that $y_0 \in \mathcal{Y}_k$ and from the definition of the objective function, while the second equality holds because the set of constraints is closed and bounded since it is the intersection of a closed set ($\mathcal{Y}_k$ is closed) and a bounded set ($\{y \in \mathcal{Y}_k : ||y|| \leq ||y_0||\}$). In conclusion, we can choose to stop the algorithm as soon as

$$\min_{y \in \mathcal{Y}_k} \| \, y = (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| < \epsilon(\epsilon_l, \epsilon_u, \epsilon_g).$$

3.1.2 Characteristics of the optimal solution of the backward error analysis problem

In this section we are interested in finding $y \in \mathcal{Y}_k$ that is optimal for

$$\chi_k \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} \| \, y = (\Delta g \, ; \, \Delta l \, ; \, \Delta u) \, \| < \epsilon(\epsilon_l, \epsilon_u, \epsilon_g),$$

in order to define a general criticality measure based on backward error analysis and suitable as a stopping criterion. In this work we have chosen to look more closely at two definitions for this specific norm. The first norm we consider is

$$\chi_k^{out} = \min_{y \in \mathcal{Y}_k} ||y = (\Delta g \, ; \, \Delta l \, ; \, \Delta u)||_{out} \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} (\alpha_g ||\Delta g||_g + \alpha_l ||\Delta l||_l + \alpha_u ||\Delta u||_u), \quad (3.1)$$

where $(\alpha_g, \alpha_l, \alpha_u) \in (0, 1]^3$ and where the norms on $\Delta l, \Delta u, \Delta g$ are *monotone* norms, in the sense that each of these three norms satisfies the following property

$$\forall j \in \{1, ..., n\} |[u]_j| \geq |[v]_j| \text{then } ||u|| \geq ||v|| \quad \forall u, v \in \mathbb{R}^n.$$

We may also impose to choose strictly monotone norms, that is monotone norms such that

$$\exists j \in \{1, ..., n\} | [u]_j | > | [v]_j | \text{then } ||u|| > ||v|| \quad \forall u, v \in \mathbb{R}^n,$$

in order to derive stronger results as will be discussed in Section 3.3. The second choice for the product norm is

$$\chi_k^{in} = \min_{y \in \mathcal{Y}_k} ||y = (\Delta g\,;\, \Delta l\,;\, \Delta u)||_{in} \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\,||_{glu}, \quad (3.2)$$

where $(\alpha_g, \alpha_l, \alpha_u) \in (0, 1]^3$ and where the norm on the sum is again a monotone norm. It is easy to check that both $||.||_{in}$ and $||.||_{out}$ satisfy all the norm properties (see Appendix B.2). Notice in addition that they are both symmetric norms because of the presence of the absolute values and the positiveness of the weights α_g, α_l and α_u. In the sequel of this chapter we refer to the set of all the optimal solutions of (3.1) and (3.2) as $\mathcal{S}_k^{out}$ and $\mathcal{S}_k^{in}$, respectively. They have the following immediate property :

$$\mathcal{S}_k^{out} \subseteq \mathcal{Y}_k \text{ and } \mathcal{S}_k^{in} \subseteq \mathcal{Y}_k.$$

We restrict ourselves to the case where $||.||_g, ||.||_l, ||.||_u$ and $||.||_{glu}$ are monotone norms because this restriction allows us to characterize easily the solution of the problem and to finally find an explicit form of this solution. Notice that, in particular, all the p-norms, $1 \leq p < \infty$, are strictly monotone norms and that the infinity norm is monotone. Moreover, the choice left for $||.||_g, ||.||_l$ and $||.||_u$ in the definition of χ_k^{out} opens the possibility of choosing for $||.||_g$ the *dual norm* of $||.||_l = ||.||_u$. But, on the contrary, the energy-norm (or A-norm) defined by

$$||v||_A^2 = v^T A v,$$

where v is a vector of $\mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, is not a strictly monotone norm. Indeed, consider for example the A-norm of $u, v \in \mathbb{R}^3$ with

$$A = \begin{pmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 0.4 & 0.4 & 1 \end{pmatrix}, \; u = (-10, 10, -10) \text{ and } v = (9, 9, 9).$$

We have $|[u]_j| > |[v]_j|$ for all j, but $||u||_A \approx 14.83$ and $||v||_A \approx 20.91$, and $||v||_A$ is larger than $||u||_A$ which contradicts the definition of a monotone norm. Finally, notice that if $||.||_g = ||.||_l = ||.||_u = ||.||_{glu} = ||.||_p$, with $1 \leq p \leq \infty$, then

$$\chi_k^{in,p} \leq \chi_k^{out,p}$$

where $\chi_k^{in,p} = \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\,||_p$ and $\chi_k^{out,p} = \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_p + \alpha_l ||\Delta l||_p + \alpha_u ||\Delta u||_p$. Indeed,

$$\begin{aligned} \chi_k^{in,p} &= \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\,||_p \\ &\leq \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g|\,||_p + ||\alpha_l |\Delta l| + \alpha_u |\Delta u|\,||_p \\ &\leq \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g|\,||_p + ||\alpha_l |\Delta l|\,||_p + ||\alpha_u |\Delta u|\,||_p \\ &= \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_p + \alpha_l ||\Delta l||_p + \alpha_u ||\Delta u||_p \end{aligned}$$

where we used the fact that α_g, α_l and α_u are positive as well as norm properties.

Now define the set of *undecided indices*

$$\mathcal{U}_k = \{j \in \{1, ..., n\} \mid [\nabla_f(x_k)]_j \neq 0 \text{ and } j \notin \mathcal{A}(x_k)\},$$

that will play an important role in the following key result which characterizes $\mathcal{S}_k^{out}$ and $\mathcal{S}_k^{in}$. Lemma 3.1.1 shows that the optimal solution y^* of (3.1), as well as the optimal solution of (3.2) is located in a specific set $\mathcal{P}_k \subseteq \mathcal{Y}_k$ such that the choice of each of its components $\{y^*\}_j = ([\Delta g^*]_j\,;\,[\Delta l^*]_j\,;\,[\Delta u^*]_j)$ is independent from the choice of the others. Moreover, this characterization of the optimal solution leaves us the choice between only two explicit sets of values for each $\{y^*\}_j$, $j \in \mathcal{U}_k$.

Lemma 3.1.1 *If $||.||_g, ||.||_l, ||.||_u$ and $||.||_{glu}$ are monotone norms, the optimal solution $y^* \in \mathcal{S}_k^{out} \subseteq \mathcal{P}_k$ of the problem (3.1), as well as the optimal solution $y^* \in \mathcal{S}_k^{in} \subseteq \mathcal{P}_k$ of the problem (3.2), where $\mathcal{P}_k \subseteq \mathcal{Y}_k$ is a set that contains all the $y \in \mathcal{Y}_k$ for which the following description holds.*

For all $j \notin \mathcal{U}_k$:

$$\{y\}_j = (0\,;\,0\,;\,0) \tag{3.3}$$

For all $j \in \mathcal{U}_k$ and $[\nabla_x f(x_k)]_j > 0$:

$$\{y\}_j = ([-\nabla_x f(x_k)]_j\,;\,0\,;\,0) \tag{3.4}$$

or

$$\{y\}_j = (0\,;\,[x_k - l]_j\,;\,0) \tag{3.5}$$

For all $j \in \mathcal{U}_k$ and $[\nabla_x f(x_k)]_j < 0$:

$$\{y\}_j = ([-\nabla_x f(x_k)]_j\,;\,0\,;\,0) \tag{3.6}$$

or

$$\{y\}_j = (0\,;\,0\,;\,[x_k - u]_j) \tag{3.7}$$

Proof. First notice that $y = (\Delta g\,;\,\Delta l\,;\,\Delta u) \in \mathcal{P}_k$ implies $y \in \mathcal{Y}_k$. Indeed, for all undecided indices $j \in \mathcal{U}_k$, either we have $[\Delta g]_j = [-\nabla_x f(x_k)]_j$ and thus $[\nabla_x f(x_k) + \Delta g]_j = 0$, or (3.5) and (3.7) imply that $j \in \mathcal{A}_\Delta(x_k)$.

We now want to prove that $\mathcal{S}_k^{out} \subseteq \mathcal{P}_k$ and $\mathcal{S}_k^{in} \subseteq \mathcal{P}_k$. For this purpose, we would like to show that for any $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$, there exists at least one solution $y \in \mathcal{P}_k$ that leads to a smaller value in the objective functions of (3.1) and (3.2). We thus consider $\hat{y} = (\widehat{\Delta g}\,;\,\widehat{\Delta l}\,;\,\widehat{\Delta u}) \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$ and prove that for all j such that the definition of $\{\hat{y}\}_j$ implies $\hat{y} \notin \mathcal{P}_k$, there exists at least one $y \in \mathcal{P}_k$ such that $\{y\}_j$ satisfies :

$$|\left[\widehat{\Delta g}\right]_j| \geq |[\Delta g]_j| \text{ and } |\left[\widehat{\Delta l}\right]_j| \geq |[\Delta l]_j| \text{ and } |\left[\widehat{\Delta u}\right]_j| \geq |[\Delta u]_j|. \tag{3.8}$$

and at least one of

$$|\left[\widehat{\Delta g}\right]_j| > |[\Delta g]_j| \text{ or } |\left[\widehat{\Delta l}\right]_j| > |[\Delta l]_j| \text{ or } |\left[\widehat{\Delta u}\right]_j| > |[\Delta u]_j|. \tag{3.9}$$

We distinguish a few cases :
First if $j \notin \mathcal{U}_k$: Equation (3.3) defines $[\Delta g]_j = [\Delta l]_j = [\Delta u]_j = 0$, thus for any other solution $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$ we obviously have that $\{\hat{y}\}_j$ satisfies (3.8)-(3.9).
Then if $j \in \mathcal{U}_k$ and $[\nabla_x f(x_k)]_j > 0$: First notice that, in this case, $\hat{y} \in \mathcal{Y}_k$ implies that one of the following holds :

(a) $\left[\widehat{\Delta g}\right]_j = [-\nabla_x f(x_k)]_j$,

(b) $\left[\widehat{\Delta l}\right]_j = [x_k - l]_j$,

(c) $\left[\widehat{\Delta u}\right]_j = [x_k - u]_j$ and $\left[\widehat{\Delta g}\right]_j < [-\nabla_x f(x_k)]_j$.

All the solutions such that (c) holds obviously satisfy (3.8)-(3.9) for all values of $\left[\widehat{\Delta l}\right]_j$, because of (3.4) and the fact that (c) and $[\nabla_x f(x_k)]_j > 0$ imply $|\left[\widehat{\Delta g}\right]_j| > |[-\nabla_x f(x_k)]_j|$. The solutions such that (a) holds but $\left[\widehat{\Delta l}\right]_j \neq 0$ or $\left[\widehat{\Delta u}\right]_j \neq 0$, as well as the solutions such that (b) holds but $\left[\widehat{\Delta g}\right]_j \neq 0$ or $\left[\widehat{\Delta u}\right]_j \neq 0$ are all satisfying also (3.8)-(3.9) because of (3.4) and (3.5), respectively.
Finally, if $j \in \mathcal{U}_k$ and $[\nabla_x f(x_k)]_j < 0$: A reasoning similar to the one used in the previous case leads to (3.8)-(3.9) hold for all $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$ In summary, if $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$, there always exists $y \in \mathcal{P}_k$ such that we have that for all j

$$|\left[\widehat{\Delta g}\right]_j| \geq |[\Delta g]_j|,\ |\left[\widehat{\Delta l}\right]_j| \geq |[\Delta l]_j| \text{ and } |\left[\widehat{\Delta u}\right]_j| \geq |[\Delta u]_j| \tag{3.10}$$

and, moreover,

$$|\left[\widehat{\Delta g}\right]_j| > |[\Delta g]_j| \text{ or } |\left[\widehat{\Delta l}\right]_j| > |[\Delta l]_j| \text{ or } |\left[\widehat{\Delta u}\right]_j| > |[\Delta u]_j| \tag{3.11}$$

for at least one j. Remember we have assumed that we used monotone norms $||.||_g, ||.||_l$ and $||.||_u$. As a consequence,

$$||\widehat{\Delta g}||_g \geq ||\Delta g||_g,\ ||\widehat{\Delta l}||_l \geq ||\Delta l||_l \text{ and } ||\widehat{\Delta u}||_u \geq ||\Delta u||_u. \tag{3.12}$$

Moreover, if $||.||_g, ||.||_l$ and $||.||_u$ are strictly monotone norms we also have at least one of

$$||\widehat{\Delta g}||_g > ||\Delta g||_g \text{ or } ||\widehat{\Delta l}||_l > ||\Delta l||_l \text{ or } ||\widehat{\Delta u}||_u > ||\Delta u||_u \tag{3.13}$$

hold. These last inequalities are not necessary to prove the current lemma but will be used later in this chapter.

Now, as $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$, for all $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$, there exists $y \in \mathcal{P}_k \subseteq \mathcal{Y}_k$ such that $\alpha_g||\widehat{\Delta g}||_g + \alpha_l||\widehat{\Delta l}||_l + \alpha_u||\widehat{\Delta u}||_u \geq \alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u$. Finally,

$$\min_{y \in \mathcal{Y}_k} \alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u = \min_{y \in \mathcal{P}_k} \alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u \tag{3.14}$$

and, therefore, $\mathcal{S}_k^{out} \subseteq \mathcal{P}_k$. On the other hand, notice that (3.10)-(3.11), and the fact that $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ also imply $\alpha_g|\widehat{\Delta g}| + \alpha_l|\widehat{\Delta l}| + \alpha_u|\widehat{\Delta u}| \geq \alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|$, where the absolute values and the inequality are understood componentwise and, consequently,

$$\left| \alpha_g|\widehat{\Delta g}| + \alpha_l|\widehat{\Delta l}| + \alpha_u|\widehat{\Delta u}| \right| \geq |\ \alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\ |\,.$$

Then using the monotonicity of the norm $||.||_{glu}$ we obtain that for all $\hat{y} \in \mathcal{Y}_k$, $\hat{y} \notin \mathcal{P}_k$, there exists $y \in \mathcal{P}_k \subseteq \mathcal{Y}_k$ satisfying

$$\left|\left| \alpha_g|\widehat{\Delta g}| + \alpha_l|\widehat{\Delta l}| + \alpha_u|\widehat{\Delta u}| \right|\right|_{glu} \geq ||\ \alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\ ||_{glu}\,,$$

which gives the desired result

$$\min_{y\in\mathcal{Y}_k} || \alpha_g|\Delta g|+\alpha_l|\Delta l|+\alpha_u|\Delta u| ||_{glu} = \min_{y\in\mathcal{P}_k} || \alpha_g|\Delta g|+\alpha_l|\Delta l|+\alpha_u|\Delta u| ||_{glu} \quad (3.15)$$

and, finally, $\mathcal{S}_k^{in} \subseteq \mathcal{P}_k$. □

We now state an important consequence of Lemma 3.1.1, which says that the minimization of $y \in \mathcal{P}_k$ is *separable* in j.

Corollary 3.1.2 *The definition of $\mathcal{P}_k$ is such that for all $y \in \mathcal{P}_k$. Moreover, for all $y \in \mathcal{P}_k$, for all $(\alpha_g, \alpha_l, \alpha_u) \in (0, 1]^3$,*

$$\min_{y\in\mathcal{P}_k}(\alpha_g[|\Delta g|]_j + \alpha_l[|\Delta l|]_j + \alpha_u[|\Delta u|]_j) = \min_{\{y\}_j\in\mathcal{P}_k}(\alpha_g[|\Delta g|]_j + \alpha_l[|\Delta l|]_j + \alpha_u[|\Delta u|]_j). \quad (3.16)$$

Proof. The definition of y and the definition of $\mathcal{P}_k$, together with the fact that all the components j of the elements of $\mathcal{P}_k$ are chosen independently between two possibilities, imply (3.16). □

3.2 Criticality measures

3.2.1 Introductory study

In the previous section, Lemma 3.1.1 gave a first idea of what $y^* \in \mathcal{S}_k^{out}$ and $y^* \in \mathcal{S}_k^{in}$ look like. For the sake of generality, we have left the choice of the norms $||.||_g, ||.||_l, ||.||_u$ and $||.||_{glu}$ unspecified until now. In this section, we will see that taking some specific norms leads to expressions of χ_k^{out} and χ_k^{in} that correspond the most common criticality measure used in the bound-constrained optimization community. Before going into the details, we begin by this preliminary lemma.

Lemma 3.2.1 *The optimal value of the unidirectional minimization problem*

$$[\mathcal{M}]_j = \min_{\{y\}_j \ s.t. \ y\in\mathcal{P}_k} \alpha_g|[\Delta g]_j| + \alpha_l|[\Delta l]_j| + \alpha_u|[\Delta u]_j|, \quad j = 1, ..., n,$$

is

$$[\mathcal{M}]_j = \begin{cases} \min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_l|[x_k - l]_j|\} & \text{if } [\nabla_x f(x_k)]_j > 0, \\ \min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_u|[x_k - u]_j|\} & \text{if } [\nabla_x f(x_k)]_j < 0, \\ 0 & \text{if } [\nabla_x f(x_k)]_j = 0. \end{cases} \quad (3.17)$$

and its optimal solution y^* is such that

$$\begin{array}{l} \text{If } j \notin \mathcal{U}_k \ : \ \{y^*\}_j = (0\,;\,0\,;\,0) \\ \text{If } j \in \mathcal{U}_k \text{ and } [\nabla_x f(x_k)]_j > 0 : \\ \quad \text{If } \alpha_l|\,[x_k - l]_j\,| \geq \alpha_g|\,[\nabla_x f(x_k)]_j\,| \ : \ \{y^*\}_j = ([-\nabla_x f(x_k)]_j\,;\,0\,;\,0) \\ \quad \text{If } \alpha_l|\,[x_k - l]_j\,| \leq \alpha_g|\,[\nabla_x f(x_k)]_j\,| \ : \ \{y^*\}_j = (0\,;\,[x_k - l]_j\,;\,0) \\ \text{If } j \in \mathcal{U}_k \text{ and } [\nabla_x f(x_k)]_j < 0 : \\ \quad \text{If } \alpha_u|\,[x_k - u]_j\,| \geq \alpha_g|\,[\nabla_x f(x_k)]_j\,| \ : \ \{y^*\}_j = ([-\nabla_x f(x_k)]_j\,;\,0\,;\,0) \\ \quad \text{If } \alpha_u|\,[x_k - u]_j\,| \leq \alpha_g|\,[\nabla_x f(x_k)]_j\,| \ : \ \{y^*\}_j = (0\,;\,0\,;\,[x_k - u]_j) \end{array} \quad (3.18)$$

Proof. If $j \notin \mathcal{U}_k$, $y \in \mathcal{P}_k$ implies that $[\Delta g]_j = [\Delta l]_j = [\Delta u]_j = 0$. But if $j \in \mathcal{U}_k$, the definition of $\mathcal{P}_k$ in Lemma 3.1.1 leaves the choice between two solutions. The first solution is $\{y\}_j = ([-\nabla_x f(x_k)]_j \,;\, 0 \,;\, 0)$ and, in this case, the objective value of the j^{th} unidirectional minimization is equal to $\alpha_g |[\Delta g]_j| = \alpha_g |[\nabla_x f(x_k)]_j|$. If the second solution

$$\{y\}_j = \begin{cases} (0 \,;\, [x_k - l]_j \,;\, 0) & \text{if } [\nabla_x f(x_k)]_j > 0 \\ (0 \,;\, 0 \,;\, [x_k - u]_j) & \text{if } [\nabla_x f(x_k)]_j < 0 \end{cases}$$

is preferred, then $[\mathcal{M}]_j$ is equal to $\alpha_l |[\Delta l]_j| = \alpha_l |[x_k - l]_j|$ if $[\nabla_x f(x_k)]_j > 0$ or to $\alpha_u |[\Delta u]_j| = \alpha_u |[x_k - u]_j|$ if $[\nabla_x f(x_k)]_j < 0$. The solution of the unidirectional problem $[\mathcal{M}]_j$ is thus $\{y^*\}_j$ such that (3.18) holds. As a consequence, for all j,

$$[\mathcal{M}]_j = \begin{cases} \min\{\alpha_g |[\nabla_x f(x_k)]_j|, \alpha_l |[x_k - l]_j|\} & \text{if } [\nabla_x f(x_k)]_j | > 0, \\ \min\{\alpha_g |[\nabla_x f(x_k)]_j|, \alpha_u |[x_k - u]_j|\} & \text{if } [\nabla_x f(x_k)]_j | < 0, \\ 0 & \text{if } [\nabla_x f(x_k)]_j | = 0. \end{cases}$$

□

3.2.2 Criticality measure based on the definition of χ_k^{out}

We are interested in finding an explicit solution of the problem (3.1) :

$$\chi_k^{out} = \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_g + \alpha_l ||\Delta l||_l + \alpha_u ||\Delta u||_u.$$

In this case, if $||.||_g = ||.||_l = ||.||_u = ||.||_1$, then it is possible to obtain an explicit value for χ_k^{out}. This is done in the following theorem, where the result of Lemma 3.1.1 plays a central role. Finding an explicit solution in the other cases, even not proved impossible, is at least not obvious and has not been done in this work.

Theorem 3.2.2 *If* $||.||_g = ||.||_l = ||.||_u = ||.||_1$ *, then*

$$\chi_k^{out,1} \overset{def}{=} \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_1 + \alpha_l ||\Delta l||_1 + \alpha_u ||\Delta u||_1 \tag{3.19}$$

$$= ||\mathcal{M}\,||_1, \tag{3.20}$$

with $[\mathcal{M}]_j$ *defined by (3.17).*

Proof. We begin by showing that the minimization problem (3.19) can be reduced to n independent unidirectional minimization problems. The definition of the 1-norm and (3.19) imply

$$\chi_k^{out,1} = \min_{y \in \mathcal{Y}_k} \alpha_g \sum_{j=1}^{n} |[\Delta g]_j| + \alpha_l \sum_{j=1}^{n} |[\Delta l]_j| + \alpha_u \sum_{j=1}^{n} |[\Delta u]_j|$$

which in turn is the same as

$$\chi_k^{out,1} = \min_{y \in \mathcal{Y}_k} \sum_{j=1}^{n} \alpha_g |[\Delta g]_j| + \alpha_l |[\Delta l]_j| + \alpha_u |[\Delta u]_j|.$$

Recall that a conclusion of Lemma 3.1.1 is $\mathcal{S}_k^{out} \subseteq \mathcal{P}_k$. Thus we can minimize only on the subset $\mathcal{P}_k \subseteq \mathcal{Y}_k$:

$$\chi_k^{out,1} = \min_{y \in \mathcal{P}_k} \sum_{j=1}^{n} \alpha_g |[\Delta g]_j| + \alpha_l |[\Delta l]_j| + \alpha_u |[\Delta u]_j|.$$

Now notice all the elements of the sum are positive and thus the minimization and the sum can be inverted, such that using Corollary 3.1.2 gives

$$\chi_k^{out,1} = \sum_{j=1}^{n} \min_{\{y\}_j, y \in \mathcal{P}_k} (\alpha_g |[\Delta g]_j| + \alpha_l |[\Delta l]_j| + \alpha_u |[\Delta u]_j|).$$

Finally, the result of Lemma 3.2.1 gives

$$\chi_k^{out,1} = \sum_{j=1}^{n} \begin{cases} \min\{\alpha_g |[\nabla_x f(x_k)]_j|, \alpha_l |[x_k - l]_j|\} & \text{if } [\nabla_x f(x_k)]_j| > 0, \\ \min\{\alpha_g |[\nabla_x f(x_k)]_j|, \alpha_u |[x_k - u]_j|\} & \text{if } [\nabla_x f(x_k)]_j| < 0, \end{cases}$$

which is exactly (3.20). □

This result leads to an important statement in the case where the weights in (3.1) are chosen specifically. Indeed, in the particular case where the weights are all equal to 1, the measure of the backward error is $||\Gamma_k||_1$, where Γ_k is the projection of the negative gradient on the feasible set, that is a vector the components of which are defined by

$$\begin{aligned} [\Gamma_k]_j &= [\text{Proj}_{\mathcal{F}}(x_k - \nabla_x f(x_k)) - x_k]_j \\ &= \begin{cases} [-\nabla_x f(x_k)]_j & \text{if } \begin{cases} [\nabla_x f(x_k)]_j > 0 \text{ and } |[\nabla_x f(x_k)]_j| \leq |[l]_j - [x_k]_j|, \\ [\nabla_x f(x_k)]_j < 0 \text{ and } |[\nabla_x f(x_k)]_j| \leq |[u]_j - [x_k]_j|, \end{cases} \\ [l]_j - [x_k]_j & \text{if } [\nabla_x f(x_k)]_j > 0 \text{ and } |[\nabla_x f(x_k)]_j| > |[l]_j - [x_k]_j|, \\ [u]_j - [x_k]_j & \text{if } [\nabla_x f(x_k)]_j < 0 \text{ and } |[\nabla_x f(x_k)]_j| > |[u]_j - [x_k]_j|. \end{cases} \end{aligned}$$

In a more general case, when $\alpha_{lu} \stackrel{def}{=} \alpha_l = \alpha_u$ then the measure of the backward error is $||\Gamma_k(\alpha_g, \alpha_{lu})||_1$, where

$$\Gamma_k(\alpha_g, \alpha_{lu}) \stackrel{def}{=} \alpha_{lu} \left(\text{Proj}_{\mathcal{F}}(x_k - \frac{\alpha_g}{\alpha_{lu}} \nabla_x f(x_k)) - x_k \right). \tag{3.21}$$

This is stated formally in the following result.

Corollary 3.2.3 *If $\alpha_{lu} = \alpha_l = \alpha_u$, and if the 1-norm is used in (3.1), then*

$$\chi_k^{out,1} = ||\Gamma_k(\alpha_g, \alpha_{lu})||_1 . \tag{3.22}$$

If, in addition, $\alpha_g = \alpha_{lu} = 1$, then the optimal value for (3.1) is

$$\chi_k^{out,1} = ||\Gamma_k||_1. \tag{3.23}$$

Proof. The result of Theorem 3.2.2, when $\alpha_l = \alpha_u$, becomes

$$\begin{aligned}
\chi_k^{out,1} &= \sum_{j=1}^{n} \begin{cases} \min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_{lu}|[x_k - l]_j|\} & \text{if } [\nabla_x f(x_k)]_j > 0 \\ \min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_{lu}|[x_k - u]_j|\} & \text{if } [\nabla_x f(x_k)]_j < 0 \end{cases} \\
&= \alpha_{lu} \sum_{j=1}^{n} \begin{cases} \min\{\frac{\alpha_g}{\alpha_{lu}}|[\nabla_x f(x_k)]_j|, |[x_k - l]_j|\} & \text{if } [\nabla_x f(x_k)]_j > 0 \\ \min\{\frac{\alpha_g}{\alpha_{lu}}|[\nabla_x f(x_k)]_j|, |[x_k - u]_j|\} & \text{if } [\nabla_x f(x_k)]_j < 0 \end{cases} \\
&= \alpha_{lu} \sum_{j=1}^{n} \begin{cases} \min\{|[\frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)]_j|, |[x_k - l]_j|\} & \text{if } [\frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)]_j > 0 \\ \min\{|[\frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)]_j|, |[x_k - u]_j|\} & \text{if } [\frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)]_j < 0 \end{cases} \\
&= \alpha_{lu} \sum_{j=1}^{n} \left| \left[\text{Proj}_{\mathcal{F}}(x_k - \frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)) - x_k \right]_j \right| \\
&= ||\Gamma_k(\alpha_g, \alpha_{lu})||_1 .
\end{aligned}$$

where we used $(\alpha_g, \alpha_{lu}) \in (0,1]^2$. In addition, (3.23) follows immediately if $\alpha_g = \alpha_{lu} = 1$. □

3.2.3 Criticality measure based on the definition of χ_k^{in}

We are now interested in finding an explicit solution of the second backward error problem (3.2)

$$\chi_k^{in} = \min_{y \in \mathcal{Y}_k} ||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\,||_{glu}.$$

In this case, if a p-norm is chosen for $||.||_{glu}$, with $1 \leq p \leq \infty$, then it is possible to obtain an explicit value for $\chi_k^{in,p}$, like in the previous case. This is done in the following theorems, where the result of Lemma 3.1.1 plays a central role. Notice that a direct consequence is that in the special case where $p = 1$, we have $\chi_k^{in,1} = \chi_k^{out,1}$.

Theorem 3.2.4 *If a p-norm is used in (3.2), with $1 \leq p < \infty$, then*

$$\chi_k^{in,p} \overset{def}{=} \min_{y \in \mathcal{Y}_k} ||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\,||_p \tag{3.24}$$

$$= ||\mathcal{M}\,||_p \tag{3.25}$$

with $[\mathcal{M}]_j$ defined by (3.17).

Proof. The definition of the p-norm and (3.24) first give

$$\chi_k^{in,p} = \min_{y \in \mathcal{Y}_k} \sqrt[p]{\sum_{j=1}^{n} ([\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\,]_j)^p}.$$

The minimization can go inside the p-root as well as inside the sum since the content of the root is positive and, in regard of Lemma 3.1.1, $\mathcal{S}_k^{in} \subseteq \mathcal{P}_k$, such that

$$\chi_k^{in,p} = \sqrt[p]{\min_{y \in \mathcal{P}_k} \sum_{j=1}^{n} ([\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\,]_j)^p}.$$

Now notice all the elements of the sum are positive and thus the minimization and the sum can be inverted and the minimization can also enter into the power of p, such that using Corollary 3.1.2 gives

$$\begin{aligned}\chi_k^{in,p} &= \sqrt[p]{\sum_{j=1}^{n}(\min_{y\in\mathcal{P}_k}[\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\ ]_j)^p} \\ &= \sqrt[p]{\sum_{j=1}^{n}\left(\min_{\{y\}_j, y\in\mathcal{P}_k}\alpha_g|[\Delta g]_j| + \alpha_l|[\Delta l]_j| + \alpha_u|[\Delta u]_j|\right)^p}.\end{aligned}$$

Finally, using the result of Lemma 3.2.1, we conclude that

$$\chi_k^{in,p} = \sqrt[p]{\sum_{j=1}^{n}\left(\begin{cases}\min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_l|[x_k-l]_j|\} & \text{if } [\nabla_x f(x_k)]_j| > 0, \\ \min\{\alpha_g|[\nabla_x f(x_k)]_j|, \alpha_u|[x_k-u]_j|\} & \text{if } [\nabla_x f(x_k)]_j| < 0,\end{cases}\right)^p},$$

which is the desired result. □

Theorem 3.2.5 *If a ∞-norm is used in (3.2) the optimal value for this problem can be defined by*

$$\begin{aligned}\chi_k^{in,\infty} &\overset{def}{=} \min_{y\in\mathcal{Y}_k}||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\ ||_\infty \\ &= ||\mathcal{M}\ ||_\infty \end{aligned} \tag{3.26}$$

with $[\mathcal{M}]_j$ defined by (3.17).

Proof. First notice that $\mathcal{P}_k$ contains a finite number of elements equal to $2^{\#\mathcal{U}_k}$, such that Lemma 3.1.1 implies

$$\begin{aligned}\chi_k^{in,\infty} &= \min_{y\in\mathcal{P}_k}||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\ ||_\infty \\ &= \min\{||y_1||_\infty, ..., ||y_{2^{\#\mathcal{U}_k}}||_\infty\} \\ &= \min\{\lim_{p\to\infty}||y_1||_p, ..., \lim_{p\to\infty}||y_{2^{\#\mathcal{U}_k}}||_p\} \\ &= \lim_{p\to\infty}\min\{||y_1||_p, ..., ||y_{2^{\#\mathcal{U}_k}}||_p\}\end{aligned}$$

As there is a finite number of elements in the minimization set, we can write

$$\begin{aligned}\chi_k^{in,\infty} &= \lim_{p\to\infty}\chi_k^{in,p} \\ &= \lim_{p\to\infty}||\mathcal{M}||_p \\ &= ||\mathcal{M}||_\infty\end{aligned}$$

where we used $\lim_{p\to\infty}||.||_p = ||.||_\infty$ and the result of Theorem 3.2.4. □

When $\alpha_l = \alpha_u$ or when all the weights in (3.2) are equal to 1, those results lead, as in the previous case, to the fact that the measure of the backward error is the p-norm of $\Gamma_k(\alpha_g, \alpha_{lu})$.

Corollary 3.2.6 *If* $\alpha_{lu} \overset{def}{=} \alpha_l = \alpha_u$*, and if the p-norm is used in (3.2), with* $1 \leq p \leq \infty$*, the optimal value for this problem can be defined by*

$$\chi_k^{in,p} = ||\Gamma_k(\alpha_g, \alpha_{lu})||_p . \tag{3.27}$$

If, in addition, $\alpha_g = \alpha_{lu} = 1$ *the the optimal value of (3.2) is,*

$$\chi_k^{in,p} = ||\Gamma_k||_p . \tag{3.28}$$

Proof. The proof is similar to the proof of Corollary 3.2.3. □

3.2.4 Further comparisons

We can also be interested in another criticality measure defined by

$$\chi_k^{tr} \overset{def}{=} \chi^{tr}(x_k) = \left| \min_{\substack{x_k + d \in \mathcal{F} \\ ||d||_\infty \leq 1}} \nabla_x f(x_k)^T d \right| \tag{3.29}$$

This measure has the main advantage that it gives a first-order approximation of the decrease that could be achieved in the negative gradient direction (see Conn et al. (1993)) and some algorithms like, for instance, all trust-region methods, are based on obtaining a decrease of the same order as the Cauchy decrease. As a consequence, the criticality measure χ_k^{tr} can be particularly interesting from this point of view. Unfortunately, χ_k^{tr} cannot be linked with backward errors as simply as the traditional $||\Gamma_k||_p$. Indeed, this measure is actually the product of two quantities : the negative gradient $-\nabla_x f(x_k)$ and a vector d^* defined by

$$d^* = \arg \max_{\substack{x_k + d \in \mathcal{F} \\ ||d||_\infty \leq 1}} -\nabla_x f(x_k)^T d$$

The result of Lemma 3.1.1 implies that $|| - \nabla_x f(x_k)||$ is the backward error if there is no uncertainty on the bounds, i.e. $\Delta l = \Delta u = 0$. In addition, d^* is equal to

$$[d^*]_j = \begin{cases} \max\{[l]_j - [x_k]_j, 1\} & \text{if } [-\nabla_x f(x_k)]_j > 0 \\ \max\{[u]_j - [x_k]_j, 1\} & \text{if } [-\nabla_x f(x_k)]_j < 0 \end{cases}$$

and therefore, as soon as $||d_k^*||_\infty < 1$, $||d_k^*||_{glu}$ is a backward error (in the χ_k^{in}-sense) in the case where we assume $\Delta g = 0$. Nevertheless, as χ_k^{tr} is the product of those quantities, we cannot relate it in itself to any of the backward error we have defined. Moreover, it is possible to show that χ_k^{tr} is not a backward error in any norm. This is done in the following lemma.

Lemma 3.2.7 *There does not exist a norm* $||.||_{tr}$ *such that, for all functions f and all sets of bounds* $\mathcal{F}$*,*

$$\chi_k^{tr} = \min_{y \in \mathcal{Y}_k} ||y||_{tr}. \tag{3.30}$$

In other words, χ_k^{tr} *is not a backward error.*

Proof. We only need to find one case (one specific x_k, f, $\mathcal{F}$) where there is no norm such that

$$\chi_k^{tr} = \min_{y \in \mathcal{Y}_k} ||y||_{tr}.$$

So consider we are minimizing a linear function subject to the bound constraints $l \leq x \leq u$ and such that its constant gradient is negative, that is $\nabla_x f(x_k) < 0$ for all iterates x_k, where the inequality is understood componentwise. If we consider some $x_k > u - 1$, in that specific case, $d_k^* = (u - x_k)$ for all k and

$$\chi_k^{tr} = |-\nabla_x f(x_k)^T (u - x_k)|.$$

So we assume there exists $||.||_{tr}$ such that

$$\chi_k^{tr} = |-\nabla_x f(x_k)^T (u - x_k)| = \min_{y \in \mathcal{Y}_k} ||y||_{tr} = ||y^*||_{tr} = ||(\Delta g^*, \Delta l^*, \Delta u^*)||_{tr}.$$

As we have assumed that $\nabla_x f(x_k) < 0$, the vectors $\Delta g^*, \Delta l^*, \Delta u^*$ are such that

$$\begin{gathered} [\nabla_x f(x_k) + \Delta g^*]_j = 0 \\ \text{or} \\ [u + \Delta u^*]_j = [x_k]_j \text{ and } [\nabla_x f(x_k) + \Delta g^*]_j < 0. \end{gathered} \tag{3.31}$$

As a consequence, we set $\Delta l^* = 0$ without loss of generality. In summary, we assume that there exists $||.||_{tr}$ such that

$$|-\nabla_x f(x_k)^T (u - x_k)| = ||(\Delta g^*, \Delta u^*)||_{tr}, \tag{3.32}$$

We then obtain, using the Cauchy-Schwarz inequality,

$$1 \geq \frac{|-\nabla_x f(x_k)^T (u - x_k)|}{||-\nabla_x f(x_k)||_2 ||u - x_k||_2} = \frac{||(\Delta g^*, \Delta u^*)||_{tr}}{||-\nabla_x f(x_k)||_2 ||u - x_k||_2}. \tag{3.33}$$

Now, we first assume that for all iterations k sufficiently big, $[\Delta g^*]_j = [-\nabla_x f(x_k)]_j$ for all j. In that case, because all norms are equivalent in finite dimension, there exists a constant ν such that

$$||(\Delta g^*, \Delta u^*)||_{tr} \geq \nu(||\Delta g^*||_2 + ||\Delta u^*||_2) \geq \nu||\Delta g^*||_2 = \nu||-\nabla_x f(x_k)||_2 \tag{3.34}$$

where we used the fact that $||(u, v)|| \stackrel{def}{=} ||u||_2 + ||v||_2$ is a norm on $\mathbb{R}^n \times \mathbb{R}^n$, where n is the dimension of the problem. Equation (3.33) therefore gives $1 \geq \nu / ||u - x_k||_2$. If we consider the sequence of iterates such that x_k is monotonically converging to the upper bound u (implying $x_k > u - 1$ for all k), then this last equation leads to

$$1 \geq \lim_{x_k \to u} \frac{\nu}{||u - x_k||_2} = +\infty,$$

which is impossible. We thus conclude that our assumpion is false and, because of (3.31), we deduce that there exists at least one index j such that $[\Delta u^*]_j = [u - x_k]_j$. This, together with the first inequality of (3.34), implies

$$||(\Delta g^*, \Delta u^*)||_{tr} \geq \nu(||\Delta g^*||_2 + ||\Delta u^*||_2) \geq \nu||\Delta u^*||_2 \geq \nu|[u - x_k]_j|$$

and, therefore, (3.33) gives

$$1 \geq \frac{\nu|[u - x_k]_j|}{|| - \nabla_x f(x_k)||_2 ||u - x_k||_2}.$$

Now consider the case where x_k is monotonically converging to the upper bound u such that $[u - x_k]_j = 1/k$, for all j, for all k. Then we have $||u - x_k||_2 = \sqrt{n}/k$ and we obtain

$$1 \geq \frac{\nu}{\sqrt{n}|| - \nabla_x f(x_k)||_2},$$

which is impossible for all problems where the constant gradient is chosen such that $|| - \nabla_x f(x_k)||_2 < \nu/\sqrt{n}$. We conclude that our assumption (3.32) is false, and the proof is complete. □

In conclusion, despite the fact that χ_k^{tr} has very interesting properties from the point of view of some specific algorithms, it is inadvisable from the perspective of backward error analysis.

3.3 Multicriteria Optimization

3.3.1 Introduction

The backward error problem is to find the minimal distance between the original problem we would like to solve and the closest problem we have already solved at iteration k. This distance is traditionally measured by means of a product norm defined on the space of the perturbations Δg, Δl and Δu. But we could see the problem from another point of view : instead of looking for the minimal norm of Δg, Δl and Δu, we could take the norm of the minimal Δg, Δl and Δu. This second approach makes us consider the problem from the point of view of multicriteria optimization and, surprisingly, does not necessarily give the same results as with the traditional approach. More precisely, we will see that all the solutions found while minimizing the norm of the perturbations (using χ_k^{out}) can be reached when doing multicriteria optimization, but the opposite is not true. We begin this section by a short introduction of the basic concepts of multicriteria optimization and refer the reader to Ehrgott (2005) for a more extensive coverage of that subject.

3.3.2 Pareto optimal solutions

The problem we want to solve in backward error analysis is actually a multicriteria optimization (MCO) problem of the form

$$\begin{array}{l} \text{“ min}'' \left([f(y)]_1, ..., [f(y)]_p\right) \\ \text{s.t. } y \in \mathcal{Y}_k, \end{array} \tag{3.35}$$

with $p = 3$, where $y = (\Delta g\,;\, \Delta l\,;\, \Delta u) \in \mathbb{R}^{3n}$ and $([f(y)]_1 = ||\Delta g||_g, [f(y)]_2 = ||\Delta l||_l, [f(y)]_3 = ||\Delta u||_u) \in \mathbb{R}^p$, and where

$$\mathcal{Y}_k = \{y : [\nabla_x f(x_k) + \Delta g]_j = 0 \text{ for all } j \notin \mathcal{A}_\Delta(x_k)\}.$$

In that context, $\mathbb{R}^{3n}$, the space of the variables, is called the *decision space*, while $\mathbb{R}^p$ is referred to as the *criterion space*. In the multicriteria optimization framework,

we want to minimize simultaneously some possibly conflicting functions. We thus look for a reasonable solution, but this notion is quite subjective and we have to define clearly what kind of solution is acceptable for our problem. The main idea is to consider acceptable all the feasible points but those which we know cannot be the best solution, even subjectively. In practice, we do not accept a feasible y if there exists another feasible point that produces a better, more *efficient*, objective function. We thus introduce an *order* on the criterion space $\mathbb{R}^p$ to make easier the comparison between objective functions in $\mathbb{R}^p$ and to help choosing if one is better than another. More precisely, we define the *componentwise order* on $\mathbb{R}^p$ by

$$f \leq g \quad \Leftrightarrow \quad f \neq g \text{ and } [f]_\ell \leq [g]_\ell, \text{ for all } \ell = \{1, ..., p\} \tag{3.36}$$

and the *strict componentwise order* on $\mathbb{R}^p$ by

$$f < g \quad \Leftrightarrow \quad [f]_\ell < [g]_\ell, \text{ for all } \ell = \{1, ..., p\}. \tag{3.37}$$

Notice that, in our case, a componentwise order corresponds to choose monotone norms for $||.||_g, ||.||_l$ and $||.||_u$ and a strict componentwise order corresponds to choose strictly monotone norms. For all $y_1, y_2 \in \mathcal{Y}_k$, if

$$\begin{array}{l} \text{for all } \ell = 1, .., p, \ [f(y_1)]_\ell \leq [f(y_2)]_\ell \text{ and} \\ \text{there exists } \ell \in \{1, .., p\}, \ [f(y_1)]_\ell < [f(y_2)]_\ell. \end{array}$$

then we say that y_1 *dominates* y_2 and $f(y_1)$ *dominates* $f(y_2)$. It is easy to see that the domination character is transitive, that is for all $y_1, y_2, y_3 \in \mathcal{Y}_k$,

$$y_1 \text{ dominates } y_2 \text{ and } y_2 \text{ dominates } y_3 \quad \Rightarrow \quad y_1 \text{ dominates } y_3. \tag{3.38}$$

We can now express formally what is called an *efficient* or a *Pareto optimal* solution of the multicriteria optimization problem :

y^* is a *Pareto optimal* solution

$$\Leftrightarrow \quad \nexists\, y \in \mathcal{Y}_k \text{ such that } \left\{ \begin{array}{l} \text{for all } \ell = 1, .., p, \ [f(y)]_\ell \leq [f(y^*)]_\ell \text{ and} \\ \text{there exists } \ell \in \{1, .., p\}, \ f_\ell(y) < f_\ell(y^*), \end{array} \right.$$

that is a feasible point that is not dominated by any other feasible point. The set of all efficient solutions is called the *efficient set* or *Pareto optimal set* and is denoted $\mathcal{Y}_E$, while $\mathcal{Z}_N$ represents the set of all *nondominated points* $z_n = f(y_e) \in \mathbb{R}^p$, where $y_e \in \mathcal{Y}_E$, and is called the *nondominated set*.

Lemma 3.1.1 stated that $\mathcal{P}_k$ contains the solution of the two backward error problems we have examined. Looking back to this lemma from the point view of multicriteria optimization, it also ensures that all $y \in \mathcal{Y}_k$ that are not represented in $\mathcal{P}_k$ are dominated by at least one point of $\mathcal{P}_k$. It follows that all $\hat{y} \notin \mathcal{P}_k$ cannot be efficient for the MCO problem (3.35). Indeed, the two results (3.12) and (3.13) from the proof of Lemma 3.1.1 imply that if $||.||_g, ||.||_l$ and $||.||_u$ are strictly monotone norms, all solutions $\hat{y} \notin \mathcal{P}_k$ are dominated by at least one solution $y \in \mathcal{P}_k$ and thus cannot be efficient for the original MCO problem. We finally have

$$\mathcal{Y}_E \subseteq \mathcal{P}_k.$$

Notice that we do not necessarily have $\mathcal{Y}_E = \mathcal{P}_k$. Nevertheless, it is sufficient to check if a solution $y \in \mathcal{P}_k$ is nondominated by all the other solutions of $\mathcal{P}_k$ to be sure that y is efficient. This is shown in the following theorem.

Theorem 3.3.1 *If $y_1 \in \mathcal{P}_k$ is nondominated by all $y_2 \in \mathcal{P}_k$, $y_2 \neq y_1$, then $y_1 \in \mathcal{P}_k$ is also nondominated by all $\hat{y} \notin \mathcal{P}_k$.*

Proof. In Lemma 3.1.1, it has already been proved that all the solutions $\hat{y} \notin \mathcal{P}_k$ are dominated by at least one solution $\tilde{y_2}$ contained in $\mathcal{P}_k$. On the other hand, the transitivity property of the domination character implies that

$$\tilde{y_2} \text{ dominates } \hat{y} \text{ and } \hat{y} \text{ dominates } y_1 \quad \Rightarrow \quad \tilde{y_2} \text{ dominates } y_1.$$

Looking at the contraposition of this statement, we obtain

$$y_1 \text{ is nondominated by } \tilde{y_2} \quad \Rightarrow \quad \begin{cases} \hat{y} \text{ is nondominated by } \tilde{y_2} \\ \text{or} \\ y_1 \text{ is nondominated by } \hat{y} \end{cases}$$

But we know that the first possibility is false since $\tilde{y_2}$ has been defined as one of the elements of $\mathcal{P}_k$ dominating $\hat{y}$. As a consequence, we have

$$y_1 \text{ is nondominated by } \tilde{y_2} \quad \Rightarrow \quad y_1 \text{ is nondominated by } \hat{y}.$$

As a result, if we manage to prove that $y_1 \in \mathcal{P}_k$ is nondominated by all $y_2 \in \mathcal{P}_k$, we thus prove in particular that y_1 is nondominated all $\tilde{y_2} \in \mathcal{P}_k$ that dominates one $\hat{y} \notin \mathcal{P}_k$ and finally we obtain that y_1 cannot be dominated by any $\hat{y} \notin \mathcal{P}_k$. □

Unfortunately, we cannot go further and say which solution of $\mathcal{P}_k$ is Pareto optimal without knowing the specific values of x_k, l, u and $\nabla_x f(x_k)$. Indeed, if we consider, for instance, a case where $x_k = (1;1), l = (0;0), u = (5;5)$ and $\nabla_x f(x_k) = (1;1)$, and $||.||_g = ||.||_l = ||.||_u = ||.||_1$, then $\mathcal{P}_k = \mathcal{Y}_E$ because $\mathcal{P}_k$ contains

$$\begin{cases} y_1 = (\Delta g_1 = (-1;-1) ; \Delta l_1 = (0;0) ; \Delta u_1 = (0;0)) \\ y_2 = (\Delta g_2 = (-1;0) ; \Delta l_2 = (0;1) ; \Delta u_2 = (0;0)) \\ y_3 = (\Delta g_3 = (0;-1) ; \Delta l_3 = (1;0) ; \Delta u_3 = (0;0)) \\ y_4 = (\Delta g_4 = (0;0) ; \Delta l_4 = (1;1) ; \Delta u_4 = (0;0)) \end{cases}$$

and we can check on the first graph of Figure 3.1 that none of its elements are dominated by any other one. But at the same time, if we consider rather the same case but where the gradient is equal to $\nabla_x f(x_k) = (1;2)$, then $\mathcal{P}_k$ contains

$$\begin{cases} y_1 = (\Delta g_1 = (-1;-2) ; \Delta l_1 = (0;0) ; \Delta u_1 = (0;0)) \\ y_2 = (\Delta g_2 = (-1;0) ; \Delta l_2 = (0;1) ; \Delta u_2 = (0;0)) \\ y_3 = (\Delta g_3 = (0;-2) ; \Delta l_3 = (1;0) ; \Delta u_3 = (0;0)) \\ y_4 = (\Delta g_4 = (0;0) ; \Delta l_4 = (1;1) ; \Delta u_4 = (0;0)) \end{cases}$$

and we can see on the second graph of Figure 3.1 that y_3 is dominated by y_2 (indeed $||\Delta g_3||_\infty = 2 > 1 = ||\Delta g_2||_\infty$, $||\Delta l_3||_1 = 1 = ||\Delta l_2||_1$ and $||\Delta u_3||_1 = 0 = ||\Delta u_2||_1$), leading to $\mathcal{Y}_E = \{y_1, y_2, y_4\} \subset \mathcal{P}_k$. In addition, the composition of $\mathcal{Y}_E$ depend on the chosen norms. Looking again at the last example, we see that if we take $||.||_g = ||.||_l = ||.||_u = ||.||_\infty$, we have $\mathcal{Y}_E = \{y_1, y_4\}$. In conclusion, many parameters play a role into the selection of the elements of $\mathcal{P}_k$ that compose $\mathcal{Y}_E$, which prevent us to characterize the efficient set more precisely a priori.

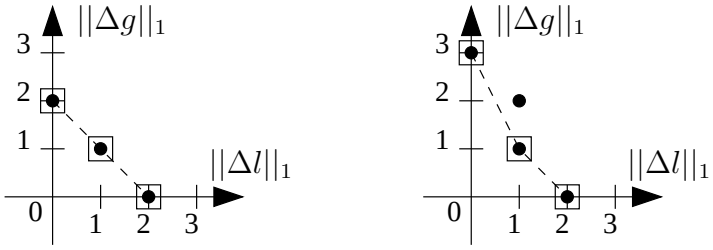


Figure 3.1: Each $f(y) = (||\Delta g||_1, ||\Delta l||_1, ||\Delta u||_1)$, $y \in \mathcal{P}_k$, is represented by a small dot, the elements of $\mathcal{Z}_N$ are surrounded by a square and the Pareto front is the dashed line. We do not represent Δu because $||\Delta u||_1 = 0$ for all $y \in \mathcal{P}_p$. We see that in the first figure no $y \in \mathcal{P}_k$ is dominated and that in the second figure $f(y_3) = (2, 1, 0)$ is dominated by $f(y_2) = (1, 1, 0)$.

3.3.3 The weighted sum scalarization

The Pareto optimal set is the set of all the feasible points that could be considered as the optimal solution of the multicriteria optimization problem. But the work is not finished when the efficient set has been identified. Of course, if it contains only one solution, then the problem is solved. Nevertheless, it is generally not the case and thus we still have to decide between all the solutions gathered in the Pareto optimal set. There are many ways to take on this choice (see Ehrgott (2005)). One option is to opt for the weighted sum scalarization technique. In this approach, we consider an alternative to the multicriteria optimization problem (3.35) by solving the following single objective problem

$$\min_{y \in \mathcal{Y}_k} \sum_{\ell=1}^{p} [\lambda]_\ell [f(y)]_\ell, \tag{3.39}$$

called the *weighted sum scalarization* of the original multicriteria optimization problem. Notice that the definition (3.1) of χ_k^{out} is precisely a weighted sum scalarization of the MCO problem (3.35) (but that it is not the case for χ_k^{in}). This technique has an interesting property in that an optimal solution of the weighted sum problem (3.39) with positive weights $[\lambda]_j$, for all j, is always efficient. As a consequence, and because the weights $\alpha_l, \alpha_u, \alpha_g$ have been chosen positive in the definition (3.1) of χ_k^{out}, all solutions y^* minimizing (3.1) are Pareto optimal, for all choices of $||.||_g, ||.||_l, ||.||_u$ and of $(\alpha_g, \alpha_l, \alpha_u) \in (0, 1]^3$, that is

$$\mathcal{S}_k^{out} \subseteq \mathcal{Y}_E \quad \text{and} \quad f(y^*) \in \mathcal{Z}_N \text{ for all } y^* \in \mathcal{S}_k^{out}.$$

Moreover, if $\mathcal{Z}_N$ is a convex set, then all efficient solutions are optimal solutions of scalarized problems with positive weights (see Section 3.1 of Ehrgott (2005) for a proof of these properties). Unfortunately, when using χ_k^{out}, we may not access all $z \in \mathcal{Z}_N$, even by trying all $\alpha_g, \alpha_l, \alpha u$, because $\mathcal{Z}_N$ may not be convex. Let us look at the following example : we consider a case where $x_k = (3; 4; 1), l = (0; 0; 0), u = (5; 5; 5)$

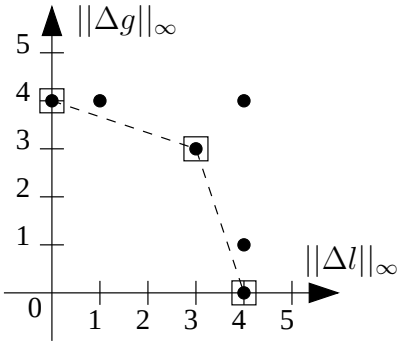


Figure 3.2: Each $f(y) = (||\Delta g||_\infty, ||\Delta l||_\infty, ||\Delta u||_\infty)$, $y \in \mathcal{P}_k$, is represented by a small dot, the elements of $\mathcal{Z}_N$ are surrounded by a square and the Pareto front is the dashed line. We do not represent Δu because $||\Delta u||_\infty = 0$ for all $y \in \mathcal{P}_p$. We see that the Pareto front is not convex so we cannot access $z_2 = (3; 3; 0)$.

and $\nabla_x f(x_k) = (4; 3; 1)$, and $||.||_g = ||.||_l = ||.||_u = ||.||_\infty$, then $\mathcal{P}_k$ contains

$$\left\{\begin{array}{lllllllllllll}
y_1 & = & (& \Delta g_1 & = & (-4;-3;-1) & ; & \Delta l_1 & = (0;0;0) & ; & \Delta u_1 & = (0;0;0) &) \\
y_2 & = & (& \Delta g_2 & = & (-4;-3;0) & ; & \Delta l_2 & = (0;0;1) & ; & \Delta u_2 & = (0;0;0) &) \\
y_3 & = & (& \Delta g_3 & = & (-4;0;-1) & ; & \Delta l_3 & = (0;4;0) & ; & \Delta u_3 & = (0;0;0) &) \\
y_4 & = & (& \Delta g_4 & = & (0;-3;-1) & ; & \Delta l_4 & = (3;0;0) & ; & \Delta u_4 & = (0;0;0) &) \\
y_5 & = & (& \Delta g_1 & = & (-4;0;0) & ; & \Delta l_1 & = (0;4;1) & ; & \Delta u_1 & = (0;0;0) &) \\
y_6 & = & (& \Delta g_2 & = & (0;-3;0) & ; & \Delta l_2 & = (3;0;1) & ; & \Delta u_2 & = (0;0;0) &) \\
y_7 & = & (& \Delta g_3 & = & (0;0;-1) & ; & \Delta l_3 & = (3;4;0) & ; & \Delta u_3 & = (0;0;0) &) \\
y_8 & = & (& \Delta g_4 & = & (0;0;0) & ; & \Delta l_4 & = (3;4;1) & ; & \Delta u_4 & = (0;0;0) &)
\end{array}\right.$$

$\mathcal{Y}_E = \{y_1, y_4, y_6, y_8\}$, leading to $\mathcal{Z}_N = \{z_1 = (4; 0; 0), z_2 = (3; 3; 0), z_3 = (0; 4; 0)\}$. These two sets and the Pareto front are represented in Figure 3.2. In this case, $z_2 = (3, 3, 0)$ cannot be reached by $\chi_k^{out,\infty}$, even when modifying the α's.

3.4 Criticality measures and convergence of RMTR$_\infty$

Now that we have discussed the advantages of different usual criticality measures, we are going back to the multilevel case and prove that the requirements made in Chapter 2 for the convergence of algorithm RMTR$_\infty$ are satisfied by at least two of them : χ_k^{tr} and $\chi_k^{in,1} = \chi_k^{out,1}$.

3.4.1 The condition of Lipschitz continuity on the criticality measures

The main property of the criticality measure that is required for the convergence is (2.27), that is it has to be Lipschitz continuous. The two following lemmas prove that the two criticality measures χ_k^{tr} and $\chi_k^{in,1}$ satisfy this property.

Lemma 3.4.1 *For all $x, y \in \mathcal{F}$, we have that*

$$|\chi^{tr}(x) - \chi^{tr}(y)| \leq \kappa_{\mathrm{L}} \|x - y\|_\infty.$$

with $\kappa_{\rm L} = 2(\kappa_H + \kappa_g)$, where κ_H and κ_g are defined by (2.24)-(2.25) and (2.26), respectively.

Proof. Let x and y be in $\mathcal{F}$. The optimization problem inside the definition (3.29) of χ^{tr} may be written as

$$\max_{\max(-1,l_i-x_i)\leq d_i\leq \min(1,u_i-x_i)} \langle -\nabla_x f(x), d\rangle. \tag{3.40}$$

Now denote by $m(x)$ the vector of average of the bounds on the variables in (3.40), whose i-th component is given by

$$m_i(x) = \tfrac{1}{2}[\max(-1, l_i - x_i) + \min(1, u_i - x_i)], \tag{3.41}$$

and by $r(x)$ the vector of "radii" whose i-th component is

$$r_i(x) = \tfrac{1}{2}[\min(1, u_i - x_i) - \max(-1, l_i - x_i)]. \tag{3.42}$$

Then, for $i = 1, \ldots n$,

$$2r_i(x) = 2|r_i(x)| =\leq |\min(1, u_i - x_i)| + |\max(-1, l_i - x_i)| \leq 2$$

and similarly, $2|m_i(x)| \leq 2$, which shows that both functions $|r_i(x)|$ and $|m_i(x)|$ are bounded by 1 for x in $\mathcal{F}$.

We now show that the functions $x \mapsto \min(1, u_i - x_i)$ and $x \mapsto \max(-1, l_i - x_i)$ are both unit Lipschitz continuous, that is Lipschitz continuous with constant 1. Consider x and y in $\mathcal{F}$, and define

$$\delta = |\min(1, u_i - x_i) - \min(1, u_i - y_i)|.$$

For $1 \leq u_i - x_i$ and $1 \leq u_i - y_i$, we have $\delta = 0$. If $1 \leq u_i - x_i$, and $1 \geq u_i - y_i$, we see that $1 - u_i + y_i \geq 0$, and that $x_i \leq y_i$. Therefore we have that

$$\delta = |1 - u_i + y_i| = 1 - u_i + y_i \leq u_i - x_i - u_i + y_i = y_i - x_i = |x_i - y_i|,$$

and we also deduce by symmetry that $\delta = |x_i - y_i|$ whenever $1 \geq u_i - x_i$, and $1 \leq u_i - y_i$. Finally, if $1 \geq u_i - x_i$, and $1 \geq u_i - y_i$, we obtain that

$$\delta = |u_i - x_i - u_i + y_i| = |x_i - y_i|.$$

Hence the function $x \mapsto \min(1, u_i - x_i)$ is unit Lipschitz continuous. The result for $x \mapsto \max(-1, l_i - x_i)$ is obtained from the same arguments. Combining these results with (3.41) and (3.42), we obtain that both $r_i(x)$ and $m_i(x)$ are also unit Lipschitz continuous.

Now defining $\tilde{d}$ such that $d = m(x) + r(x) \circ \tilde{d}$ where $\circ$ is the (Hadamard) componentwise product, i.e. $x \circ y = [x_1 y_1, \ldots, x_n y_n]^T$, we observe that the minimization problem (3.40) may also be written as

$$\max_{\|\tilde{d}\|_\infty \leq 1} \langle -\nabla_x f(x), m(x) + r(x) \circ \tilde{d}\rangle,$$

whose solution is then analytically given by

$$\chi^{tr}(x) = \langle -\nabla_x f(x), m(x)\rangle + \|\nabla_x f(x) \circ r(x)\|_1.$$

Using this formula, we now show that $\chi(x)$ is Lipschitz continuous in $\mathcal{F}$. From the mean-value theorem, we know that

$$\nabla_x f(x) = \nabla_x f(y) + G_{[x,y]}(x-y), \tag{3.43}$$

where $G_{[x,y]} = \int_0^1 \nabla_x^2 f(x + t(y-x))\, dt$ is the Hessian matrix computed along the direction linking x and y and where , from (2.24),

$$\|G_{[x,y]}\|_{\infty,1} = \| \int_0^1 \nabla_x^2 f(x + t(y-x))\, dt\|_{\infty,1} \leq \max_{z\in[x,y]} \|\nabla_x^2 f(z)\|_{\infty,1} \leq \kappa_{\rm H}. \tag{3.44}$$

Hence, using $|\langle u, v\rangle| \leq \|u\|_1 \|v\|_\infty$, the inequality $\|m(x)\|_\infty \leq 1$, (2.26) and the unit Lipschitz continuity of $m(x)$, we obtain that

$$\begin{aligned} &|\langle \nabla_x f(x), m(x)\rangle - \langle \nabla_x f(y), m(y)\rangle| \\ &\qquad \leq |\langle \nabla_x f(x) - \nabla_x f(y), m(x)\rangle + \langle \nabla_x f(y), m(x) - m(y)\rangle| \\ &\qquad \leq (\kappa_{\rm H} + \kappa_{\rm g})\|x-y\|_\infty. \end{aligned}$$

In addition,

$$\begin{aligned} &\|\nabla_x f(x) \circ r(x)\|_1 - \|\nabla_x f(y) \circ r(y)\|_1 \\ &\qquad \leq \|\nabla_x f(x) \circ r(x) - \nabla_x f(y) \circ r(y)\|_1 \\ &\qquad \leq \|\nabla_x f(x) \circ (r(x) - r(y))\|_1 + \|(\nabla_x f(x) - \nabla_x f(y)) \circ r(y)\|_1. \end{aligned}$$

Using now the inequality $\|u \circ v\|_1 \leq \|u\|_1 \|v\|_\infty$, we obtain from $\|r(y)\|_\infty \leq 1$, (3.43) and (3.44) that

$$\|(\nabla_x f(x) - \nabla_x f(y)) \circ r(y)\|_1 \leq \|\nabla_x f(x) - \nabla_x f(y)\|_1 \|r(y)\|_\infty \leq \kappa_{\rm H}\|x-y\|_\infty$$

and, similarly, from the unit Lipschitz continuity of $r(x)$, and (2.26), that

$$\|\nabla_x f(x) \circ (r(x) - r(y))\|_1 \leq \|\nabla_x f(x)\|_1 \|r(x) - r(y)\|_\infty \leq \kappa_{\rm g}\|x-y\|_\infty.$$

Putting together the above results yields that $|\chi(x) - \chi(y)| \leq 2(\kappa_{\rm H} + \kappa_{\rm g})\|x-y\|_\infty$. □

Lemma 3.4.2 *For all $x, y \in \mathcal{F}$, we have that*

$$|\chi^{in,1}(x) - \chi^{in,1}(y)| \leq \kappa_{\rm L}||x-y||_\infty.$$

with $\kappa_{\rm L} = (\kappa_H + 2n)$, where κ_H is defined by (2.24)-(2.25).

Proof. Making explicit the definitions of $\chi^{in,1}(x)$ and the 1-norm, using $|u+v| \leq |u| + |v|$, $||u| - |v|| \leq |u - v|$, and $|[\mathrm{Proj}_{\mathcal{F}}(u)]_j - [\mathrm{Proj}_{\mathcal{F}}(v)]_j| \leq |[u]_j - [v]_j|$ for all j

because $\mathcal{F}$ is a bound-constrained set, we obtain

$$\begin{aligned}
|\chi^{in,1}(x) - \chi^{in,1}(y)| &= |||\mathrm{Proj}_{\mathcal{F}}(x - \nabla_x f(x)) - x||_1 - ||\mathrm{Proj}_{\mathcal{F}}(y - \nabla_x f(y)) - y||_1| \\
&= \left| \sum_{j=1}^{n} |[\mathrm{Proj}_{\mathcal{F}}(x - \nabla_x f(x)) - x]_j| - |[\mathrm{Proj}_{\mathcal{F}}(y - \nabla_x f(y)) - y]_j| \right| \\
&\leq \sum_{j=1}^{n} |[\mathrm{Proj}_{\mathcal{F}}(x - \nabla_x f(x)) - x]_j - [\mathrm{Proj}_{\mathcal{F}}(y - \nabla_x f(y)) - y]_j| \\
&\leq \sum_{j=1}^{n} [|\mathrm{Proj}_{\mathcal{F}}(x - \nabla_x f(x)) - \mathrm{Proj}_{\mathcal{F}}(y - \nabla_x f(y))| + |y - x|]_j \\
&\leq \sum_{j=1}^{n} [|\nabla_x f(y) - \nabla_x f(x)| + 2|y - x|]_j.
\end{aligned}$$

Now the mean-value theorem implies

$$\nabla_x f(x) = \nabla_x f(y) + G_{[x,y]}(x - y)$$

where, from (2.24),

$$\|G_{[x,y]}\|_{\infty,1} = \| \int_0^1 \nabla_x^2 f(x + t(y - x))\, dt\|_{\infty,1} \leq \max_{z \in [x,y]} \|\nabla_x^2 f(z)\|_{\infty,1} \leq \kappa_{\mathrm{H}}.$$

We therefore have

$$\begin{aligned}
|\chi^{in,1}(x) - \chi^{in,1}(y)| &\leq \sum_{j=1}^{n} [|G_{[x,y]}(x - y)| + 2|y - x|]_j \\
&= \left|\left| |G_{[x,y]}(x - y)| + 2|y - x| \right|\right|_1 \\
&\leq \left|\left| G_{[x,y]}(x - y) \right|\right|_1 + 2\, ||y - x||_1 \\
&\leq \left|\left| G_{[x,y]} \right|\right|_{\infty,1} ||y - x||_\infty + 2\, ||y - x||_1 \\
&\leq (\kappa_{\mathrm{H}} + 2n)\, ||y - x||_\infty
\end{aligned}$$

□

3.4.2 The second condition on the criticality measures

A second condition (2.28) has been imposed on the criticality measures and it is shown by the next lemma that it holds for χ^{tr}.

Lemma 3.4.3 *We have*

$$\chi^{tr}_{i-1,0} = \chi^{tr}(x_{i-1,0}) \leq 2\kappa_g \Delta_{i,k} \qquad \textit{for all } k, \textit{ for all } i = 1, \ldots, r.$$

Proof. Since (2.34) implies that $2\Delta_{i,k} \leq 1$, we deduce from (2.32) (with $x = x_{i-1,0} + d \in \mathcal{L}_{i-1}$) that $\mathcal{L}_{i-1} \subseteq \{x_{i-1,0} + d \mid \|d\|_\infty \leq 1\}$ and thus that

$$\chi^{tr}_{i-1,0} = |\min_{x_{i-1,0} + d \in \mathcal{L}_{i-1}} \langle g_{i-1,0}, d\rangle| = |\langle g_{i-1,0}, d_{i-1,0}\rangle|$$

with

$$\|d_{i-1,0}\|_\infty \leq 2\Delta_{i,k}.$$

Then, these two results combined with the inequality $|\langle u, v\rangle| \leq \|u\|_1 \|v\|_\infty$ and (2.26) give

$$\chi^{tr}_{i-1,0} = |\langle g_{i-1,0}, d_{i-1,0}\rangle| \leq \|g_{i-1,0}\|_1 \|d_{i-1,0}\|_\infty \leq 2\kappa_{\rm g}\Delta_{i,k}.$$

□

Moreover, the condition (2.28) is satisfied by $\chi^{out,1}$, up to a multiplicative constant equal to the dimension n_r of the problem. The appearance of this constant does not prevent the convergence of the algorithm RMTR$_\infty$ because it suffices to define

$$\kappa_2 \stackrel{\rm def}{=} \frac{1}{2}\min\left[1, \frac{\epsilon_{\rm min}}{2n_r\kappa_{\rm g}}, \Delta^{\rm s}_{\rm min}\right] \in (0,1)$$

in Lemma 2.3.4 to adapt the convergence proof of Chapter 2. However, the third remark after Corollary 2.3.9 does not hold anymore, that is our algorithm is not proved to be mesh-independant if $\chi^{out,1}$ is chosen. But it is not a big loss since the assumption (2.26) on the gradient (with $\kappa_{\rm g}$ independant of the dimension n_r), needed to have this result when using χ^{tr}, is already very restrictive.

Lemma 3.4.4 *We have*

$$\chi^{in,1}_{i-1,0} = \chi^{out,1}_{i-1,0} = \chi^{out,1}(x_{i-1,0}) \leq 2n_r\kappa_g\Delta_{i,k} \qquad \textit{for all } k, \ \textit{for all } i = 1, \ldots, r.$$

Proof. Because of (2.32), we have

$$\begin{aligned}
\chi^{out,1}_{i-1,0} &= ||{\rm Proj}_{\mathcal{L}_{i-1}}(x_{i-1,0} - g_{i-1,0}) - x_{i-1,0}||_1 \\
&\leq n_{i-1}||{\rm Proj}_{\mathcal{L}_{i-1}}(x_{i-1,0} - g_{i-1,0}) - x_{i-1,0}||_\infty \\
&\leq 2n_r\Delta_{i,k} \\
&\leq 2n_r\kappa_{\rm g}\Delta_{i,k}.
\end{aligned}$$

□

3.4.3 Active constraints identification

In Section 2.4, we have left for later the proof of Theorem 2.4.4 because the criticality measure has not yet been defined. We are now ready to prove this result, both for χ^{tr} and for $\chi^{in,1}$.

Theorem 3.4.5 *There exists $k_2 \geq k_1$ (where k_1 is defined in Theorem 2.4.3) such that, if there is a $j \in \{1, ..., m\}$ with*

$$j \in \mathcal{A}({\rm L}_{*k}) \textit{ and } j \notin \mathcal{A}(x^{SD}_k) \tag{3.45}$$

for some $k \geq k_2$, then

$$\pi^{SD}_k \geq \epsilon_* \tag{3.46}$$

where

$$\pi^{SD}_k = \min\{1, \chi^{SD}_k\} = \min\{1, |\min_{\substack{x^{SD}_k + d \in \mathcal{F}_{\mathcal{A}\left(x^{SD}_k\right)} \\ ||d||_\infty \leq 1}} < g_k, d > |\}$$

and

$$\mathcal{F}_{\mathcal{A}(x)} = \{x \in I\!R^n : c_i(x) \geq 0 \; \forall i \in \mathcal{A}(x)\} \subseteq \mathcal{F} \text{ for all } x \in \mathcal{F},$$

for some $\epsilon_* \in (0,1)$ *independent of* k *and* j.

Proof. The proof is mainly the same as in Conn et al., 2000 as the assumptions are independent of the method, but as it handles the criticality measure, the choice of infinity norm implies small changes.
Consider a given $x_* \in \mathrm{L}_*$, where L_* is compact, such that $\mathcal{A}(x_*) \neq \emptyset$ and a given $i \in \mathcal{A}(x_*)$. Consider then the quantity

$$\chi_{*i}(x_*) \stackrel{def}{=} \left| \min_{\substack{x_* + d \in \mathcal{F}_{\{i\}} \\ ||d||_\infty \leq \frac{1}{2}}} \langle \nabla_x f(x_{*k}), d \rangle \right|,$$

where $\mathcal{F}_{\{i\}}$ is defined by

$$\mathcal{F}_{\{i\}} \stackrel{def}{=} \cap_{j \in \{1,\ldots,m\} \setminus \{i\}} [\mathcal{F}]_j.$$

Because of the definition of $\mathcal{F}$ and the assumptions made about it, one has that $\chi_{*i}(x_*) > 0$ for all choices of $x_* \in \mathrm{L}_*$ and $i \in \mathcal{A}(x_*)$. The continuity of the projection operator and the continuity of $\nabla_x f(.)$ also guarantee that $\chi_{*i}(.)$ is continuous. We first minimize $\chi_{*i}(x_*)$ on the compact set of all $x_* \in \mathrm{L}_*$ such that $i \in \mathcal{A}(x_*)$. For each such set, this produces a strictly positive result. We next take the smallest of these results over all i such that $i \in \mathcal{A}(x_*)$ for some $x_* \in \mathrm{L}_*$, yielding a strictly positive lower bound $2\epsilon_*$. In short,

$$\min_i \min_{x_*} \chi_{*i}(x_*) \geq 2\epsilon_* \tag{3.47}$$

for some $\epsilon_* \in (0,1)$ independent of k, j and δ. Now reduce δ, if necessary, to ensure that

$$\kappa_H \delta \leq \epsilon_*, \tag{3.48}$$

and consider $k \geq k_1$. Then, by Theorem 2.4.3, we can associate with x_k a compact connected set of limit points L_{*k} such that (2.82) holds. We then select a particular $x_{*k} \in \mathrm{L}_{*k} \cap \mathcal{V}(\{x_k\}, \delta)$, which ensures that

$$\{x_{*k} + d \in \mathcal{F}_{\{i\}} |\; ||d||_\infty \leq \frac{1}{2}\} \subset \{x_k + d \in \mathcal{F}_{\{i\}} |\; ||d||_\infty \leq 1\} \tag{3.49}$$

for all $i \in \{1, ..., m\}$, where we used the bound $\delta \leq \frac{1}{2}$ coming from Theorem 2.4.3.

Given a $k \geq k_1$ and such that x_k satisfies (3.45), we now distinguish two cases. The first is when $\pi_k^{SD} \geq \chi_{*j}(x_{*k})$, in which case (3.46) immediately follows from (3.47). The second is when $\pi_k^{SD} < \chi_{*j}(x_{*k})$. If $\chi_k^{SD} \geq 1$, then $\pi_k^{SD} = 1$ by definition, and (3.46) again follows since $\epsilon_* \in (0,1)$. Suppose therefore that $\chi_k^{SD} < 1$, in which case $\pi_k^{SD} = \chi_k^{SD}$, and define d_k^{SD} and d_{*k} as two vectors satisfying

$$\pi_k^{SD} = -\langle g_k, d_k^{SD} \rangle, \quad ||d_k^{SD}||_\infty \leq 1, \quad x_k + d_k \in \mathcal{A}(x_k)$$

and

$$\chi_{*j}(x_*) = -\langle \nabla_x f(x_{*k}), d_{*k} \rangle, \quad ||d_{*k}||_\infty \leq \frac{1}{2}, \quad x_{*k} + d_{*k} \in \mathcal{F}_{\{j\}}.$$

We can write, using the Cauchy-Schwarz inequality, that

$$\begin{array}{rcl} 0 & < & \chi_{*j}(x_*) - \pi_k^{SD} \\ & = & \langle g_k, d_k^{SD} \rangle - \langle \nabla_x f(x_{*k}), d_{*k} \rangle \\ & = & \langle g_k, d_k^{SD} - d_{*k} \rangle + \langle g_k - \nabla_x f(x_{*k}), d_{*k} \rangle \\ & \leq & \langle g_k, d_k^{SD} - d_{*k} \rangle + ||g_k - \nabla_x f(x_{*k})||_1 ||d_{*k}||_\infty \\ & \leq & \langle g_k, d_k^{SD} - d_{*k} \rangle + \frac{1}{2} ||g_k - \nabla_x f(x_{*k})||_1. \end{array} \tag{3.50}$$

Now, combining (3.49) and the definitions of π_k^{SD}, d_k^{SD} and d_{*k}, we obtain that

$$\langle g_k, d_k^{SD} \rangle = -\pi_k^{SD} \leq \langle g_k, d_{*k} \rangle.$$

Substituting this last inequality in (3.50), we choose $k_2 \geq k_1$ sufficiently large to ensure that, for $k \geq k_2$,

$$\begin{array}{rcl} 0 < \chi_{*j}(x_*) - \pi_k^{SD} & \leq & ||g_k - \nabla_x f(x_{*k})||_1 \\ & = & ||\nabla_x f(x_k) - \nabla_x f(x_{*k})||_1 \\ & \leq & \kappa_H ||x_k - x_{*k}||_\infty \\ & \leq & \kappa_H \delta \\ & \leq & \epsilon_*, \end{array}$$

where we used AM.3, the mean-value theorem, AF.3, the definition of x_{*k}, and (3.48). The inequality (3.46) then follows from (3.47). □

Theorem 3.4.6 *There exists $k_2 \geq k_1$ (where k_1 is defined in Theorem 2.4.3) such that, if there is a $j \in \{1, ..., m\}$ with*

$$j \in \mathcal{A}(\mathrm{L}_{*k}) \text{ and } j \notin \mathcal{A}(x_k^{SD}) \tag{3.51}$$

for some $k \geq k_2$, then

$$\pi_k^{SD} \geq \epsilon_* \tag{3.52}$$

where

$$\pi_k^{SD} = \min\{1, \chi_k^{SD}\} = \min\{1, ||\mathrm{Proj}_{\mathcal{F}_{\mathcal{A}(x)}}(x_k^{SD} - \nabla_x(x_k^{SD})) - x_k^{SD}||_1\}$$

and

$$\mathcal{F}_{\mathcal{A}(x)} = \{x \in I\!R^n : c_i(x) \geq 0 \ \forall i \in \mathcal{A}(x)\} \subseteq \mathcal{F} \text{ for all } x \in \mathcal{F},$$

for some $\epsilon_* \in (0, 1)$ independent of k and j.

Proof. The proof is mainly the same as in Conn et al., 2000 as the assumptions are independent of the method, but we have chosen another criticality measure and its definition is central in this theorem. Moreover, the choice of infinity norm implies small changes.
Consider a given $x_* \in \mathrm{L}_*$, where L_* is compact, such that $\mathcal{A}(x_*) \neq \emptyset$ and a given $i \in \mathcal{A}(x_*)$. Consider then the quantity

$$\chi_{*i}(x_*) \stackrel{def}{=} ||\mathrm{Proj}_{\mathcal{F}_{\{i\}}}(x_* - \nabla_x(x_*)) - x_*||_1$$

where $\mathcal{F}_{\{i\}}$ is defined by

$$\mathcal{F}_{\{i\}} \stackrel{def}{=} \cap_{j\in\{1,\ldots,m\}\setminus\{i\}}[\mathcal{F}]_j.$$

Because of the definition of $\mathcal{F}$ and the assumptions made about it, one has that $\chi_{*i}(x_*) > 0$ for all choices of $x_* \in \mathrm{L}_*$ and $i \in \mathcal{A}(x_*)$. The continuity of the projection operator and the continuity of $\nabla_x f(.)$ also guarantee that $\chi_{*i}(.)$ is continuous. We first minimize $\chi_{*i}(x_*)$ on the compact set of all $x_* \in \mathrm{L}_*$ such that $i \in \mathcal{A}(x_*)$. For each such set, this produces a strictly positive result. We next take the smallest of these results over all i such that $i \in \mathcal{A}(x_*)$ for some $x_* \in \mathrm{L}_*$, yielding a strictly positive lower bound $2\epsilon_*$. In short,

$$\min_i \min_{x_*} \chi_{*i}(x_*) \geq 2\epsilon_* \tag{3.53}$$

for some $\epsilon_* \in (0,1)$ independent of k, j and δ. Now reduce δ, if necessary, to ensure that

$$\kappa_H \delta \leq \epsilon_*, \tag{3.54}$$

and consider $k \geq k_1$. Then, by Theorem 2.4.3, we can associate with x_k a compact connected set of limit points L_{*k} such that (2.82) holds. We then select a particular $x_{*k} \in \mathrm{L}_{*k} \cap \mathcal{V}(\{x_k\}, \delta)$.

Given a $k \geq k_1$ and such that x_k satisfies (2.83), we now distinguish two cases. The first is when $\pi_k^{SD} \geq \chi_{*j}(x_{*k})$, in which case (2.84) immediately follows from (3.53). The second is when $\pi_k^{SD} < \chi_{*j}(x_{*k})$. If $\chi_k^{SD} \geq 1$, then $\pi_k^{SD} = 1$ by definition, and (2.84) again follows since $\epsilon_* \in (0,1)$. Suppose therefore that $\chi_k^{SD} < 1$, in which case $\pi_k^{SD} = \chi_k^{SD}$. In that case we have

$$\begin{aligned}
0 < \chi_{*j}(x_*) - \pi_k^{SD} &= |\,||\mathrm{Proj}_{\mathcal{F}_{\mathcal{A}(x)}}(x_k^{SD} - \nabla_x(x_k^{SD})) - x_k^{SD}||_1 \\
&\quad -||\mathrm{Proj}_{\mathcal{F}_{\{i\}}}(x_* - \nabla_x(x_*)) - x_*||_1\,| \\
&\leq ||\mathrm{Proj}_{\mathcal{F}_{\mathcal{A}(x)}}(x_k^{SD} - \nabla_x(x_k^{SD})) - x_k^{SD} \\
&\quad -(\mathrm{Proj}_{\mathcal{F}_{\{i\}}}(x_* - \nabla_x(x_*)) - x_*)||_1 \\
&\leq ||\nabla_x f(x_k^{SD}) - \nabla_x f(x_{*k})||_1 \\
&\leq \kappa_H ||x_k - x_{*k}||_\infty \\
&\leq \kappa_H \delta \\
&\leq \epsilon_*,
\end{aligned}$$

where we used $|\,||a|| - ||b||\,| \leq ||a-b||$, $\mathcal{F}_{\mathcal{A}(x)} \subseteq \mathcal{F}_{\{i\}}$, the fact that the distance between two vectors is always greater than the distance between their projections on imbricated convex sets, the mean-value theorem, the boundedness of the Hessian matrix, the definition of x_{*k}, and (3.54). The inequality (2.84) then follows from (3.53). □

3.5 Conclusion

In this chapter, we have shown that some well known criticality measures for bound-constrained optimization correspond to the backward error of the original problem (2.2) in some specific norm. We have also looked at some interpretation of this backward error in terms of multicriteria optimization. We finally have proved that the criticality measures studied in this chapter satisfy the properties needed for the convergence of RMTR_∞ and the identification of active constraints.

Chapter 4

Numerical experiments

In this chapter, we first define a practical algorithm that is used to select a set of suitable parameters for our method. This default version of the algorithm is then challenged to other classical algorithms. Finally, we compare the criticality measures discussed in the previous chapter.

4.1 A practical algorithm

Our algorithm description so far leaves a number of practical choices unspecified. It is the purpose of this section to provide the missing details of the particular implementation for which numerical performance is reported in this chapter. These details are of course influenced by our focus on discretized problems, where the different levels correspond to different discretization grids, from coarser to finer.

4.1.1 Taylor iterations: smoothing and solving

The most important issue is how to enforce sufficient decrease at Taylor iterations, that is, when Step 3 of Algorithm 2.2.1 is executed. At the coarsest level ($i = 0$), the cost of fully minimizing (2.16) inside the trust region remains small, since the subproblem is of low dimension. We thus solve the subproblem using the PTCG (Projected Truncated Conjugate-Gradient) algorithm designed for the standard trust-region algorithm (see Conn et al., 1988, or Conn, Gould and Toint, 1992).

At finer levels ($i > 0$), we use an adaptation of multigrid smoothing techniques. The main characteristics of multigrid algorithms (see Briggs et al., 2000) are based on the observation that different *frequencies* are present in the initial error on the solution of the finest grid problem (or even of the infinite-dimensional one), and become only progressively visible in the hierarchy from coarse to fine grids. Low frequencies are visible from coarse grids and up, but higher ones can only be distinguished when the mesh-size of the grid becomes comparable to the frequency in question. In multigrid strategies, some algorithms, called *smoothers*, are known to very efficiently reduce the high frequency components of the error on a grid (that is, in most cases, the components whose "wavelength" is comparable to the grid size). But these algorithms have little effect on the low frequency error components. It is observed however that such components on a fine grid *appear* more oscillatory on a coarser grid. They may thus be viewed as high frequency components on some coarser grid

and be in turn reduced by a smoother. Moreover, this is done at a lower cost since computations on coarser grids are typically much cheaper than on finer ones. The multigrid strategy consists therefore in alternating between solving the problem on coarse grids, essentially annihilating low frequency components of the error, and on fine grids, where high frequency components are reduced (at a higher cost). This last operation is often called *smoothing* because the effect of reducing high frequency components without altering much the low frequency ones has a *smoothing effect* of the error's behavior. The effect of the smoothing is illustrated in Figure 4.1. In what follows, we adapt the multigrid smoothing technique to the computation of a Taylor step satisfying the requirements of Step 3 of Algorithm RMTR$_\infty$.

A very well-known multigrid smoothing technique is the Gauss-Seidel method, in which each equation of the Newton system is solved in succession[(1)]. To extend this procedure to our case, rather than successively solving equations, we perform successive one-dimensional bound-constrained minimizations of the model (2.16) along the coordinate axes, provided the curvature of this model along each axis is positive. More precisely, consider the minimization of (2.16) at level i along the j-th axis (starting each minimization from s such that $\nabla m_{i,k}(x_{i,k}+s) \stackrel{\text{def}}{=} g$). Then, provided that the j-th diagonal entry of $H_{i,k}$ is positive, the j-th one-dimensional minimization then results in the updates

$$\alpha_j = \text{Proj}_{\mathcal{W}_{i,k}}(-[g]_j/[H_{i,k}]_{jj}), \quad [s]_j \leftarrow [s]_j + \alpha_j \text{ and } g \leftarrow g + \alpha_j H_{i,k} e_{i,j}, \tag{4.1}$$

where $\text{Proj}_{\mathcal{W}_{i,k}}(.)$ is the orthogonal projection on the feasible set at level i, that is on $\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{A}_i \cap \mathcal{B}_{i,k}$, and where $e_{i,j}$ is the j-th vector of the canonical basis of $\mathbb{R}^{n_i}$. If, on the other hand, $[H_{i,k}]_{jj} \leq 0$, then a descent step is made along the j-th coordinate axis until the boundary of $\mathcal{W}_{i,k}$ is reached and the model gradient is updated accordingly. This process is the well-known Sequential Coordinate Minimization (SCM) (see, for instance, Ortega and Rheinboldt (1970), Section 14.6), which we adapted to handle bound constraints. In what follows, we refer to a set of n_i successive unidimensional minimizations as a *smoothing cycle*. A SCM *smoothing iteration* then consists of one or more of these cycles.

In order to enforce convergence to first-order points, we still have to ensure that a sufficient model decrease (2.21) has been obtained within the trust region after one or more complete smoothing cycles. To do so, we start the first smoothing cycle by selecting the axis corresponding to the index j_m such that

$$j_m = \underset{j}{\text{argmin}} \quad [g_{i,k}]_j [d_{i,k}]_j, \tag{4.2}$$

where

$$d^*_{i,k} = \underset{\substack{x_{i,k}+d \in \mathcal{L}_i \\ \|d\|_\infty \leq 1}}{\text{argmin}} \quad \langle g_{i,k}, d\rangle. \tag{4.3}$$

[(1)] See Briggs et al., 2000, page 10, or Golub and Van Loan, 1989, page 510, or Ortega and Rheinboldt, 1970, page 214, amongst many others.

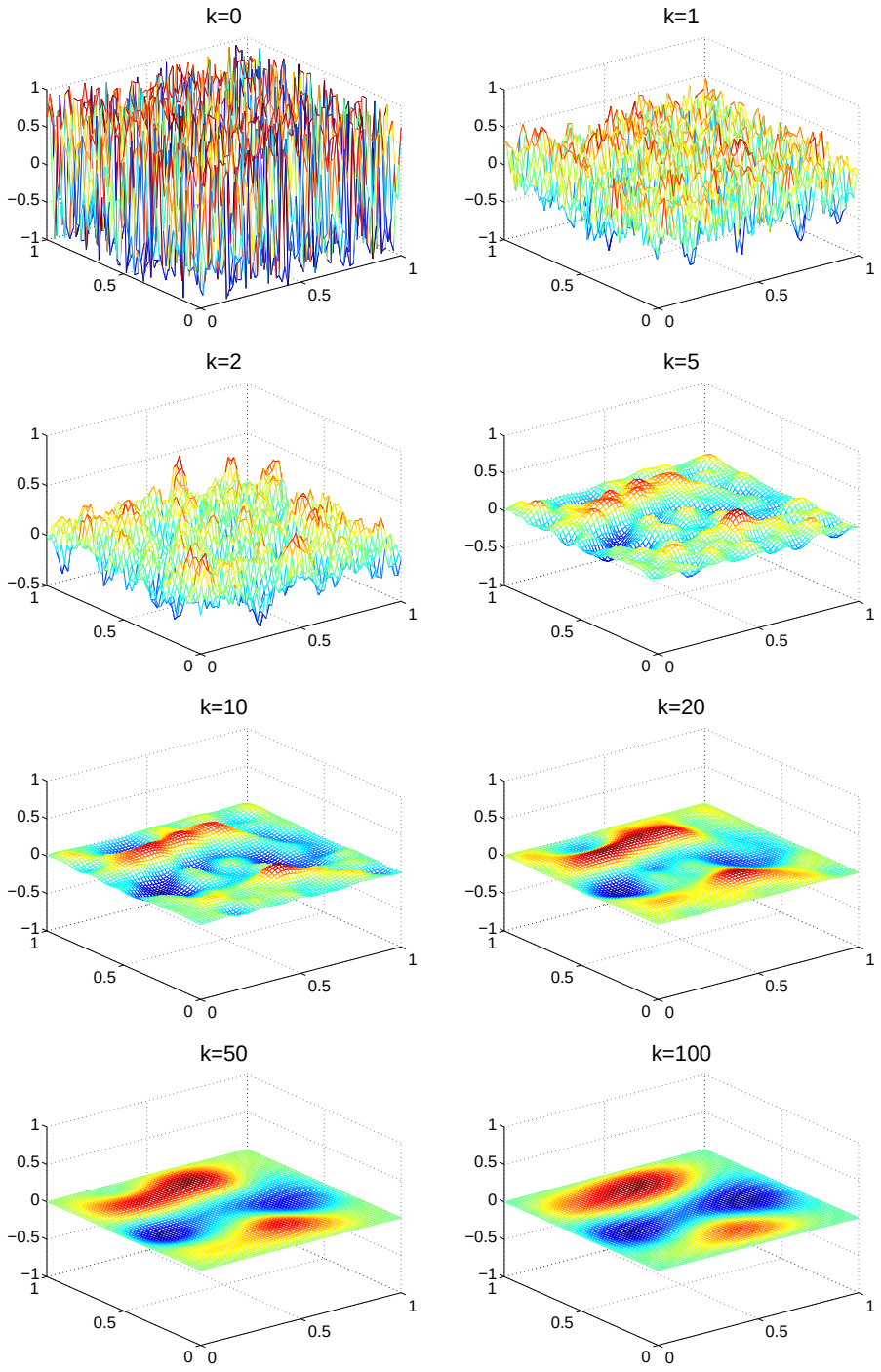


Figure 4.1: Illustration of the smoothing effect on the error of quadratic problem of 3969 variables along the iterations k, when starting from a random starting point whose components are all in (-1,1). All the very high-frequencies of the error are removed in only 5 iterations, the error is progressively "smoothed" but the low-frequencies need a huge number of iterations to be removed.

Algorithm 4.1.1: SCM Smoothing

Step 0: Initialization. Define $s_{i,k} = s^+_{i,k} = 0$, $x_{i,k+1} = 0$.

For each direction $j = 0, ..., n$ do

Step 1: Computation of the step. If $[H_k]_{j,j} > 0$ compute $[s^+_{i,k}]_j = \frac{[-g_{i,k}]_j}{[H_{i,k}]_{j,j}}$. Otherwise set

$$[s^+_{i,k}]_j = \begin{cases} [l(\mathcal{W}_{i,k}) - x_k]_j & \text{if } [g_{i,k}]_j > 0 \\ [u(\mathcal{W}_{i,k}) - x_k]_j & \text{if } [g_{i,k}]_j < 0 \\ 0 & \text{if } [g_{i,k}]_j = 0 \end{cases}$$

where $l(\mathcal{W}_{i,k})$ and $u(\mathcal{W}_{i,k})$ are the lower and upper bounds of $\mathcal{W}_{i,k}$, respectively.

Step 2: Projection on the feasible set. Define $[x_{i,k+1}]_j = \text{Proj}_{[\mathcal{W}_{i,k}]_j}([x_{i,k}]_j + [s^+_{i,k}]_j)$ and $[s_{i,k}]_j = [x_{i,k+1}]_j - [x_{i,k}]_j$, where $\text{Proj}_{[\mathcal{W}_{i,k}]_j}$ is the orthogonal projection on the feasible set in the j^{th} direction.

Step 3: Gradient update For all directions ℓ update $[g_{i,k}]_\ell = [g_{i,k}]_\ell + [H_{i,k}]_{\ell,j}[s_{i,k}]_j$.

if χ^{tr} has been chosen as a criticality measure, and

$$j_m = \arg\max_j \left| [\Gamma_{i,k}(\alpha_l, \alpha_u)]_j \right|,$$

where $\Gamma_{i,k}(\alpha_l, \alpha_u)$ is defined in (3.21) with $\mathcal{C} = \mathcal{L}_i$, if $\chi^{out,1}$ has been preferred. Indeed in both cases the step $s_{i,k}$ obtained after one smoothing cycle is then such that (2.21) holds. We follow here the reasoning in the case where $\chi_{i,k} = \chi^{tr}_k$, but we get similar results using $\chi^{out,1}_k$, as shown in Appendix B.1. In the aim of making the following reasoning easier to read, we suppress the first subscript i of all the quantities until then end of this section.

Theorem 4.1.1 *A cycle of Gauss-Seidel relaxation applied when minimizing on the model (2.16) on a bound-constrained set, and beginning the minimization sequence on the direction j_m defined by (4.2) produces a decrease bounded below by*

$$\Delta m_k \geq \kappa_{gen} \chi^{tr}_k \min\left[\frac{\chi^{tr}_k}{\beta_k}, \Delta_k, 1\right],$$

where $\kappa_{gen} \in (0, \frac{1}{2})$.

Proof. The definition of j_m and the box shape of the feasible set imply

$$\begin{aligned}
\chi_k^{tr} &= \sum_{j=1}^{n} -[g_k]_j [d_k^*]_j \\
&\leq -n[g_k]_{j_m}[d_k^*]_{j_m} \\
&= n \left| \min_{\substack{[l(\mathcal{W})]_{j_m} \leq [x_k]_{j_m} + d_{k,j_m} \leq [u(\mathcal{W})]_{j_m} \\ |d_{k,j_m}| \leq 1}} [g_k]_{j_m} d_{k,j_m} \right| \\
&\stackrel{def}{=} n\chi_{k,j_m}. \qquad (4.4)
\end{aligned}$$

On another hand, as we minimize exactly the model m_k in the j_m^{th} direction, the sufficient decrease property is respected in this direction, which gives (see Conn et al., 2000)

$$\Delta m_k^m \geq \kappa_{dcp} \chi_{k,j_m} \min\left[\frac{\chi_{k,j_m}}{\beta_k}, \Delta_k, 1\right],$$

with $\kappa_{dcp} \in (0, \frac{1}{2})$. Then, using (4.4), and as each unidirectional minimization does not increase the model (notice that the projection on the feasible set does not increse the objective function since we are minimizaing a quadratic function in a one dimensional space), which implies $\Delta m_k^j \geq 0$ for all j, the total decrease is

$$\Delta m_k \geq \Delta m_k^m \geq \kappa_{gen} \chi_k^{tr} \min\left[\frac{\chi_k^{tr}}{\beta_k}, \Delta_k, 1\right],$$

where $\kappa_{gen} \stackrel{def}{=} \frac{\kappa_{dcp}}{n^2} \in (0, \frac{1}{2})$. □

Notice that this proof also works if we use another criticality measure, defined like χ_{tr} but where the set of level-dependent bounds $\mathcal{L}$ is replaced by the set of active constraints of this set only. This comes from the fact that the definition of the set $\mathcal{L}$ does not play any role in the proof, except in the definition of the criticality measure. This Corollary implies that Gauss-Seidel steps satisfy (2.80) (with $\pi_k^{SD} = \pi_k^{tr,a}$) and thus that the active constraints identification theory holds when using a Gauss-Seidel step.

Corollary 4.1.2 *When $\mathcal{L}_i$ is replaced by $\mathcal{L}_{\mathcal{A}(x_k)} \subseteq \mathcal{L}_i$ the set of the constraints that are active at the current iterate, a cycle of Gauss-Seidel relaxation beginning the minimization sequence on the direction m defined by (4.2) produces a decrease bounded below by*

$$\Delta m_k \geq \kappa_{gen} \pi_k^{tr,a} \min\left[\frac{\pi_k^{tr,a}}{\beta_k}, \Delta_k\right],$$

with $\kappa_{gen} \in (0,1)$, where

$$\pi_k^{tr,a} = \min\{1, \chi_k\} \stackrel{def}{=} \min\{1, | \min_{\substack{x_k + d \in \mathcal{L}_{\mathcal{A}(x_k)} \\ ||d||_\infty \leq 1}} < g(x_k), d > |\}$$

is the criticality measure used in the active constraints theory.

Proof. First notice that in the proof of Theorem 4.1.1 the definition of $\mathcal{C}$ does not intervene anywhere else than in the criticality measure. We have to distinguish two cases depending on whether a constraint is active in the direction m or not. If it is the case, then χ_{k,j_m} stays unchanged and the proof is exactly the same as in Theorem 4.1.1. Otherwise, the exact unidirectional minimization on m finds the same minimum as if the problem was unconstrained so we obtain the sufficient decrease property for unconstrained trust-region methods (see Conn et al., 2000) and, as $\mathcal{C}_{[x]_{j_m}} = \mathbb{R}$ by assumption, we have $\chi_{k,j_m} = |[g_k]_{j_m}|$ and thus

$$\Delta m_k^m \geq \kappa_{mdc} \chi_{k,j_m} \min\left[\frac{\chi_{k,j_m}}{\beta_k}, \Delta_k, 1\right]$$

with $\kappa_{mdc} \in (0,1)$. The rest of the proof is also identical to the proof of Theorem 4.1.1 with $\kappa_{gen} = \frac{\min(\kappa_{mdc}, \kappa_{dcp})}{n^2 \kappa_{k,j_m}^2} \in (0,1)$ and the result follows from $\pi_k^{tr,a} = \min\{1, \chi_k\}$. □

4.1.2 Linesearch

The implementation whose numerical performance is discussed in Section 4.2 uses a version that combines the traditional trust-region techniques with a linesearch, in the spirit of Toint (1983, 1987), Nocedal and Yuan (1998) and Gertz (1999) (see Conn et al., 2000, Section 10.3.2). More precisely, if $\rho_{i,k} < \eta_1$ in Step 4 of Algorithm RMTR$_\infty$ and the step is *gradient related* in the sense that

$$|\langle g_{i,k}, s_{i,k}\rangle| \geq \epsilon_{\text{gr}} \|g_{i,k}\|_2 \, \|s_{i,k}\|_2$$

for some $\epsilon_{\text{gr}} \in (0,1)$, the step corresponding to a new iteration and a smaller trust-region radius can be computed by backtracking along $s_{i,k}$, instead of recomputing a new one using SCM smoothing. On the other hand, if some iteration at the topmost level is successful and the minimizer of the quadratic model in the direction $s_{r,k}$ lies sufficiently far beyond the trust-region boundary, then a single doubling of the step is attempted to obtain further descent, a strategy reminiscent of the *internal doubling* procedure of Dennis and Schnabel (1983) (see Conn et al., 2000, Section 10.5.2), or the *magical step* technique of Conn, Vicente and Visweswariah (1999) and Conn et al. (2000), Section 10.4.1. The theoretical arguments developed in these references guarantee that global convergence of the modified algorithm to first-order critical points is not altered.

4.1.3 Second-order and Galerkin models

In Chapter 2, we have assumed that f_i and f_{i-1} coincide at first order (up to the constant σ_i) in the range of the prolongation operator, since we can always re-define the coarse model f_{i-1} of f_i by adding a gradient correction term to the original coarse function f_{i-1} as in

$$f_{i-1}(x_{i-1,0} + s_{i-1}) \leftarrow f_{i-1}(x_{i-1,0} + s_{i-1}) + \langle R_i g_{i,q} - \nabla_x f_{i-1}(x_{i-1,0}), s_{i-1}\rangle, \quad (4.5)$$

and therefore

$$\langle g_{i,k}, P_i s_{i-1}\rangle = \frac{1}{\sigma_i}\langle R_i g_{i,k}, s_{i-1}\rangle = \frac{1}{\sigma_i}\langle g_{i-1,0}, s_{i-1}\rangle.$$

Although this feature is theoretically crucial, our experience indicates that is not enough to obtain an efficient numerical method. We can also achieve second-order coherence through the levels by modifying the original coarse function like in

$$f_{i-1}(x_{i-1,0}+s_{i-1}) \leftarrow f_{i-1}(x_{i-1,0}+s_{i-1}) + \langle v_{i-1}, s_{i-1}\rangle + \tfrac{1}{2}\langle s_{i-1}, W_{i-1}s_{i-1}\rangle, \quad (4.6)$$

where $W_{i-1} = R_i\nabla^2 f_i(x_{i,k})P_i - \nabla^2 f_{i-1}(x_{i-1,0})$, since we then have that

$$\langle P_i s_{i-1}, \nabla^2 f_i(x_{i,k})P_i s_{i-1,}\rangle = \frac{1}{\sigma_i}\langle s_{i-1}, \nabla^2 f_{i-1}(x_{i-1,0})s_{i-1}\rangle.$$

The second-order model (4.6) is of course more costly, as the matrix W_{i-1} must be computed when starting the minimization at level $i-1$ and must also be used to update the gradient of f_{i-1} at each successful iteration at level $i-1$.

Another strategy consists in choosing the original $f_{i-1}(x_{i-1,0}+s_{i-1}) = 0$ for all s_{i-1} in the right hand side of (4.6). This strategy amounts to considering the lower-level model as the "restricted" version of the quadratic model at the upper level (this is known as the *Galerkin approximation*) and is interesting in that no evaluation of f_{i-1} is required. In the unconstrained case, when this model is strictly convex and the trust region is large enough, one minimization in Algorithm RMTR$_\infty$ (without premature termination) corresponds to applying a Galerkin multigrid linear solver (cfr. Briggs et al. (2000)) on the associated Newton's equation. Note that this choice is allowed within the theory presented in Chapter 2, since the zero function is obviously twice-continuously differentiable, bounded below and has uniformly bounded Hessians.

4.1.4 Hessian of the models

Computing a model Hessian $H_{i,k}$ is often one of the heaviest tasks in Algorithm RMTR$_\infty$. Our choice in the experiments described in Section 4.2 is to use the exact second derivative matrix of the objective functions f_i. However, we have designed an automatic strategy that avoids recomputing the Hessian at each iteration when the gradient variations are still well predicted by the available $H_{i,k-1}$. More specifically, we choose to recompute the Hessian at the beginning of iteration (i,k) $(k>0)$ whenever the preceding iteration is not successful enough (i.e. $\rho_{i,k-1} < \eta_H$) or when (since it indicates that the Hessian approximation is relatively poor)

$$\|g_{i,k} - g_{i,k-1} - H_{i,k-1}s_{i,k-1}\|_2 > \epsilon_H\|g_{i,k}\|_2,$$

where $\epsilon_H \in (0,1)$ is a small user-defined constant. Otherwise, we use $H_{i,k} = H_{i,k-1}$. Default values of $\epsilon_H = 0.15$ and $\eta_H = 0.5$ appear to give satisfactory results in most cases and these are the values we use in our reported tests.
Moreover, note that the Hessian of the MINPACK test problem are not supplied by the MINPACK code and have been obtained once and for all at the beginning of the calculation by applying an optimized finite-difference scheme (see Powell and Toint, 1979).

4.1.5 Prolongations and restrictions

We have chosen to define the prolongation and restriction operators P_i and R_i as follows. The prolongation is chosen as the *linear interpolation* operator, for example

equal to

$$P_i = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

in 1-D when $n_i = 7$ and $n_{i-1} = 3$, and the restriction is its transpose normalized to ensure that $\|R_i\|_\infty = 1$ and $\sigma_i = \|P_i^T\|_\infty^{-1}$ (see (2.6)). These operators are never assembled, but are rather applied locally for improved efficiency. Cubic interpolation could also be used in principle, but the operator is denser and therefore produces denser Galerkin models and is very restrictive in the context of Gelman-Mandel restrictions, as shown in 1-D and with $n_i = 7$ and $n_{i-1} = 3$ again :

$$P_i = \frac{1}{16}\begin{pmatrix} 12 & -2 & 0 \\ 16 & 0 & 0 \\ 9 & 9 & -1 \\ 0 & 16 & 0 \\ -1 & 9 & 9 \\ 0 & 0 & 16 \\ 0 & -2 & 12 \end{pmatrix}.$$

Moreover our experience is that the algorithm is computationally less efficient.

4.1.6 Free and fixed form recursions

An interesting feature of the RMTR$_\infty$ framework is that its convergence properties are preserved if the minimization at lower levels ($i = 0, \ldots, r-1$) is stopped after the *first successful iteration*. The flexibility of this allows us to consider different recursion patterns, namely *fixed-form* and *free-form* ones. In a fixed form recursion pattern, a maximum number of successful iterations at each level is specified (like in V- and W-cycles in multigrid algorithms, see Briggs et al. (2000)). If no such premature termination is used but the minimization at each level is carried out until one of the classical termination conditions on the criticality measure and step size (see Step 5 of Algorithm RMTR$_\infty$) is satisfied, then the actual recursion pattern is uniquely determined by the progress of minimization at each level (hence yielding a free form recursion pattern).

In Section 4.2, we compare three recursion forms. In the first form, which we call the V-form, the minimization at the lower levels consists of one successful smoothing iteration, followed by a successful recursive iteration, itself followed by a second successful smoothing iteration[(2)]. The second form is called W-form and is defined as a V-form to which is added one successful recursive iteration, and a final successful smoothing iteration. The third form is the free form recursion as explained above, in which, however, we impose that smoothing iterations and recursive (successful) iterations alternate at all levels but the coarsest. Indeed, during our experiments,

[(2)] At the coarsest level, 0, smoothing iterations are skipped and recursion is impossible, such that we only perform PTCG iterations at this level

we have found this alternation very fruitful (and rather natural in the interpretation of the algorithm as an alternation of high frequency reductions and low frequency removals).

Note that for each recursion form, any remaining iteration is skipped if one of the termination conditions in Step 5 of Algorithm RMTR_∞ is satisfied.

4.1.7 Criticality measure choice

We have shown in Section 3.4 that the convergence theory of RMTR_∞ holds when using both χ^{tr} and $\chi^{out,1}$ as a criticality measure. In addition, their value only differ on bound-constrained problems since both reduce to $||g||_1$ if there is no constraint. Nevertheless, we have chosen to select χ^{tr} to perform our tests because of the extensive number of informal tests we have made using this measure that have helped us to select the most important parameters. Moreover, PTCG has not yet been proved to satisfy the sufficient decrease condition (2.21) when $\chi^{out,1}$ is selected. This does not prevent convergence of RMTR_∞ because PTCG can be forced to (and generally does) converge exactly at the coarsest level and then the practical algorithm described in this section is completely covered by our convergence theory. However, more practical algorithms are theoretically ensured to be globally convergent when using χ^{tr}, like, e.g., the one where PTCG is used for Taylor iterations at all levels. Notice in addition that requiring that $\chi^{tr}_{r,k} \leq \epsilon_r = 10^{-3}$, for example, is approximately the same as requiring the *scaled criticality measure* $\frac{\chi^{tr}_{r,k}}{n_r}$ (whose value is comparable, for instance, with $\chi^{in,\infty}_{r,k}$ when $\alpha_g = \alpha_l = \alpha_u = 1$) to be such that $\frac{\chi^{tr}_{r,k}}{n_r} \leq \frac{\epsilon_r}{n_r}$. This last tolerance is, for example, $\frac{\epsilon_r}{n_r} \approx 10^{-9}$ in the case where $n_r = 1046529$ and $\frac{\epsilon_r}{n_r} \approx 10^{-8}$ if $n_r = 65025$. This scaled measure will be used in graphic representations to make it easier to read. Finally, a numerical comparison of the two measures is done in Section 4.2.4.

4.1.8 Computing the starting point at the fine level

We also take advantage of the multilevel recursion idea to compute the starting point $x_{r,0}$ at the finest level by first restricting the user-supplied starting point to the lowest level and then applying Algorithm RMTR_∞ successively at levels 0 up to $r-1$. In our experiments based on regular meshes (see Section 4.2), the accuracy of the criticality measure that is required for termination at level $i < r$ is given by

$$\epsilon^\chi_i = \epsilon^\chi_{i+1}\sigma_{i+1}, \tag{4.7}$$

where ϵ^χ_r is the user-supplied criticality requirement for the topmost level and σ_{i+1} is due to the definition (3.29) of the criticality measure and the fact that (2.18) yields this constant as the ratio between two linearized decreases at successive levels. Once computed, the solution at level i is then prolongated to level $i+1$ using *cubic interpolation*. The criteria (4.7) comes from the fact that we want that the prolongation of our step stay critical for the upper level $i+1$ excepted for the highest frequencies of the error that are not visible at level i and only appear at level $i+1$ and finer levels.

4.1.9 Constants choice and recursive termination thresholds

We conclude the description of our practical algorithm by specifying our choice for the constants and the level-dependent criticality thresholds ϵ_i^χ. We set

$$\eta_1 = 0.01, \quad \eta_2 = 0.95, \quad \gamma_1 = 0.05 \text{ and } \gamma_2 = 1.00, \tag{4.8}$$

as this choice appears most often appropriate. The value 1 is also often satisfactory for the $\Delta_{i,0}$. We considered two possible expressions for the criticality thresholds. The first is related to the descent condition (2.19) and is given by

$$\epsilon_i^\chi = \kappa_\chi \chi_{i,k}^{tr} \sigma_{i+1}. \tag{4.9}$$

Note that since we have chosen the criticality measure χ_k^{tr}, we have replaced $\kappa_\chi \in (0, \sigma_{i+1})$ by $\kappa_\chi \sigma_{i+1}$ with $\kappa_\chi \in (0, 1)$ in the descent condition (2.19). We also considered using (4.7), but this was found to be unsuitable for recursive iterations. Indeed, it often prevented the effective use of coarse level computations because it was satisfied at $x_{0,i}$, resulting in an immediate return to the fine level. We thus considered an adaptation of this rule given by

$$\epsilon_i^\chi = \min\{\epsilon_{i+1}^\chi, \kappa_\chi \chi_{i,k}^{tr}\} \sigma_{i+1}. \tag{4.10}$$

This adaptation was further motivated by the observation that the alternation between SCM smoothing and recursive iterations is very efficient in practice and we want thus to impose that at least one lower-level iteration is done if the descent condition (2.19) allows it. Note that the convergence theory of Chapter 2 still holds when using those coarse stopping criteria since it requires only that at least one successful iteration is done at the coarser level.

4.2 Numerical tests

The algorithm described above has been coded in FORTRAN 95 by Dimitri Tomanos[(3)] and all experiments below were run on a 3.0 Ghz single-processor PC with 2 Gbytes of RAM.

4.2.1 Test problems

We have considered a suite of minimization problems in infinite-dimensional spaces, involving differential operators. These problems are detailed in Appendix B. The differential operators are discretized on a hierarchy of regular grids such that the coarse grid at level $i-1$ is defined by taking every-other point in the grid at level i: the ratio between the grid spacing of two consecutive levels in each coordinate direction is therefore 2. The grid transfer operators P_i are defined as in classical geometric multigrid settings, using interpolation operators. The restriction operators R_i are such that (2.6) holds.

All experiments discussed below consider the solution of the test problem on the finest grid, whose size may be found in Table 4.1, together with other problem characteristics. The algorithms were terminated when the criticality measure χ^{tr} at the finest level was below 10^{-3} for all the test cases.

[(3)] Department of Mathematics, University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium. Email: dimitri.tomanos@fundp.ac.be

Problem name	n_r	r	Comment
DNT	511	8	1-D, quadratic
P2D	1046529	9	2-D, quadratic
P3D	250047	5	3-D, quadratic
DEPT	1046529	9	2-D, quadratic, (Minpack 2)
DPJB	1046529	9	2-D, quadratic, with bound constraints, (Minpack 2)
DODC	65025	7	2-D, convex, (Minpack 2)
MINS-SB	1046529	9	2-D, convex, smooth boundary conds.
MINS-OB	65025	7	2-D, convex, oscillatory boundary conds.
MINS-DMSA	65025	7	2-D, convex, (Minpack 2)
IGNISC	65025	7	2-D, convex
DSSC	1046529	9	2-D, convex, (Minpack 2)
BRATU	1046529	9	2-D, convex, (Minpack 2)
MINS-BC	65025	7	2-D, convex, with bound constraints
MEMBR	393984	9	2-D, convex, free boundary, with bound constraints
NCCS	130050	7	2-D, nonconvex, smooth boundary conds.
NCCO	130050	7	2-D, nonconvex, oscillatory boundary conds.
MOREBV	1046529	9	2-D, nonconvex

Table 4.1: Test problem characteristics

Our testing strategy, which is discussed in the next paragraphs, is first designed to establish a good default value for the algorithmic parameters, and, in a second step, to compare the resulting method with other competing approaches.

4.2.2 In search of efficient default parameters

Given the relatively large number of parameters in our method, a complete discussion of all possible combinations is outside the scope of this section. We have therefore adopted the following approach. We first fixed the parameters for which a reasonable consensus already exists, namely the trust-region parameters η_1, η_2, γ_1 and γ_2, which are set as in (4.8), in accordance with Conn et al. (2000) and Gould, Orban, Sartenaer and Toint (2005). The initial trust-region radii $\Delta_{i,0}$ are set to 1, as suggested in Section 17.2 of the first of these references. A second class of parameters was then isolated, containing algorithmic options with very marginal effect on the computational results. These are the choice of activating the linesearch mechanism (we allow for backtracking if the initial step is unsuccessful and at most one extrapolation evaluation if it is successful and gradient-related with $\epsilon_{\rm gr} = 0.01$), the parameters ϵ_H and η_H of the Hessian evaluation strategy (we choose $\eta_H = 0.5$ and $\epsilon_H = 0.15$), and the degree of the interpolation in the prolongation operator (linear interpolation is used within recursive iterations, and cubic interpolation when prolongating the solution at a coarse level into a starting point at the next finer one). The remaining algorithmic parameters were either central in the definition of our method or found to alter the performance of the method significantly, and we focus the rest of our discussion on their choice.

We begin by determining the optimal combination of these parameters. For this purpose, we ran a large number (192) of possible combinations of these options on

our set of 17 test problems and report all results of the 3264 runs on a comet-shape graph representing a measure of the effort spent in function evaluations as a function of CPU-time. More precisely, we have first scaled, separately for each test problem, the number of function evaluations and CPU-time by dividing them by the best obtained for this problem by all algorithmic variants. We then plotted the averages of these scaled measures on all test problems for each algorithmic variant separately, after removing the variants for which the CPU limit of 1000 seconds was reached on at least one problem. In the first of these plots (Figures 4.2 and 4.3), we have used triangles for variants where the coarse Galerkin model is chosen at recursive iterations and stars for variants where the second-order model (4.6) is chosen instead[(4)].

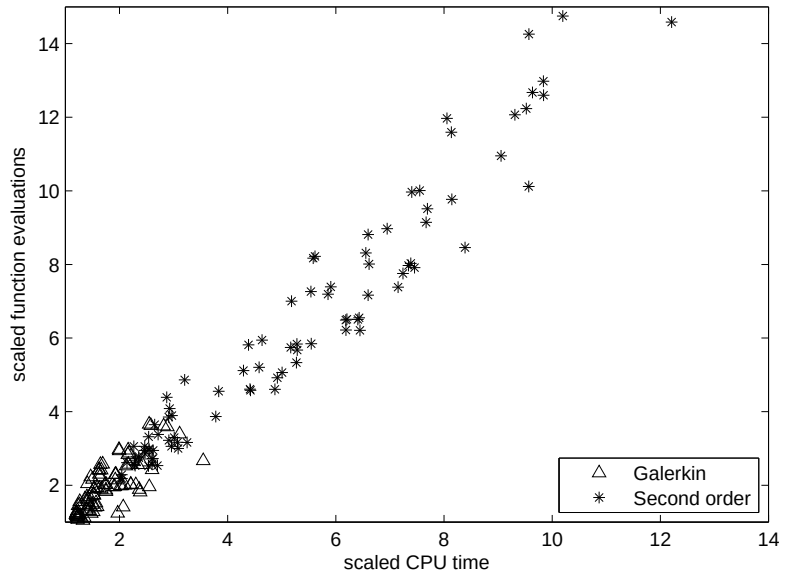


Figure 4.2: Average scaled function evaluations versus average scaled CPU-time for all algorithmic variants, distinguishing the type of model used.

We note a substantial spread of the results, with some options being up to fifteen times worse than others. The worst cases (in the top right corner) correspond to combinations of the quadratic model (4.6) with a single smoothing cycle and small values of κ_χ. On the other hand, the choice of the Galerkin model is very clearly the best. This is mainly due to the numerical cost of the alternative because it requires a function/Hessian evaluation and a matrix update for each model in (4.6). Even on the testcases for which this choice proves superior in number of iterations, the advantage is then lost in CPU-time. In view of this conclusion, we therefore select the Galerkin model as our default and restrict further analysis to this case.

We now consider the number of smoothing cycles performed at each Taylor iteration (at a level $i > 0$) and illustrate our results in Figure 4.4. All algorithmic variants

[(4)] Notice that we did not represent the tests where the coarse model is defined as in (4.5) because preliminary tests showed that performing only a first-order correction is indisputably not competitive (nearly all the tests failed to find a solution in the allowed amount of time).

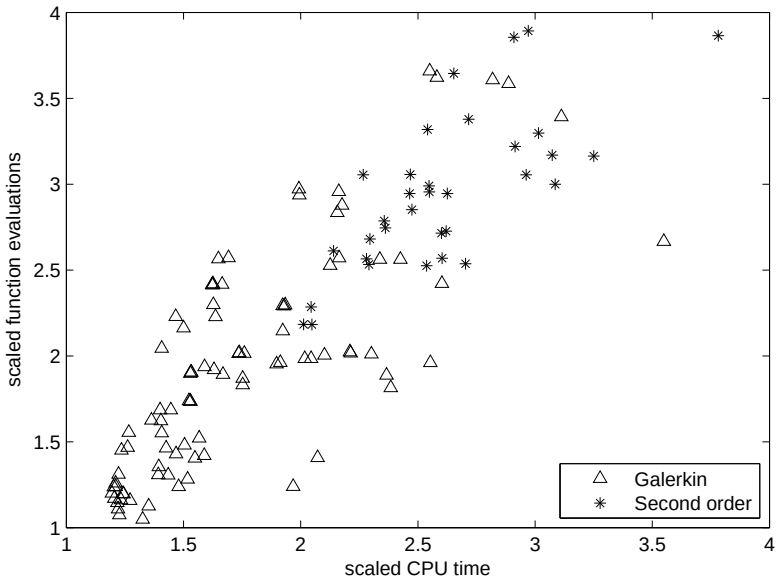


Figure 4.3: Detail of the lower left-hand corner of Figure 4.2.

(with the coarse Galerkin model) are again represented in a picture similar to Figure 4.2, where different symbols are used to isolate variants using different number of smoothing cycles.

An important property of this option is that the number of function evaluations decreases as the number of cycles increases, because a single evaluation is exploited to a fuller extent if more cycles are performed consecutively. This correlation is maintained up to a level (probably depending on the quadraticity of the objective function) beyond which the work of additional cycles is no longer effective. The correlation is much less clear when considering CPU-time, even if our result indicate that too few smoothing cycles is seldom the best option. Good choices seem to range between 2 and 7 cycles.

Choosing between the values for κ_χ is not easy. We have considered four possible values (1/2, 1/4, 1/8, 1/16). We first note that choosing κ_χ to be significantly larger than 1/2 results in a poor exploitation of the multilevel nature of the problem, since recursive iterations become much less frequent. On the other hand, values much smaller than 1/16 are also problematic because recursive iterations are then initiated for a too marginal benefit in optimality, although this strategy is closer to the unconditional recursive nature of multigrid algorithms for linear systems. In our tests the best threshold has been obtained for either $\kappa_\chi = 1/2$ or $\kappa_\chi = 1/4$, with a slight advantage for the second choice (see Figure 4.5, which is built on the same principle as the previous ones).

We now turn to the impact of the cycle types on performance, which is illustrated in Figure 4.6.

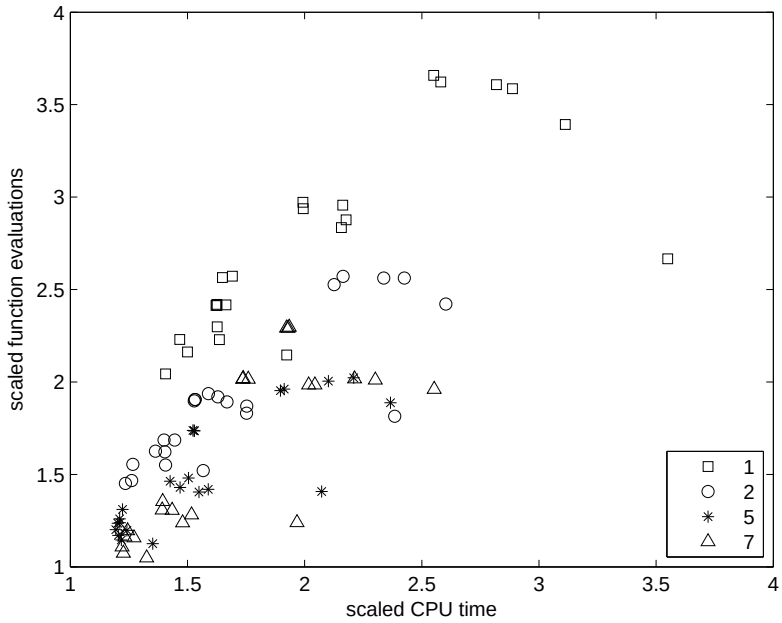


Figure 4.4: Average scaled function evaluations versus average scaled CPU-time for all algorithmic variants, distinguishing the number of smoothing cycles per Taylor iteration.

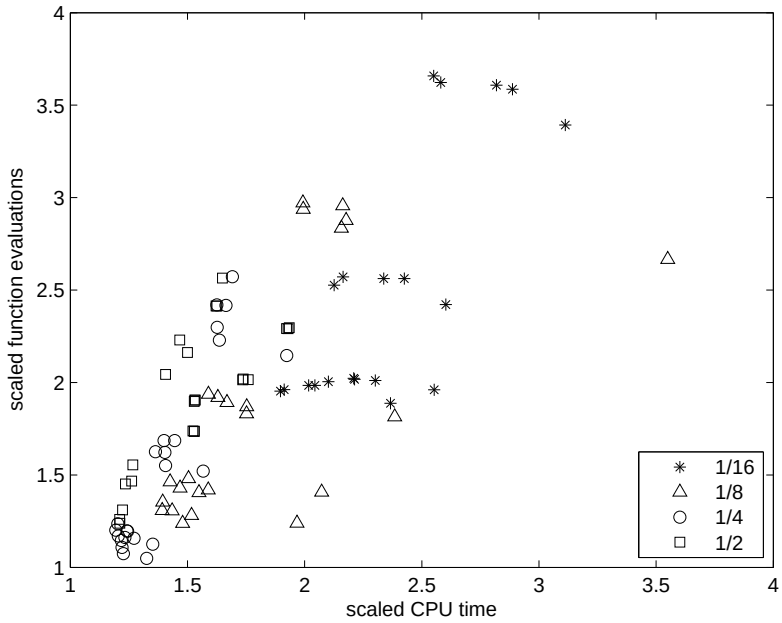


Figure 4.5: Average scaled function evaluations versus average scaled CPU-time for all algorithmic variants, distinguishing the values of κ_χ.

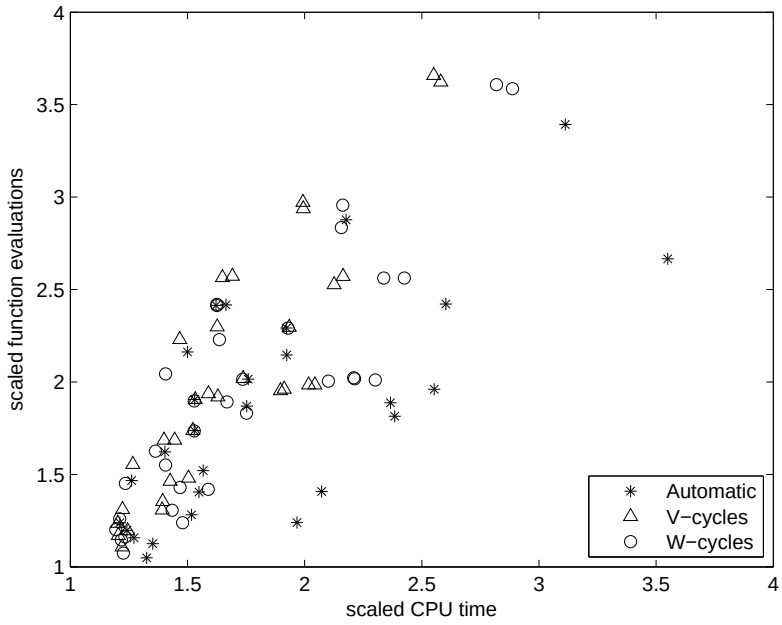


Figure 4.6: Average scaled function evaluations versus average scaled CPU-time for all algorithmic variants, distinguishing the type of recursive cycles.

Remarkably, an excellent performance can be obtained with the three considered cycle styles, quite independently of the other algorithmic parameters. In particular, this indicates that the strategy for automatically adapting the cycle type to the problem at run-time is reasonably efficient. It is however slightly more complicated and the simpler V-form may often be preferred in practice.

Finally, Figure 4.7 shows the effect of the coarse criticality threshold choice between (4.9) (nomin) and (4.10) (min). It indicates that (4.10) is generally preferable, although the performance remains mixed.

As a conclusion of this analysis, we decided to select the defaults as the use of the Galerkin model, 7 smoothing cycles per Taylor iteration, a value of $\kappa_\chi = 1/4$, V-form iterations and the (4.10) termination rule.

4.2.3 Performance of RMTR$_\infty$

We now analyze the performance of the resulting recursive trust-region algorithm in comparison with other approaches on our battery of 17 test problems. This analysis is conducted by comparing four algorithms:

- the *all on finest* (**AF**) algorithm, which is a standard Newton trust-region algorithm (with PTCG as subproblem solver) applied at the finest level, without recourse to coarse-level computations;
- the *mesh refinement technique* (**MR**), where the discretized problems are solved from the coarsest level (level 0) to the finest one (level r) successively, using the

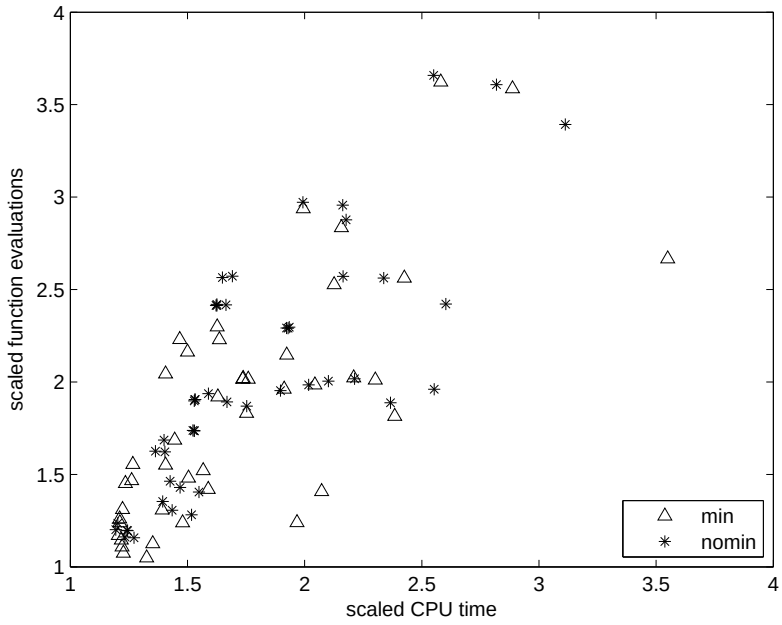


Figure 4.7: Average scaled function evaluations versus average scaled CPU-time for all algorithmic variants, distinguishing the type of lower level criticality threshold.

same standard Newton trust-region method (with PTCG as subproblem solver), and where the starting point at level $i+1$ is obtained by prolongating (using P_{i+1}) the solution obtained at level i;

- the *multilevel on finest* (**MF**) method, where Algorithm RMTR$_\infty$ is applied directly on the finest level;
- the *full multilevel* (**FM**) algorithm where Algorithm RMTR$_\infty$ is applied successively on progressively finer discretizations (from coarsest to finest) and where the starting point at level $i+1$ is obtained by prolongating (using P_{i+1}) the solution obtained at level i.

A CPU-time performance profile (see Dolan and Moré, 2002) is presented in Figure 4.8 for all our test problems and these four variants. The vertical axis of this profile represents the fraction of the total number of problems for which the tested algorithm is within a factor (represented on the horizontal axis) of the best CPU time. As a consequence, the top line on the left of the graph represents the most efficient variant of the algorithm, while the top line on the right represents the most reliable one. The first conclusion is that the full multilevel variant (FM) clearly outperforms all other variants, both in terms of efficiency and reliability. The second observation is that the AF variant is, as expected, by far the worst. The remaining two variants are surprisingly close, and the use of recursive iterations on the fine level appears to have an efficiency similar to that of optimizing on successively finer grids. These observations are confirmed by a detailed analysis of the complete numerical results presented in Appendix C.

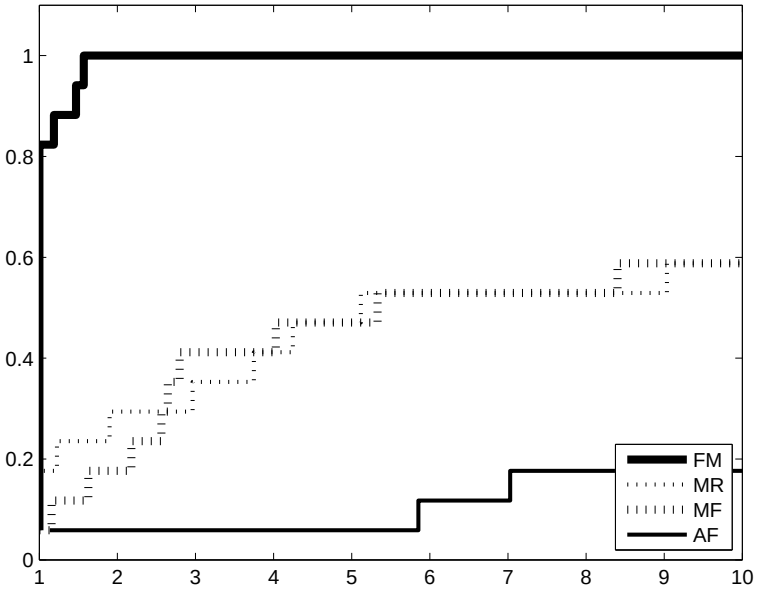


Figure 4.8: Performance profile for CPU time with variants AF, MF, MR and FM (17 test problems).

4.2.3.1 Unconstrained problems

The conclusions of the previous paragraph do not tell the whole story, as we may be interested to see if the gain in performance obtained is indeed the result of a multigrid-like gain in efficiency. To answer this question, we now turn to a more detailed comparison of the MR and FM variants on three specific unconstrained test problems (P2D, MINS-SB and NCCS), which we consider representative of the various problem classes mentioned in Table 4.1.

The performance of the algorithms is illustrated for each of these problems by a figure showing the history of the scaled criticality measure defined in Section 4.1.7 when the MR (thin line) and the FM (bold line) algorithms are used. In these figures, the dashed line represents the increase of the scaled criticality measure when a solution is prolongated during the application of a mesh refinement process. Moreover, and because iterations at coarse levels are considerably cheaper than those at higher ones, we have chosen to represent these histories as a function of the *equivalent number of finest iterations*, given by

$$q = \sum_{i=0}^{r} q_i \left(\frac{n_i}{n_r} \right), \tag{4.11}$$

where q_i is the number of iterations at level i.

We first consider the quadratic minimization problem P2D in Figure 4.9. Because this problem is equivalent to solving a linear system of equations, we expect algorithm FM to exhibit a multigrid-type behavior. Looking at Figure 4.9, we see that this is effectively the case. We note that FM is considerably more efficient than MR (by a factor approaching 100). This last result confirms that our trust-region globalization

is not hindering the known efficiency of the multigrid methods for this type of problems. Note that the significant increase of the scaled criticality measure when a lower level solution is prolongated to an upper level starting point is due to the fact that oscillatory components of the error cannot be represented on the coarser levels and therefore could not have been reduced at these levels.

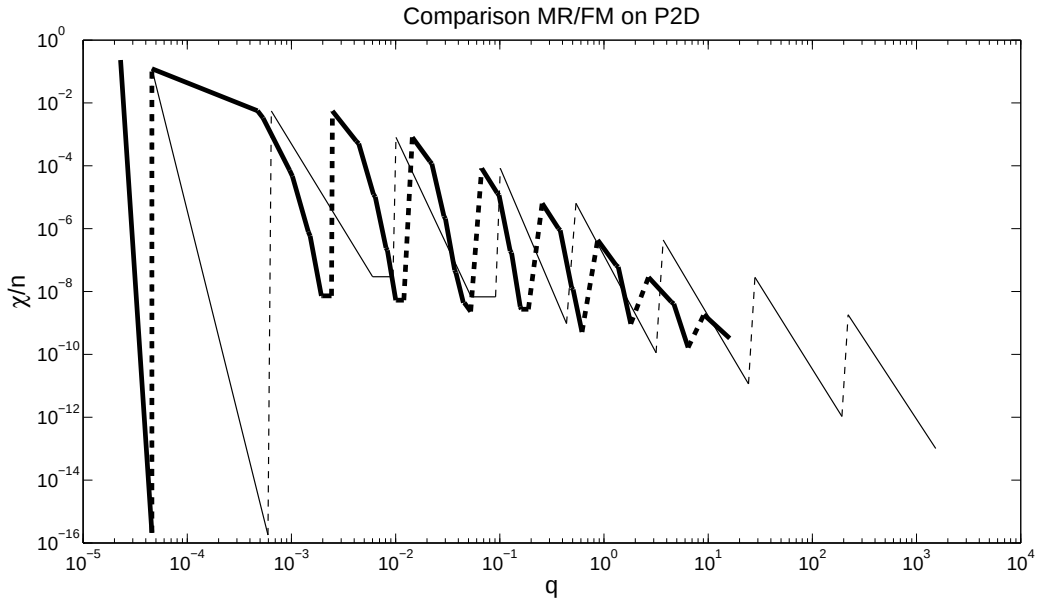


Figure 4.9: History of the scaled criticality measure on P2D. A small circle surrounds the iterations where the trust region is active. *Note that both axes are decadically logarithmic.*

The same conclusions seem to apply when we consider Figures 4.10[(5)] and 4.11, where the same algorithms are tested on MINS-SB and NCCS, respectively. This is remarkable because the problems are now more general and do not correspond anymore to linear systems of equations (MINS-SB is nonquadratic) or elliptic problems (NCCS is non-convex).

An important feature of the classical trust-region algorithm is that its convergence is speeded up when the trust-region becomes inactive (because the algorithm then reduces to Newton's method and thus achieves quadratic convergence under the assumption that the second-order Taylor model (2.16) is chosen). Iterations where the trust-region is active have been indicated, in the above figures, by a small circle (observe that they often correspond to non-monotonic decrease of the scaled criticality). We note that no such iteration occurs for MR and FM on P2D, and also that convergence speeds up for all methods as soon as the trust region becomes inactive, even if the rate is at most linear for the multilevel methods.

4.2.3.2 Bound-constrained problems

We finally evaluate the RMTR_∞ algorithm on the bound-constrained problems DPJB, MINS-BC and MEMBR. The results for these problems are presented in Fig-

[(5)] Observe that the MR variant had to be stopped after 1 hour of computing on this problem.

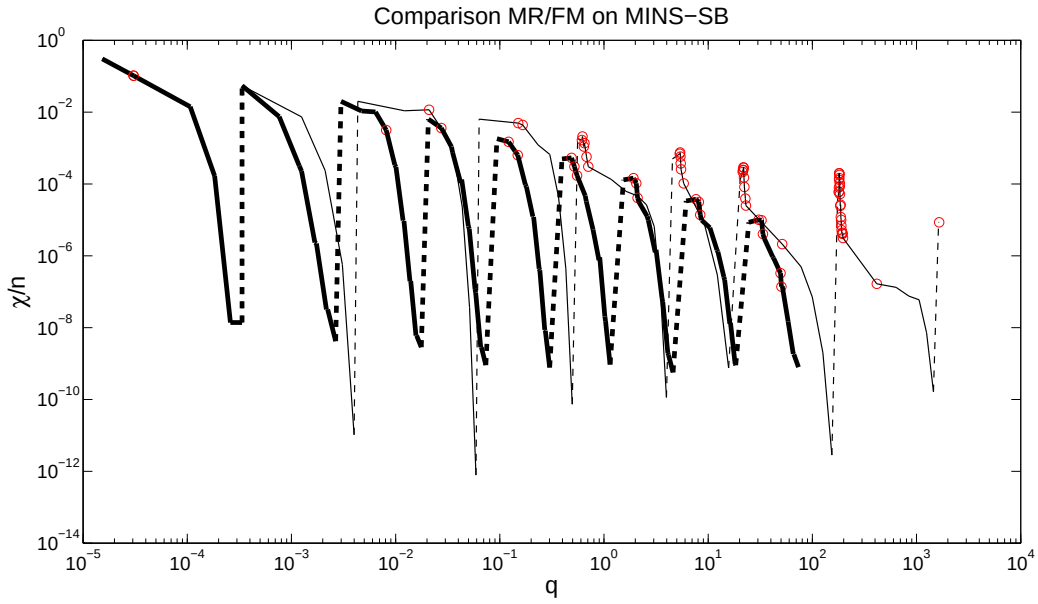


Figure 4.10: History of the scaled criticality measure on MINS-SB. A small circle surrounds the iterations where the trust region is active. *As above, both axes are decadically logarithmic.*

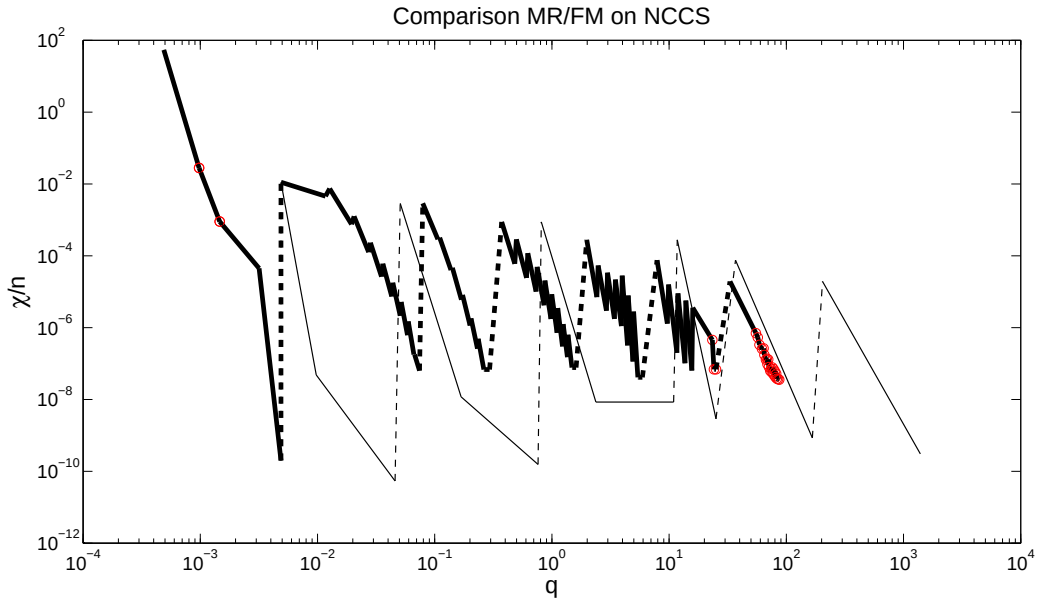


Figure 4.11: History of the scaled criticality measure on NCCS. A small circle surrounds the iterations trust region is active. *As above, both axes are decadically logarithmic.*

ures 4.12 to 4.14.

We first note that the relative performance of the considered algorithms is very

similar to that already analyzed for unconstrained problems, at least for DPJB[6] and MEMBR. On this last problem, the figure indicates that further efficiency gains could be obtained by a finer tuning of the termination accuracy at levels 5, 6 and 7. On all three problems, a gain in CPU time of a factor exceeding 10 is typically obtained when considering the multilevel variant. Again, the trust-region constraint is mostly inactive on these examples. This is in sharp contrast with MINS-BC, where it plays an important role, except in the asymptotics (as expected from trust-region convergence theory).

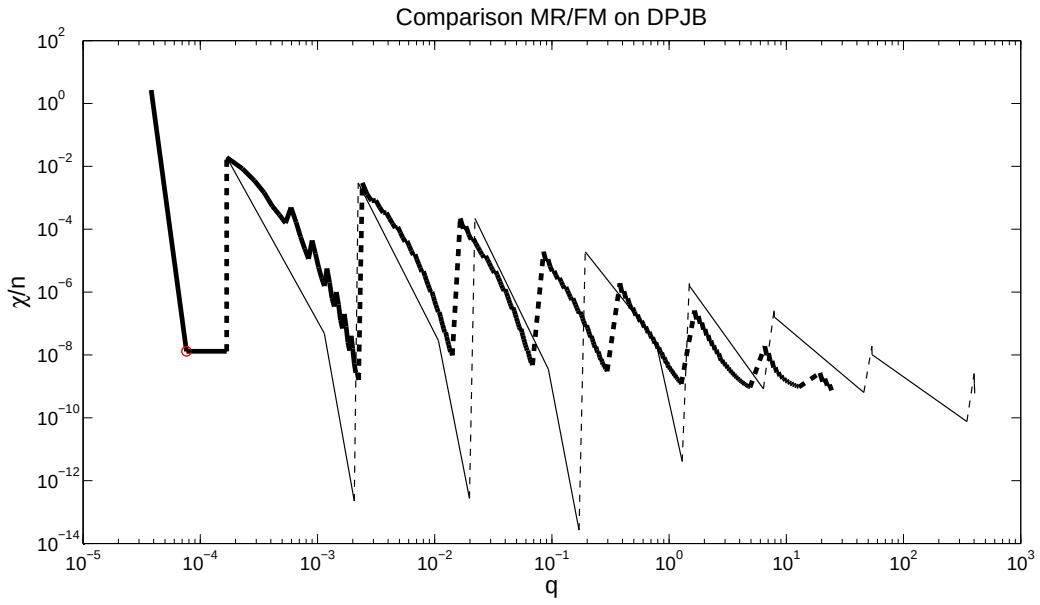


Figure 4.12: History of the scaled criticality measure on DPJB. *As above, both axes are decadically logarithmic.*

4.2.4 Numerical comparison between criticality measures

We finally would like to compare numerically some of the criticality measures defined in Chapter 3. For this purpose, we run the MATLAB version of the all on finest (AF) variant of RMTR$_\infty$, which reduces to a basic trust-region algorithm with PTCG as internal solver, on a few examples. We compare χ^{tr} and $\chi^{out,1}$ since this comparison makes sense because they both reduce to $||g||_1$ on unconstrained problems (unlike $\chi^{out,1}$ and $\chi^{out,\infty}$, for instance). We first run AF on the previously defined bound-constrained test problem MINS-BC and stop the algorithm as soon as $\chi_k^{tr} < \epsilon_r = 10^{-3}$. The history of the two (scaled) criticality measures is represented in Figure 4.15 and we see that, in this case, $\chi_k^{out,1} \geq \chi_k^{tr}$ for all k and, in particular, the backward error $\chi_k^{out,1} = 2.6 \cdot 10^{-3}$ has not yet reached the desired threshold when the algorithm is stopped. Moreover, we define the following bound-constrained problem

[6] We should note here that the Hessian of this quadratic problem is not supplied by the MINPACK code and has been obtained once and for all at the beginning of the calculation by applying an optimized finite-difference scheme (see Powell and Toint, 1979).

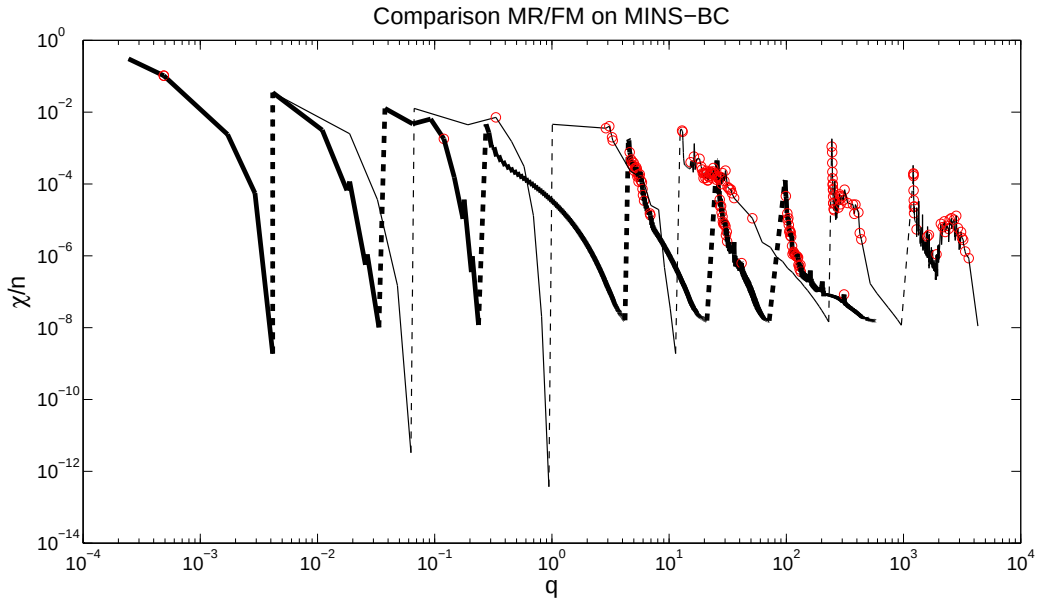


Figure 4.13: History of the scaled criticality measure on MINS-BC. A small circle surrounds the iterations where the trust region is active. *As above, both axes are decadically logarithmic.*

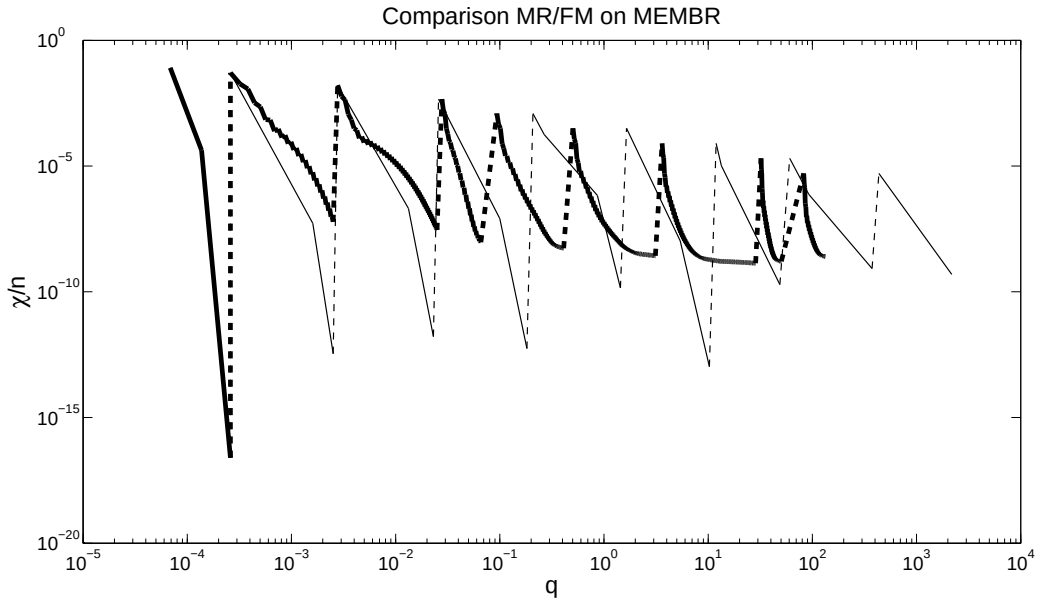


Figure 4.14: History of the scaled criticality measure on MEMBR. *As above, both axes are decadically logarithmic.*

$$\text{ACA-BC}: \quad \min_{l \le x} f(x) = \sum_{j=1}^{n} \frac{1}{10} \left([x]_j^3 + (1 + [v]_j)[x]_j \right)$$

where v is an arbitrary vector whose components are all $[v]_j \in (0, 100)$, and where $[l]_j = -10 + \sin(j)$ for all j. The solution of this problem is simply $x^* = l$ (because the

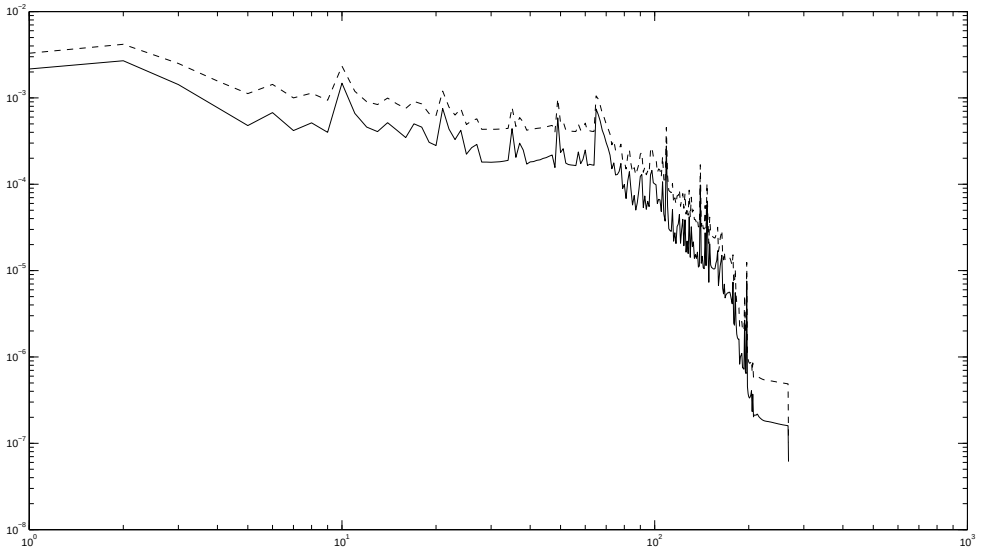


Figure 4.15: History of the two scaled criticality measures χ^{tr}/n^r (plain line) and $\chi^{out,1}/n^r$ (dashed line) on MINS-BC. *As above, both axes are decadically logarithmic.*

gradient of $f(.)$ is positive for all $x \in \mathbb{R}^n$). Figure 4.16 shows the history of the two scaled measures when AF is run on this problem, until it converges with $\chi_k^{tr} \leq 10^{-3}$. On this second example, $\chi_k^{out,1} \leq \chi_k^{tr}$ for all k and, even if the two measures are close when reaching convergence, their value can be really different during the process. Notice that the maximum into the curve representing χ^{tr} corresponds to the last iteration where no constraint is active. It shows that, on this specific example, χ^{tr} does not indicate that we are approaching the solution even if the algorithm gets progressively closer the lower bound because the huge gradient dominates the behavior of χ^{tr} and this criticality measure suddenly decreases when a bunch of constraints become active. These two small numerical examples confirm the results of the theoretical comparisons between the two criticality measures (see Lemma 3.2.7). In conclusion, χ^{tr} may behave very differently from the measure representing a backward error and, despite the possible lack of theory regarding the use of $\chi^{out,1}$ (as seen in Section 4.1.7), it can still be recommended as long as the user has some knowledge about the range of the error made when computing the bounds and the gradient of the objective function.

4.3 Conclusion

We have presented an implementation of the recursive multilevel trust-region algorithm for bound-constrained problems RMTR$_\infty$, as well as numerical experience on multilevel test problems. A suitable choice of the algorithm's parameters has been identified on these problems, yielding a good compromise between reliability and efficiency. The resulting default algorithm has then been compared to alternative optimization techniques, such as mesh refinement and direct solution of the fine-level

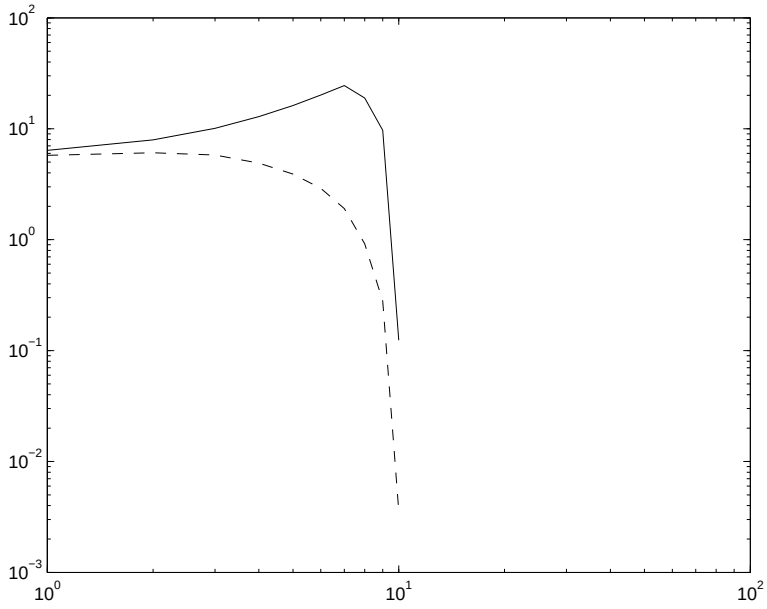


Figure 4.16: History of the two scaled criticality measures χ^{tr}/n^r (plain line) and $\chi^{out,1}/n^r$ (dashed line) on ACA-BC. *As above, both axes are decadically logarithmic.*

problem.

We are well aware that continued experimentation is needed on a larger spectrum of applications, but the numerical experience gained so far is very encouraging. Further comparison with other proposals, such as those by Kornhuber (1994,1996), is also desirable.

This chapter mainly corresponds to Gratton, Mouffe, Sartenaer, Toint and Tomanos, 2009.

Chapter 5

Conclusion

First, we have built a multilevel trust-region algorithm in infinity norm that handles the multilevel information of problems arising from a discretization, as well as the possible presence of bound constraints. This algorithm has been inspired by the method described by Gratton, Sartenaer and Toint (2008*b*), adapted to the use of the infinity-norm in the definition of the trust region and to the treatment of bound constraints. It follows the idea of multigrid methods for linear systems to eliminate progressively both low- and high-frequencies of the error by combining computations on coarser levels of discretization and the use of a smoothing technique. The main features of the new algorithm have been explained in detail, in particular the way coarser discretizations are exploited in order to construct a model of the objective function that is cheaper than Taylor's model (generally used in trust-region methods), the introduction of a descent condition indicating if it is worth using a coarse model or not, or the different ways to handle the constraints according to whether they are coming from the original bound constraints of the problem or from the trust-region method itself.

A general algorithm has been formally presented, leaving a certain number of choices open, like the solvers used to compute non-recursive steps (also called Taylor steps), the definition of the transfer operators allowing to pass information through the different levels of discretization, or the definition of the criticality measure used in the stopping criterion. This general algorithm has then been proved to be globally convergent, which means that it converges from all (feasible) starting points, under reasonable assumptions.

Moreover, we have extended the theory of Conn et al. (1993) about the identification of active constraints, in the way that all trust-region methods for convex-contrained problems that use an internal solver satisfying a sufficient decrease condition on the model (the Cauchy condition) are now proved to identify the entire active set of constraints of the solution after a finite number of iterations. We later prove that, in our bound-constrained framework, the smoothing used to compute the non-recursive steps at all levels but the coarsest satisfies the Cauchy condition. In addition, we can drive to exact convergence the projected truncated conjugate gradient (PTCG) algorithm used on the coarsest level. As a consequence, the correct active set is also identified in a finite number of steps by an adaptation of our multilevel trust-region algorithm where the active constraints are identified only by Taylor steps, while the currently active constraints are frozen during the computation of recursive

steps.

In the second Chapter, we were interested into finding a stopping criterion for our algorithm that is suitable for the solution of nonlinear bound-constrained problems. More importantly, and because our concern is on discretized problems, we have looked for a stopping criterion adapted to the case where uncertainties on the problem are known (such as discretization errors). For this reason, and comforted by the extensive theory existing in the linear case, we followed a backward error analysis approach. It traditionally consists of assuming that the current approximate solution of the original problem is the exact solution of a nearby problem and measuring the distance between these two problems in a suitable norm. The application of this technique to our case led, for some specific definition of the norm, to a well-known stopping criterion for bound-constrained nonlinear optimization : the norm of the projection of the negative gradient on the feasible set. Moreover, we have proved that a second stopping criterion, often used in trust-region methods, does not correspond to any backward error in any norm. On the other hand, this second criterion makes the algorithm stop when no significant decrease can be achieved anymore on a first-order model of the objective function, which may be an advantage in some situations. Despite their numerous differences, these two stopping criteria have been both proved to satisfy all the conditions required for the convergence of the multilevel trust-region algorithm. In addition, the general algorithm has been proved to be mesh-independent, for a specific class of problems, when the second stopping criteria (designed for trust-region methods) is chosen.

In our context (finding a first-order solution of a nonlinear bound-constrained problem), the norm used in traditional backward error analysis takes into account the distance between the two gradients and between the two sets of bounds. The possibility is left to act on weights to insist more either on the gradient or on the bounds, for example according to the uncertainties we know on those quantities. Nevertheless, we may also be interested into ensuring that both the distance between the gradients and between the bounds are sufficiently small (i.e. smaller than their respective uncertainties, for instance). This point of view led us to consider the backward error problem as a multicriteria optimization problem. We have proved that some solutions of the backward error problem viewed as a multicriteria optimization problem may unfortunately not be reached when using the traditional (norm) approach. In other words, defining a stopping criteria inspired by the multicriteria optimization point of view of backward error could lead to stop at an approximate solution unreachable by traditional stopping criteria. As a consequence, we may think of exploring the relevance of these solutions for practical situations.

The last part of the thesis has been devoted to numerical experiments. We have first specified a practical algorithm, where Taylor and recursive iterations are alternated, where a smoothing technique is used to compute the step at Taylor iterations for all levels but the coarsest while the PTCG algorithm is used for step computations at the remaining coarsest level (notice that the general algorithm authorizes the use of PTCG at all levels). Practical implementations have been tested for various algorithmic parameters, as the transfer operators, the definition of the coarse model or the constant in the descent condition, for instance. We also briefly explained algorithmic features, like the way we compute the Hessian matrices or when we allow

for linesearches. We have chosen the criticality measure designed for trust-region algorithms as a stopping criteria for these numerical tests. However, our experiments have shown that the two measures may behave very differently during the iterative process on some problems.

All the experiments were done on a representative panel of test problems. The first part of our numerical tests was focused on finding an optimal combination of the parameters of the method in order to define a reasonable set of default values. We retain that a second-order Galerkin coarse model is certainly advisable, and that the best number of smoothing cycles decreases when the nonlinearity of the problem increases. We then used the selected values to compare the multilevel trust-region algorithm to competing methods in this field. These comparisons showed a significant advantage for our method, both in terms of efficiency and reliability.

These numerical results are really encouraging and raise the interest for developing other methods of this kind. In particular, we think about the extension of the algorithm to the treatment of more general constraints, which could be first handled by introducing a penalty function in an augmented Lagrangian setting. Moreover, the multilevel algorithm we have presented here is still based on geometrical multilevel ingredients, and it could be interesting to develop an adaption of the method based on the algebraic techniques. Nevertheless, as in algebraic multigrid methods for linear systems, the computation of new transfer operators at each iteration is extremely costly and we should think about ways to tackle this problem.

In conclusion, this thesis is in line with the growing interest for multilevel methods within the nonlinear optimization community. Indeed, this community is increasingly confronted to infinite dimensional problems involving integral and/or partial differential operators, where a hierarchy of cost functions and constraints is naturally available. In this setting, the results presented here constitute another manifestation that whenever the underlying nature of the problem (here, the discretized aspect) can be taken into account, amazingly efficient algorithms can be designed. Moreover, meaningful stopping criteria can be defined for these problems when uncertainties due to their nature are known.

Appendix A

Notations and constants

Notations	Value	Meaning/Origin
$[v]_j$		The j^{th} component of a vector v.
$[M]_{ij}$		The component of the i^{th} row and the j^{th} column of a matrix M.
$[v]^T$		The transpose of a vector v.
$[M]^T$		The transpose of a matrix M.
$\|\|.\|\|_g$		Norm on the perturbation of the gradient in χ_k^{out}.
$\|\|.\|\|_{glu}$		Norm on the sum of the perturbations in χ_k^{in}.
$\|\|.\|\|_{in}$		Product norm on the perturbation in χ_k^{in}.
$\|\|.\|\|_l$		Norm on the perturbation of the lower bound constraint in χ_k^{out}.
$\|\|.\|\|_{out}$		Product norm on the perturbation in χ_k^{out}.
$\|\|.\|\|_u$		Norm on the perturbation of the upper bound constraint in χ_k^{out}.
α_g	$\in (0,1)$	Weight on the perturbation on g in the definition of χ_k^{in} and χ_k^{in}.
α_l	$\in (0,1)$	Weight on the perturbation on l in the definition of χ_k^{in} and χ_k^{in}.
α_{lu}	$\alpha_{lu} = \alpha_l = \alpha_u \in (0,1)$	Weight on the perturbation on l and u in the definition of χ_k^{in} and χ_k^{in}.
α_u	$\in (0,1)$	Weight on the perturbation on u in the definition of χ_k^{in} and χ_k^{in}.
$\beta_{i,k}$	$\beta_{i,k} = 1 + \|H_{i,k}\|_{\infty,1}$	Bound on the approximation of the Hessian matrix of the objective function at some level i and some iteration k.
Γ_k	$\Gamma_k = \text{Proj}_{\mathcal{F}}(x_k - \nabla_x f(x_k)) - x_k$	The projection of the negative gradient on the set of constraints.
$\Gamma_k(\alpha_g, \alpha_{lu})$	$\Gamma_k(\alpha_g, \alpha_{lu}) =$ $\alpha_{lu}\text{Proj}_{\mathcal{F}}(x_k - \frac{\alpha_g}{\alpha_{lu}}\nabla_x f(x_k)) - x_k$	The projection of the negative gradient on the set of constraints.

Notations	Value	Meaning/Origin
γ_1	$\in (0, \gamma_2)$	Trust-region update.
γ_2	$\in (\gamma_1, 1)$	Trust-region update.
Δg		The perturbation on the gradient.
Δl		The perturbation on the lower bound constraint.
Δu		The perturbation on the upper bound constraint.
$\Delta_{i,k}$		The trust-region radius at some level i and some iteration k.
$\Delta_{\min}$	$\Delta_{\min} = \gamma_1 \min[\kappa_2, \kappa_3 \epsilon_{\min}] \in (0, 1)$	Lower bound on all trust-region radii at all levels.
Δ_i^{s}		The initial trist-region radius at level i.
$\Delta_{\min}^{\mathrm{s}}$	$\Delta_{\min}^{\mathrm{s}} = \min_{i=0,\ldots,r} \Delta_i^{\mathrm{s}}$	The minimal initial trust-region radius over all the levels.
ϵ_g		A chosen tolerance representing an order of magnitude corresponding to the accuracy of the computation of g.
ϵ_i		The tolerance on the stopping criterion at some level i.
ϵ_l		A chosen tolerance representing an order of magnitude corresponding to the accuracy of the computation of l.
$\epsilon_{\min}$	$\epsilon_{\min} = \min[1, \kappa_\chi^r \epsilon_r] \in (0, 1]$	Lower bound on the criticality measure at all iterations but the latest of a minimization sequences.
ϵ_u		A chosen tolerance representing an order of magnitude corresponding to the accuracy of the computation of u.
η_1	$\in (0, \eta_2)$	Trust-region update.
η_2	$\in (\eta_1, 1)$	Trust-region update.
κ_2	$\kappa_2 = \frac{1}{2} \min\left[1, \frac{\epsilon_{\min}}{2\kappa_{\mathrm{g}}}, \Delta_{\min}^{\mathrm{s}}\right] \in (0, 1)$	If $\Delta_{i,k}$ is smaller than κ_2, then no recursion occurs in iteration (i, k).
κ_3	$\kappa_3 = \min\left[1, \frac{\kappa_{\mathrm{red}}(1-\eta_2)}{\kappa_{\mathrm{H}}}\right] \in (0, 1)$	If $\Delta_{i,k}$ is smaller than $\min[\kappa_2, \kappa_3 \chi_{i,k}]$, then iteration (i, k) is very successful.
κ_χ	$\in (0, \max\{1, \sigma_i\})$	Descent condition.
κ_{all}	$\in (0, \max\{1, \frac{1}{2}\})$	Sufficient decrease (modified Cauchy) condition when the criticality measure π_k^{SD} is used.
κ_{g}	$\in [1, +\infty)$	Uniformly bounded gradients.
κ_{H}	$\in [1, +\infty)$	Uniformly bounded Hessian matrices and approximations.
κ_{h}	$\kappa_{\mathrm{h}} = \kappa_{\mathrm{red}} \epsilon_{\min} \min\left[1, \frac{\epsilon_{\min}}{\kappa_{\mathrm{H}}}, \Delta_{\min}\right]$	Constant in the lower bound on the decrease of the objective function during a minimization sequence.

Notations	Value	Meaning/Origin
κ_{L}	$\in [1, +\infty)$	Lipschitz continuity of the criticality measures.
κ_P	$\in [1, +\infty)$	Uniformly bounded prolongation operators.
κ_{red}	$\in (0, \frac{1}{2})$	Sufficient decrease (modified Cauchy) condition.
μ	$\mu = \eta_1/\sigma_{\max}$	Constant in the lower bound on the decrease of the objective function during a minimization sequence.
$\pi(i,k)$	$\pi(i,k) = (i+1, q)$ for some $q \geq 1$	The predecessor of the current iteration k at level i.
π_k^{SD}		A specific criticality measure depending on the current active set.
σ_i	$\sigma_i = 1/\|P_i^T\|_\infty$	Condition on the transfer operators.
$\sigma_{\max}$	$\sigma_{\max} = \max[1, \max_{i=1,\ldots,r} \sigma_i] \in [1, \infty)$	Upper bound on the constant linking the transfer operators.
$\tau_{i,t}$		The total number of successful Taylor iterations in $\bigcup_{\ell=0}^{t} \mathcal{R}(i,\ell)$.
$\chi_{i,k}$		The criticality measure at some level i and some iteration k.
χ_k^{in}	$\min_{y \in \mathcal{Y}_k} \|\|\alpha_g \lvert\Delta g\rvert + \alpha_l \lvert\Delta l\rvert + \alpha_u \lvert\Delta u\rvert \|\|_{glu}$	A criticality measure based on backward error analysis.
χ_k^{out}	$\chi_k^{out} = \min_{y \in \mathcal{Y}_k} (\alpha_g \|\|\Delta g\|\|_g + \alpha_l \|\|\Delta l\|\|_l + \alpha_u \|\|\Delta u\|\|_u)$	A criticality measure based on backward error analysis.
χ_k^{tr}		A criticality measure not based on backward error analysis.
$\mathcal{A}(x)$		The set of active constraints at $x \in \mathcal{F}$.
$\mathcal{A}_\Delta(x)$		The perturbated set of active constraints at $x \in \mathcal{F}$.
$\mathcal{A}_i$	$\mathcal{F}_i = \{x_i \in \mathbb{R}^{n_i} \mid v_i \leq x \leq w_i\}$	The representation of the upper-levels trust-region at some level i.
$\mathcal{A}(\mathrm{L})$		The set of active constraints associated with the connected set of limit points L.
$\mathcal{A}_*^{max}$	$\mathcal{A}_*^{max} \not\subset \mathcal{A}(u_*) \ \forall u_* \in \mathrm{L'}_* \neq \mathrm{L}_*^{max}$	A maximal active set
$\mathcal{B}_{i,k}$	$\mathcal{B}_{i,k} = \{x_{i,k} + s_i \in \mathbb{R}^{n_i} \mid \|\|s_i\|\|_\infty \leq \Delta_{i,k}\}$	The trust-region at some level i and some iteration k.
c_i	$\mathbb{R}^n \to \mathbb{R}$	One of the convex-constraints in the Section on the identification of active constraints.
$\mathcal{F}$	$\mathcal{F} = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$	The original feasible bound-constrained set.
$\mathcal{F}_i$	$\mathcal{F}_i = \{x_i \in \mathbb{R}^{n_i} \mid l_i \leq x \leq u_i\}$	The set of bound constraints at some level i.

Notations	Value	Meaning/Origin
$[\mathcal{F}]_i$	$[\mathcal{F}]_i = \{x \in \mathbb{R}^n \vert c_i(x) \geq 0\}$	The feasible set for one of the convex-constraints in the Section on the identification of active constraints.
f	$\mathbb{R}^n \to \mathbb{R}$	The objective function.
$\nabla_x f$	$\mathbb{R}^n \to \mathbb{R}^n$	The gradient of the objective function.
$\nabla_x^2 f$	$\mathbb{R}^n \to \mathbb{R}^{n \times n}$	The Hessian matrix of the objective function.
f_i		The representation of the objective function f at some level i, generally used as a model for the function f_{i+1}
$g_{i,k}$	$g_{i,k} = \nabla_x f_i(x_{i,k})$	The gradient of the objective function at some level i and some iteration k.
$H_{i,k}$		A symmetric $n \times n$ approximation of the Hessian matrix of the objective function at some level i and some iteration k.
$\mathcal{L}_i$	$\mathcal{L}_i = \mathcal{F}_i \cap \mathcal{A}_i$	The set of level-dependant constraints at some level i
L_*		The set of all limit points.
L	$\mathrm{L} \subseteq \mathrm{L}_*$	A connected set of limit points.
L_{*k}		A connected set of limit points associated with some iterate x_k.
l		The lower bound of the initial set of bound constraints.
$m_{i,k}$		The Taylor-Newton model of the objective function at some level i and some iteration k.
$\mathcal{N}(x)$		The normal cone of $\mathcal{F}$ at $x \in \mathcal{F}$.
n_i		The dimension of the problem at level i.
P_i	linear/cubic interpolation	The prolongation operator.
$\mathcal{P}_k$	$\mathcal{P}_k \subseteq \mathcal{Y}_k$	A set inside which are the solutions of the minimization problem of both χ_k^{out} and χ_k^{in} when $\vert\vert.\vert\vert_g, \vert\vert.\vert\vert_l, \vert\vert.\vert\vert_u$ and $\vert\vert.\vert\vert_{glu}$ are monotone norms.
$\mathrm{Proj}_{\mathcal{C}}(x)$		The orthogonal projection of a vector x on a set $\mathcal{C}$.
q	$q = \sum_{i=0}^{r} q_i \left(\frac{n_i}{n_r}\right)$	The equivalent number of some quantity (e.g. function evaluations)
R_i	$R_i = \sigma_i P_i^T$	The restriction operator.
$\mathcal{R}(i,k)$		The set of all iterations occuring within iteration k at level i.
$\mathcal{S}_{i,k}$	$\mathcal{S}_{i,k} = \mathcal{A}_i \cap \mathcal{B}_{i,k}$	The set of trust-region "soft" constraints at some level i.

Notations	Value	Meaning/Origin
$\mathcal{S}_k^{in}$	$\mathcal{S}_k^{in} \subseteq \mathcal{Y}_k$	The set of solutions of the minimization problem inside χ_k^{in}.
$\mathcal{S}_k^{out}$	$\mathcal{S}_k^{out} \subseteq \mathcal{Y}_k$	The set of solutions of the minimization problem inside χ_k^{out}.
$s_{i,k}$		The step computed at some level i and some iteration k.
s^{SD}		A step satisfying the (modified Cauchy) sufficient decrease condition.
$\mathcal{T}(i,k)$		The set of all Taylor iterations occuring within iteration k at level i.
$\mathcal{T}(x)$		The tangent cone of $\mathcal{F}$ at $x \in \mathcal{F}$.
u		The upper bound of the initial set of bound constraints.
$\mathcal{V}$	$\mathcal{V}(\varepsilon,\delta) = \{x \in \mathbb{R}^n \vert \text{dist}(x,\varepsilon) \leq \delta\}$	Le voisinage de rayon δ autour d'un ensemble ε.
$\mathcal{W}_{i,k}$	$\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{A}_i \cap \mathcal{B}_{i,k}$	The feasible set at some level i.
$x_{i,k}$		The current iterate at some level i and some iteration k.
x_k^{SD}		An iterate obtained by a step s_k^{SD} ensuring the sufficient decrease condition.
x^*		The exact solution of the problem.
$\mathcal{Y}_k$	$\mathcal{Y}_k = \{y \in \mathbb{R}^{3n} : [\nabla_x f(x_k) + \Delta g]_j = 0 \ \forall j \notin \mathcal{A}_\Delta(x_k)\}$	The set of perturbations such that x_k is the exact solution of the perturbated problem.
y	$y = (\Delta g\,;\, \Delta l\,;\, \Delta u)$	A vector gathering the three perturbations on the gradient and on the bounds.

Appendix B

Theoretical complements

B.1 Gauss-Seidel smoothing and sufficient decrease

The sufficient decrease is an essential property of trust-region methods. It is required to demonstrate their first-order convergence as well as in the active constraints identification theory. This section is designed to show that Gauss-Seidel iterations satisfy the sufficient decrease property (2.21) in the sense

$$\Delta m_k \geq \kappa_{gen} \chi_k \min \left[\frac{\chi_k}{\beta_k}, \Delta_k, 1 \right], \tag{B.1}$$

with $\kappa_{gen} \in (0, \frac{1}{2})$ in the case of bound-constrained optimization. This result has already been proved using $\chi_k = \chi_k^{tr}$ in Chapter 4, but it remains to prove it for $\chi_k = \chi_k^{out,1}$.

We now want to prove that (B.1) holds for a cycle of SCM (Gauss-Seidel) smoothing described by Algorithm 4.1.1, when $\chi_k = \chi_k^{out,1}$, in the case where the feasible set $\mathcal{W}_{i,k}$ is bound-constrained, under the condition that the Gauss-Seidel minimization begins in the direction j_m, where

$$j_m = \arg\max_j \left| [\Gamma_{i,k}(\alpha_g, \alpha_{lu})]_j \right|, \tag{B.2}$$

and

$$\Gamma_k(\alpha_g, \alpha_{lu}) = \alpha_{lu} \left(\text{Proj}_{\mathcal{W}_{i,k}}(x_{i,k} - \frac{\alpha_g}{\alpha_{lu}} g_{i,k}) - x_{i,k} \right),$$

leading to $\chi_{i,k}^{out,1} = ||\Gamma_{i,k}(\alpha_g, \alpha_{lu})||_1$. We also define $[\mathcal{W}_{i,k}]_j = \{x \in \mathbb{R}^n : l(\mathcal{W}_{i,k}) \leq [x]_j \leq u(\mathcal{W}_{i,k})\}$ and simplify the following notation $\Gamma_{i,k} \overset{not.}{=} \Gamma_{i,k}(1,1)$. We finally supress the first subscript i in all the quantities in order to make the proof easier to read.

Theorem B.1.1 *A cycle of Gauss-Seidel relaxation applied when minimizing on the model (2.16) on a bound-constrained set, and beginning the minimization sequence on some direction j_m satisfying (B.2) produces a decrease bounded below by*

$$\Delta m_k \geq \kappa_{gen} \chi_k^{out,1} \min \left[\frac{\chi_k^{out,1}}{\beta_k}, \Delta_k, 1 \right],$$

with $\kappa_{gen} \in (0, \frac{1}{2})$.

Proof. For each iteration of index k define $\mathcal{W}_k^s \stackrel{def}{=} \{s \in \mathbb{R}^n : l(\mathcal{W}_k) \leq s + x_k \leq u(\mathcal{W}_k)\}$, $[\mathcal{W}_k^s]_{j_m} = \{[s]_{j_m} \in \mathbb{R} : [l(\mathcal{W}_k)]_{j_m} \leq [s]_{j_m} + [x_k]_{j_m} \leq [u(\mathcal{W}_k)]_{j_m}\}$, $[\Gamma_k^s]_{j_m} \stackrel{def}{=} \text{Proj}_{[\mathcal{W}_k^s]_{j_m}}(-[g_k]_{j_m})$ and $[\Gamma_{H,k}^s]_{j_m} \stackrel{def}{=} \text{Proj}_{[\mathcal{W}_k^s]_{j_m}}\left(\frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}}\right)$. We then begin by a simple result, that is

$$|[g_k]_{j_m}| \geq |[\Gamma_k^s]_{j_m}| = |[\Gamma_k]_{j_m}| \geq \left|[\Gamma_k(\alpha_g, \alpha_{lu})]_{j_m}\right| \tag{B.3}$$

because the projection of a scalar on a subspace of $\mathbb{R}$ containing the origin (as $[\mathcal{W}_k^s]_{j_m}$ does) is always smaller in absolute value than the absolute value of the original quantity, the definition of $[\Gamma_k^s]_{j_m}$ and $[\Gamma_k]_{j_m}$, and also because $0 < \alpha_g, \alpha_{lu} \leq 1$. Now consider the first case where $[H_k]_{j_m,j_m} > 0$. In this case, if $[s_k^+]_{j_m} \in [\mathcal{W}_k^s]_{j_m}$, then the unidirectional minimization guarantees (see Conn et al., 2000)

$$\Delta m_k^m \geq \kappa_{mdc}|[g_k]_{j_m}| \min\left[\frac{|[g_k]_{j_m}|}{\beta_k}, \Delta_k\right],$$

with $\kappa_{mdc} \in (0, \frac{1}{2})$. As a consequence, (B.3), the definition of j_m, the norm equivalence and the definition of $\chi_k^{out,1}$ imply

$$\begin{aligned}
\Delta m_k^m &\geq \kappa_{mdc}|[\Gamma_k(\alpha_g, \alpha_{lu})]_{j_m}| \min\left[\frac{|[\Gamma_k(\alpha_g, \alpha_{lu})]_{j_m}|}{\beta_k}, \Delta_k\right] \\
&\geq \frac{\kappa_{mdc}}{n^2}\chi_k^{out,1} \min\left[\frac{\chi_k^{out,1}}{\beta_k}, \Delta_k, 1\right].
\end{aligned}$$

Then if $[s_k^+]_{j_m} \notin [\mathcal{W}_k^s]_{j_m}$, in view of Algorithm 4.1.1 and the definition of $[\mathcal{W}_k^s]_{j_m}$, we see that it means $\exists j$ such that either

$$\frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}} + [x_k]_{j_m} < [l(\mathcal{W}_k)]_{j_m} \text{ or } \frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}} + [x_k]_{j_m} > [u(\mathcal{W}_k)]_{j_m}.$$

In the first case, this means $s_k^+ = \frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}} < [l(\mathcal{W}_k)]_{j_m} - [x_k]_{j_m} = [\Gamma_{H,k}^s]_{j_m} \leq 0$, where the last inequality comes from the fact that $x_k \in \mathcal{W}_k$ implies $[x_k]_{j_m} \in [\mathcal{W}_k]_{j_m}$. We thus have

$$-[H_k]_{j_m,j_m}([\Gamma_{H,k}^s]_{j_m})^2 > [g_k]_{j_m}[\Gamma_{H,k}^s]_{j_m}, \tag{B.4}$$

because $[H_k]_{j_m,j_m} > 0$ and $\Gamma_{H,k,j_m}^s \leq 0$. Notice that $\left|[\Gamma_{H,k}^s]_{j_m}\right| \geq \frac{|\Gamma_k^s|}{\beta_k}$ because $\beta_k \geq [H_k]_{j_m,j_m}$ and $\beta_k \geq 1$. Therefore, the choice of the model, the definition of s_k in Algorithm 4.1.1, equation (B.4), $[g_k]_{j_m} \geq 0$, $[\Gamma_{H,k}^s]_{j_m} \leq 0$ and (B.3) imply that the decrease is bounded by

$$\begin{aligned}
\Delta m_k^m &= -[g_k]_{j_m}[s_k]_{j_m} - \frac{1}{2}[H_k]_{j_m,j_m}[s_k]_{j_m}^2 \\
&= -[g_k]_{j_m}[\Gamma_{H,k}^s]_{j_m} - \frac{1}{2}[H_k]_{j_m,j_m}([\Gamma_{H,k}^s]_{j_m})^2 \\
&\geq -\frac{1}{2}[g_k]_{j_m}[\Gamma_{H,k}^s]_{j_m} \\
&\geq \frac{1}{2}|[g_k]_{j_m}||[\Gamma_{H,k}^s]_{j_m}| \\
&\geq \frac{1}{2}\chi_k\frac{\chi_k}{\beta_k} \\
&\geq \frac{1}{2}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k, 1\right].
\end{aligned} \tag{B.5}$$

If on the other hand we have

$$\frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}} + [x_k]_{j_m} > [u(\mathcal{W}_k)]_{j_m}$$

then we obtain $s_k^+ = \frac{-[g_k]_{j_m}}{[H_k]_{j_m,j_m}} > [\Gamma^s_{H,k}]_{j_m} = [u(\mathcal{W}_k)]_{j_m} - [x_k]_{j_m} \geq 0$, and therefore $-[H_k]_{j_m,j_m}([\Gamma^s_{H,k}]_{j_m})^2 \geq [g_k]_{j_m}[\Gamma^s_{H,k}]_{j_m}$. With a similar reasoning as in the previous case, we finally also get (B.5). Now consider the case where $[H_k]_{j_m,j_m} \leq 0$. The decrease of the model is

$$\Delta m_k^m = -[g_k]_{j_m}[s_k]_{j_m} - \frac{1}{2}[H_k]_{j_m,j_m}[s_k]^2_{j_m} \geq -[g_k]_{j_m}[s_k]_{j_m}.$$

If $[g_k]_{j_m} > 0$, then $[s_k]_{j_m} = [l(\mathcal{W}_k)]_{j,m} - [x_k]_{j_m} \leq [\Gamma^s_k]_{j_m} \leq 0$ and thus, using (B.3), we get

$$\Delta m_k^m \geq |[g_k]_{j_m}|\,|[\Gamma^s_k]_{j_m}| \geq \chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k, 1\right]. \tag{B.6}$$

Similarly, if $[g_k]_{j_m} < 0$, then $[s_k]_{j_m} = [u(\mathcal{W}_k)]_{j_m} - [x_k]_{j_m} \geq [\Gamma^s_k]_{j_m} \geq 0$, which implies (B.6). In summary, the model decrease on the j_m^{th} component always satisfies

$$\Delta m_k^m \geq \kappa_{gen}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k, 1\right],$$

where $\kappa_{gen} = \frac{\kappa_{mdc}}{n^2} \in (0, \frac{1}{2})$. But the total decrease of a Gauss-Seidel iteration is $\Delta m_k = \sum_{j=0}^n \Delta m_k^j$. Then, because the unidirectional minimization guarantees $\Delta m_k^j \geq 0, \forall j = 1, ..., n$ (note that the projection on the feasible set does not increse the objective function since we are minimizaing a quadratic function in a one dimensional space), we have that

$$\Delta m_k = \Delta m_k^m + \sum_{j=0, j\neq j_m}^{n} \Delta m_k^j \geq \kappa_{gen}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k, 1\right].$$

□

Notice that this proof works also if we use another criticality measure where the entire set of bounds is replaced by the set of active constraints only. The result becomes the following.

Corollary B.1.2 *When the feasible set $\mathcal{A}(x_k)$ is the set of the constraints that are active at the current iterate, a cycle of Gauss-Seidel relaxation beginning the minimization sequence on the direction j_m defined by (B.2) produces a decrease bounded below by*

$$\Delta m_k \geq \kappa_{gen}\pi_k^{out,1,a} \min\left[\frac{\pi_k^{out,1,a}}{\beta_k}, \Delta_k\right],$$

with $\kappa_{gen} \in (0, 1)$, where

$$\pi_k^{out,1,a} = \min\{1, \chi_k\} \stackrel{def}{=} \min\{1, ||(\text{Proj}_{\mathcal{A}(x_k)}x_k - g(x_k)) - x_k||_1\}$$

is the criticality measure used in the active constraints theory.

Proof. It is the immediate consequence of the definition of $\pi_k^{out,1,a}$ and the fact that the definition of the feasible set does not intervene in the proof as long as it is a bound-constrained set, such that it can be replaced by $\mathcal{A}(x_k)$ without modifying the conclusion : it suffices to replace $l(\mathcal{W}_k)$ and $u(\mathcal{W}_k)$ by $l(\mathcal{A}(x_k))$ and $u(\mathcal{A}(x_k))$. □

This last Corollary again implies that Gauss-Seidel steps satisfy (2.80) (with $\pi_k^{SD} = \pi_k^{tr,a}$) and thus that the active constraints identification theory holds when using a Gauss-Seidel step for bound-constrained problems.

B.2 Product norms

Theorem B.2.1 *The two quantities*

$$||(\Delta g\,;\,\Delta l\,;\,\Delta u)||_{out} = \alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u$$

and

$$||(\Delta g\,;\,\Delta l\,;\,\Delta u)||_{in} = ||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\,||_{glu}$$

are norms on $\mathbb{R}^{3n}$.

Proof.

1. $||.||_{out}$ is a norm on $\mathbb{R}^{3n}$:

   (a) $\forall y \in \mathbb{R}^{3n}\ \ ||y||_{out} = 0 \ \Rightarrow y = 0$:
   The definition of $||.||_{out}$ together with the fact that $||.||_g, ||.||_l$ and $||.||_u$ are norms and $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ imply

   $$\begin{array}{ll} & \alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u = 0 \\ \Rightarrow & \alpha_g||\Delta g||_g = 0 \text{ and } \alpha_l||\Delta l||_l = 0 \text{ and } \alpha_u||\Delta u||_u = 0 \\ \Rightarrow & ||\Delta g||_g = 0 \text{ and } ||\Delta l||_l = 0 \text{ and } ||\Delta u||_u = 0 \\ \Rightarrow & \Delta g = 0 \text{ and } \Delta l = 0 \text{ and } \Delta u = 0 \end{array} \quad ||.||_u$$

   (b) $\forall (\lambda, y) \in \mathbb{R} \times \mathbb{R}^{3n}\ \ ||\lambda y||_{out} = |\lambda|\,||y||_{out}$:
   The definition of $||.||_{out}$ together with the fact that $||.||_g, ||.||_l$ and $||.||_u$ are norms give

   $$\begin{array}{rcl} ||\lambda(\Delta g, \Delta l, \Delta u)||_{out} & = & \alpha_g||\lambda\Delta g||_g + \alpha_l||\lambda\Delta l||_l + \alpha_u||\lambda\Delta u||_u \\ & = & |\lambda|\alpha_g||\Delta g||_g + |\lambda|\alpha_l||\Delta l||_l + |\lambda|\alpha_u||\Delta u||_u \\ & = & |\lambda|(\alpha_g||\Delta g||_g + \alpha_l||\Delta l||_l + \alpha_u||\Delta u||_u) \\ & = & |\lambda|\,||(\Delta g, \Delta l, \Delta u)||_{out} \end{array}$$

   (c) $\forall (y_1, y_2) \in \mathbb{R}^{3n} \times \mathbb{R}^{3n}\ \ ||y_1 + y_2||_{out} \leq ||y_1||_{out} + ||y_2||_{out}$:
   The definition of $||.||_{out}$ together with the fact that $||.||_g, ||.||_l$ and $||.||_u$ are

norms and $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ imply

$$\begin{aligned} ||(\Delta g^1, \Delta l^1, \Delta u^1) &+ (\Delta g^2, \Delta l^2, \Delta u^2)||_{out} \\ &= \alpha_g||\Delta g^1 + \Delta g^2||_g + \alpha_l||\Delta l^1 + \Delta l^2||_l \\ &\quad +\alpha_u||\Delta u^1 + \Delta u^2||_u \\ &\leq \alpha_g(||\Delta g^1||_g + ||\Delta g^2||_g) + \alpha_l(||\Delta l^1||_l + ||\Delta l^2||_l) \\ &\quad +\alpha_u(||\Delta u^1||_u + ||\Delta u^2||_u) \\ &= \alpha_g||\Delta g^1||_g + \alpha_l||\Delta l^1||_l + \alpha_u||\Delta u^1||_u \\ &\quad +\alpha_g||\Delta g^2||_g + \alpha_l||\Delta l^2||_l + \alpha_u||\Delta u^2||_u \\ &= ||(\Delta g^1, \Delta l^1, \Delta u^1)||_{out} + ||(\Delta g^2, \Delta l^2, \Delta u^2)||_{out} \end{aligned}$$

2. $||.||_{in}$ is a norm on $\mathbb{R}^{3n}$:

   (a) $\forall y \in \mathbb{R}^{3n} \;\; ||y||_{in} = 0 \;\Rightarrow y = 0$:
   The definition of $||.||_{in}$, the fact that $||.||_{glu}$ is a norm, the presence of the absolute values inside the glu-norm and $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ imply

$$\begin{aligned} & ||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|\;||_{glu} = 0 \\ \Rightarrow\;& \alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u| = 0 \\ \Rightarrow\;& |\Delta g| + |\Delta l| + |\Delta u| = 0 \\ \Rightarrow\;& |\Delta g| = 0 \text{ and } |\Delta l| = 0 \text{ and } |\Delta u| = 0 \\ \Rightarrow\;& \Delta g = 0 \text{ and } \Delta l = 0 \text{ and } \Delta u = 0 \end{aligned}$$

   (b) $\forall(\lambda, y) \in \mathbb{R} \times \mathbb{R}^{3n} \;\; ||\lambda y||_{in} = |\lambda|\,||y||_{in}$:
   The definition of $||.||_{in}$, the presence of the absolute values inside the glu-norm together with the fact that $||.||_{glu}$ is a norm give

$$\begin{aligned} ||\lambda(\Delta g, \Delta l, \Delta u)||_{in} &= ||\alpha_g|\lambda\Delta g| + \alpha_l|\lambda\Delta l| + \alpha_u|\lambda\Delta u|\;||_{glu} \\ &= ||\alpha_g|\lambda|\,|\Delta g| + \alpha_l|\lambda|\,|\Delta l| + \alpha_u|\lambda|\,|\Delta u|\;||_{glu} \\ &= ||\;|\lambda|(\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|)||_{glu} \\ &= |\lambda|\;||\alpha_g|\Delta g| + \alpha_l|\Delta l| + \alpha_u|\Delta u|||_{glu} \\ &= |\lambda|\;||(\Delta g, \Delta l, \Delta u)||_{in} \end{aligned}$$

   (c) $\forall(y_1, y_2) \in \mathbb{R}^{3n} \times \mathbb{R}^{3n} \;\; ||y_1 + y_2||_{in} \leq ||y_1||_{in} + ||y_2||_{in}$:
   The definition of $||.||_{in}$, the fact that $||.||_{glu}$ is a norm and the triangular inequality imply

$$\begin{aligned} ||(\Delta g^1, \Delta l^1, \Delta u^1) &+ (\Delta g^2, \Delta l^2, \Delta u^2)||_{in} \\ &= ||\alpha_g|\Delta g^1 + \Delta g^2| + \alpha_l|\Delta l^1 + \Delta l^2| \\ &\quad +\alpha_u|\Delta u^1 + \Delta u^2|\;||_{glu} \\ &\leq ||\alpha_g(|\Delta g^1| + |\Delta g^2|) + \alpha_l(|\Delta l^1| + |\Delta l^2|) \\ &\quad +\alpha_u(|\Delta u^1| + |\Delta u^2|)||_{glu} \\ &= ||\alpha_g|\Delta g^1| + \alpha_l|\Delta l^1| + \alpha_u|\Delta u^1| \\ &\quad +\alpha_g|\Delta g^2| + \alpha_l|\Delta l^2| + \alpha_u|\Delta u^2|\;||_{glu} \\ &= ||\alpha_g|\Delta g^1| + \alpha_l|\Delta l^1| + \alpha_u|\Delta u^1|\;||_{glu} \\ &\quad +||\alpha_g|\Delta g^2| + \alpha_l|\Delta l^2| + \alpha_u|\Delta u^2|\;||_{glu} \\ &= ||(\Delta g^1, \Delta l^1, \Delta u^1)||_{in} + ||(\Delta g^2, \Delta l^2, \Delta u^2)||_{in} \end{aligned}$$

□

Appendix C

Test problems

We have built a suite of test problems as extensive as we could, from a variety of sources. We have kept the problems aleady discussed in Gratton et al. (2006*a*) and have also used Lewis and Nash (2005) and the Minpack-2 collection (Averick and Moré, 1991). In what follows, we denote by S_2 and S_3 respectively the unit square and cube

$$S_2 = [0,1] \times [0,1] = \{(x,y), 0 \leq x \leq 1, 0 \leq y \leq 1\}$$

and

$$S_3 = [0,1] \times [0,1] \times [0,1] = \{(x,y,z), 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}.$$

We also denote by $\mathcal{H}^1(\mathcal{D})$ the Hilbert space of all functions with compact support in the domain $\mathcal{D}$ such that v and $||\nabla v||^2$ belong to $\mathcal{L}^2(\mathcal{D})$, and by $\mathcal{H}^1_0(\mathcal{D})$ its subspace consisting of all function vanishing on the domain's boundary. For all problems, the starting value of the unknown function is chosen to be equal to one (at the finest level).

C.1 DNT: a Dirichlet-to-Neumann transfer problem

Let S be the square $[0,\pi] \times [0,\pi]$ and let Γ be its lower edge defined by $\{(x,y), 0 \leq x \leq \pi, y = 0\}$. The Dirichlet-to-Neumann transfer problem (Lewis and Nash, 2005) consists of finding the function $a(x)$ defined on $[0,\pi]$, that minimizes

$$\int_0^\pi \left(\frac{\partial u}{\partial y}(x,0) - f(x) \right)^2,$$

where $u(x,y)$ is the solution of the boundary value problem

$$\begin{array}{lcll} \Delta u & = & 0 & \text{in } S, \\ u(x,y) & = & a(x) & \text{on } \Gamma, \\ u(x,y) & = & 0 & \text{on } \partial S \backslash \Gamma, \end{array}$$

and Δ is the Laplacian operator. The problem is a 1D minimization problem, but the computations of the objective function, gradient and Hessian involve a partial differential equation in 2D. To introduce oscillatory components in the solution, we define $f(x) = \sum_{i=1}^{15} \sin(i\,x) + \sin(40\,x)$. The discretization of the problem is performed by finite differences with the same grid spacing in the two directions. The discretized problem is a linear least-squares problem.

C.2 P2D and P3D: two quadratic examples

We consider here the two-dimensional Poisson model problem P2D for multigrid solvers defined in S_2

$$\begin{aligned} -\Delta u(x) &= f(x) \text{ in } S_2 \\ u(x) &= 0 \text{ on } \partial S_2, \end{aligned}$$

where $f(x)$ is such that the analytical solution to this problem is $u(x) = 2x_2(1-x_2) + 2x_1(1-x_1)$. This problem is discretized using a 5-point finite-difference scheme. We consider the variational formulation of this problem, given by

$$\min_{x \in \mathbb{R}^{n_r}} \tfrac{1}{2} x^T A x - x^T b, \tag{C.1}$$

which is obviously equivalent to the linear system $Ax = b$, where A and b are the discretizations of the Laplacian and the right-hand side f, respectively. The main purpose of this example is to illustrate that our multilevel algorithm exhibits performances similar to traditional linear multigrid solvers on a quadratic model problem.

Problem P3D is a more nonlinear 3D version of P2D. We consider the differential equation

$$\begin{aligned} -(1+\sin^2(3\pi x_1))\Delta u(x) &= f(x) \quad \text{in } S_3, \\ u(x) &= 0 \qquad \text{on } \partial S_3. \end{aligned}$$

The right-hand side $f(x)$ is chosen such that $u(x) = x_1(1-x_1)x_2(1-x_2)x_3(1-x_3)$ is the desired solution. The Laplacian is discretized using the standard 7-point finite-difference approximation on a uniform 3D mesh. As for P2D, the solution algorithms are applied to the variational formulation (C.1).

C.3 MINS-SB, MINS-OB, MINS-BC and MINS-DMSA: four minimum surface problems

The domain of calculus of variation consists of finding stationary values v of integrals of the form $\int_a^b f(v, \dot{v}, x)\, dx$, where $\dot{v}$ is the first-order derivative of v. The multilevel trust-region algorithm can be applied to discretized versions of problems of this type. As representative of these, we consider several variants of the minimum surface problem

$$\min_{v \in \mathcal{K}} \int_{S_2} \sqrt{1 + \|\nabla_x v\|_2^2},$$

where $\mathcal{K} = \{v \in H^1(S_2) \mid v(x) = v_0(x) \text{ on } \partial S_2\}$. This convex problem is discretized using a finite-element basis defined using a uniform triangulation of S_2, with the same grid spacing, h, along the two coordinate directions. The basis functions are the classical P1 functions which are linear on each triangle and take the value 0 or 1 at each vertex. The boundary condition $v_0(x)$ is chosen as

$$v_0(x) = \begin{cases} f(x_1), & x_2 = 0, \quad 0 \le x_1 \le 1, \\ 0, & x_1 = 0, \quad 0 \le x_2 \le 1, \\ f(x_1), & x_2 = 1, \quad 0 \le x_1 \le 1, \\ 0, & x_1 = 1, \quad 0 \le x_2 \le 1, \end{cases}$$

where $f(x_1) = x_1(1 - x_1)$ (for MINS-SB) or $f(x_1) = \sin(4\pi x_1) + \frac{1}{10}\sin(120\pi x_1)$ (for MINS-OB). To define problem MINS-BC, we introduce, in MINS-SB, the following lower bound constraint:

$$v(x) \geq \sqrt{2} \qquad \text{whenever} \quad \frac{4}{9} \leq x_1, x_2 \leq \frac{5}{9},$$

thereby creating an obstacle problem where the surface is constrained in the middle of the domain. The fourth variant of the minimum surface problem, MINS-DMSA, is the Enneper problem proposed in Minpack-2, where the domain is now given by $\mathcal{D} = (-\frac{1}{2}, \frac{1}{2}) \times (-\frac{1}{2}, \frac{1}{2})$. The boundary condition is chosen on $\partial\mathcal{D}$ as

$$v_{\mathcal{D}}(x) = u^2 - v^2,$$

where u and v are the unique solutions to the equations

$$x_1 = u + uv^2 - \frac{1}{3}u^3, \quad x_2 = -v - u^2v + \frac{1}{3}v^3.$$

C.4 MEMBR: a membrane problem

We consider the problem suggested by Domorádová and Dostál (2007) given by

$$\min_{u \in \mathcal{K}} \int_{S_2} \Big(||\nabla u(x)||_2^2 + u(x) \Big),$$

where the boundary of S_2 is composed of three parts: $\Gamma_u = \{0\} \times [0, 1]$, $\Gamma_l = \{1\} \times [0, 1]$ and $\Gamma_f = [0, 1] \times \{0, 1\}$ and where $\mathcal{K} = \{u \in \mathcal{H}^1(S_2) \mid u(x) = 0 \text{ on } \Gamma_u \text{ and } l \leq u(x) \text{ on } \Gamma_l\}$. The obstacle l on the boundary Γ_l is defined by the upper part of the circle with the radius one and center $S = (1; 0.5; -1.3)$.

The solution of this problem can be interpreted as the displacement of the membrane under the traction defined by the unit density. The membrane is fixed on Γ_u and is not allowed to penetrate the obstacle on Γ_l. We discretized the problem by piecewise linear finite elements using a regular triangular grid.

C.5 IGNISC, DSSC and BRATU: three combustion - Bratu problems

We first consider the following optimal-control problem (IGNISC), introduced by Borzi and Kunisch (2006), and related to the solid-ignition model:

$$\min_{u \in \mathcal{H}_0^1(S_2)} \left[\int_{S_2} \Big(u(x) - z \Big)^2 + \frac{\beta}{2} \int_{S_2} \Big(e^{u(x)} - e^z \Big)^2 + \frac{\nu}{2} \int_{S_2} \|\Delta u(x) - \delta e^{u(x)}\|_2^2 \right].$$

For the numerical tests, we chose $\nu = 10^{-5}$, $\delta = 6.8$, $\beta = 6.8$ and $z = \frac{1}{\pi^2}$.

The second problem of this type is the steady-state combustion problem DSSC of Minpack 2, stated as the infinite-dimensional optimization problem

$$\min_{u \in \mathcal{H}_0^1(S_2)} \int_{S_2} \Big(\tfrac{1}{2} ||\nabla u(x)||_2^2 - \lambda e^{u(x)} \Big)$$

and $\lambda = 5$. This problem is the variational formulation of the boundary value problem

$$\begin{aligned} -\Delta u(x) &= \lambda e^{u(x)}, & x \in S_2, \\ u(x) &= 0, & x \in \partial S_2. \end{aligned}$$

The third variant is a simple least-squares formulation of the same problem, where we solve

$$\min_{u \in \mathcal{H}_0^1(S_2)} \int_{S_2} \|\Delta u(x) + \lambda e^{u(x)}\|_2^2,$$

with $\lambda = 6.8$. For all these convex problems, we use standard 5-point finite differences on a uniform grid.

C.6 NCCS and NCCO: two nonconvex optimal control problems

We introduce the nonlinear least-squares problem

$$\min_{u,v \in \mathcal{H}_0^1(S_2)} \left[\int_{S_2} (u(x) - u_0(x))^2 + \int_{S_2} (v(x) - v_0(x))^2 + \int_{S_2} \|\Delta u(x) - v(x)u(x) + f_0(x)\|_2^2 \right].$$

We distinguish two variants: the first with relatively smooth target functions and the second with more oscillatory ones. Theses functions $v_0(x)$ and $u_0(x)$ are defined on S_2 by

$$\begin{aligned} v_0(x) = u_0(x) &= \sin(6\pi x_1)\sin(2\pi x_2) && \text{(for NCCS)} \\ v_0(x) = u_0(x) &= \sin(128\pi x_1)\sin(32\pi x_2) && \text{(for NCCO)} \ . \end{aligned}$$

The function $f_0(x)$ is such that $-\Delta u_0(x) + v_0(x)u_0(x) = f_0(x)$ on S_2. This problem corresponds to a penalized version of a constrained optimal control problem, and is discretized using finite differences. The nonconvexity of the resulting discretized fine-grid problem has been assessed by a direct eigenvalue computation on the Hessian of the problem.

C.7 DPJB: pressure distribution in a journal bearing

The journal bearing problem arises in the determination of the pressure distribution in a thin film of lubricant between two circular cylinders. This problem is again proposed by Minpack 2, and is of the form

$$\min_{v \in \mathcal{K}} \tfrac{1}{2} \int_{\mathcal{D}} \Big(w_q(x) ||\nabla v(x)||_2^2 - \tfrac{1}{10} w_l(x) v(x) \Big),$$

where

$$w_q(x) = (1 + \tfrac{1}{10}\cos x_1)^3 \quad \text{and} \quad w_l(x) = \tfrac{1}{10}\sin x_1$$

for some constant $\epsilon \in (0,1)$ and $\mathcal{D} = (0, 2\pi) \times (0, 20)$. The convex set $\mathcal{K}$ is defined by $\mathcal{K} = \{v \in \mathcal{H}_0^1(\mathcal{D}) \mid v(x) \geq 0 \ \text{ on } \ \mathcal{D}\}$. A finite-element approach of this problem is obtained by minimizing over the space of piecewise linear functions v with values $v_{i,j}$ at $z_{i,j} \in \mathbb{R}^2$, which are the vertices of the regular triangulations of $\mathcal{D}$.

C.8 DEPT: an elastic-plastic torsion problem

The elastic-plastic torsion problem DEPT from Minpack 2 arises from the determination of the stress field on an infinitely long cylindric bar. The infinite-dimensional version of this problem is of the form

$$\min_{v\in\mathcal{K}} \tfrac{1}{2}\int_{S_2}\Big(\,||\nabla v(x)||_2^2 - 5v(x)\,\Big).$$

The convex set $\mathcal{K}$ is defined by $\mathcal{K} = \{v \in \mathcal{H}_0^1(S_2) \mid |v(x)| \leq \mathrm{dist}(x, \partial S_2) \;\text{ on }\; S_2\}$, where $\mathrm{dist}(., \partial S_2)$ is the distance function to the boundary of S_2. A finite-element approach of this problem is obtained by minimizing over the space of piecewise linear functions v with values $v_{i,j}$ at $z_{i,j} \in \mathbb{R}^2$ which are the vertices of the regular triangulations of S_2.

C.9 DODC: an optimal design with composite materials

The Minpack 2 DODC optimal design problem is defined by

$$\min_{v\in\mathcal{H}_0^1(S_2)} \int_{\mathcal{D}}\Big(\,\psi_\lambda(||\nabla v(x)||_2) + v(x)\,\Big),$$

where

$$\psi_\lambda(t) = \begin{cases} \frac{1}{2}\mu_2 t^2, & 0 \leq t \leq t_1, \\ \mu_2 t(t - \frac{1}{2}t_1), & t_1 \leq t \leq t_2, \\ \frac{1}{2}\mu_1(t^2 - t_2^2) + \mu_2 t_1(t_2 - \frac{1}{2}t_1), & t_2 \leq t, \end{cases}$$

with the breakpoints t_1 and t_2 defined by

$$t_1 = \sqrt{2\lambda\frac{\mu_1}{\mu_2}} \quad \text{and} \quad t_2 = \sqrt{2\lambda\frac{\mu_2}{\mu_1}},$$

and we choose $\lambda = 0.008$, $\mu_1 = 1$ and $\mu_2 = 2$. A finite-element approach for this problem is obtained by minimizing over the space of piecewise linear functions v with values $v_{i,j}$ at $z_{i,j} \in \mathbb{R}^2$ which are the vertices of the regular triangulations of S_2.

C.10 MOREBV: a nonlinear boundary value problem

The MOREBV problem is adapted (in infinite dimensions) from Moré, Garbow and Hillstrom (1981) and is described by

$$\min_{u\in\mathcal{H}_0^1(S_2)} \int \|\Delta u(x) - \tfrac{1}{2}[u(x) + \langle e, x\rangle + 1]^3\|_2^2,$$

where e is the vector of all ones. Once again, the problem is discretized by linear finite-elements on regular triangular grids.

Appendix D

Complete numerical results

We give here the complete numerical results for all test problems and all variants. The columns of the following tables report CPU time (in seconds), the number of matrix-vector products or smoothing cycles and the number of objective function/gradient/Hessian evaluations (in equivalent number of finest products, cycles and evaluations, like in (4.11)).

P2D	CPU	Mv prods	Eval f	Eval g	eval H
FM	26.05	13.52	4.66	3.38	1.33
MR	569.72	1494.99	2.67	2.67	1.33
MF	72.85	52.93	10.00	10.00	1.00
AF	1122.83	3022.00	4.00	4.00	1.00
DODC	CPU	Mv prods	Eval f	Eval g	eval H
FM	36.00	218.92	65.98	220.55	0.00
MR	184.23	4014.31	38.43	354.44	0.00
MF	58.58	282.99	93.00	399.00	0.00
AF	894.76	11472.00	493.00	4707.00	0.00
MINS-SB	CPU	Mv prods	Eval f	Eval g	eval H
FM	153.92	81.89	26.43	18.62	11.91
MR	3600.00	-	-	-	-
MF	3600.00	-	-	-	-
AF	3600.00	-	-	-	-
MINS-OB	CPU	Mv prods	Eval f	Eval g	eval H
FM	27.49	305.67	84.99	61.42	21.33
MR	116.73	1807.44	26.93	18.43	25.60
MF	70.44	564.15	261.00	185.00	69.00
AF	1545.63	5955.00	475.00	388.00	460.00
NCCS	CPU	Mv prods	Eval f	Eval g	eval H
FM	331.89	69.57	69.77	1100.27	0.00
MR	279.51	1342.26	2.68	57.50	0.00
MF	3600.00	-	-	-	-
AF	3600.00	-	-	-	-
MINS-DMSA	CPU	Mv prods	Eval f	Eval g	eval H
FM	18.23	88.74	26.89	138.65	0.00
MR	289.64	2860.34	26.31	242.01	0.00
MF	73.41	200.25	137.00	591.00	0.00
AF	1196.81	5677.00	428.00	4116.00	0.00
DPJB	CPU	Mv prods	Eval f	Eval g	eval H
FM	83.61	11.17	16.98	28.98	0.00
MR	247.71	341.66	5.02	17.02	0.00
MF	1390.02	297.00	297.00	306.00	0.00

AF	3600.00	-	-	-	-
IGNISC	CPU	Mv prods	Eval f	Eval g	eval H
FM	398.18	65.60	14.98	13.91	1.34
MR	488.22	1882.86	2.69	2.69	1.36
MF	398.34	257.11	60.00	46.00	1.00
AF	2330.42	11572.00	6.00	6.00	5.00
MEMBR	CPU	Mv prods	Eval f	Eval g	eval H
FM	153.96	76.73	98.43	98.43	1.33
MR	292.43	2103.35	3.00	3.00	1.33
MF	335.25	413.97	203.00	183.00	1.00
AF	1082.05	7423.00	43.00	43.00	1.00
DSSC	CPU	Mv prods	Eval f	Eval g	eval H
FM	12.11	3.41	1.93	4.85	0.00
MR	122.32	211.51	1.67	4.68	0.00
MF	1051.56	760.65	165.00	134.00	0.00
AF	3183.85	6012.00	6.00	42.00	0.00
MINS-BC	CPU	Mv prods	Eval f	Eval g	eval H
FM	140.02	402.25	551.00	540.88	31.64
MR	524.61	4055.91	413.59	400.60	47.15
MF	161.84	414.09	581.00	560.00	84.00
AF	2706.41	3935.00	1105.00	1001.00	1103.00
BRATU	CPU	Mv prods	Eval f	Eval g	eval H
FM	10.15	3.68	2.06	1.91	0.33
MR	91.71	203.00	1.67	1.67	0.33
MF	236.82	184.41	43.00	32.00	1.00
AF	2314.11	5458.00	6.00	6.00	4.00
DNT	CPU	Mv prods	Eval f	Eval g	eval H
FM	6.73	33.62	9.33	7.33	1.33
MR	4.58	246.40	2.66	2.66	1.33
MF	24.41	131.82	37.00	28.00	1.00
AF	5.20	299.00	3.00	3.00	1.00
NCCO	CPU	Mv prods	Eval f	Eval g	eval H
FM	224.20	44.01	35.33	791.37	0.00
MR	3589.62	17993.03	3.33	43.37	0.00
MF	3600.00	-	-	-	-
AF	3600.00	-	-	-	-
P3D	CPU	Mv prods	Eval f	Eval g	eval H
FM	28.78	39.38	8.92	8.64	1.33
MR	18.33	102.08	2.82	2.74	1.33
MF	47.47	64.75	12.00	12.00	1.00
AF	626.07	987.00	257.00	142.00	1.00
MOREBV	CPU	Mv prods	Eval f	Eval g	eval H
FM	41.73	12.83	4.54	3.60	0.33
MR	3600.00	-	-	-	-
MF	704.88	301.01	55.00	44.00	1.00
AF	3600.00	-	-	-	-
DEPT	CPU	Mv prods	Eval f	Eval g	eval H
FM	8.58	3.37	1.92	4.43	0.00
MR	95.44	206.38	1.66	4.25	0.00
MF	69.55	52.93	10.00	18.00	0.00
AF	1364.45	3019.00	4.00	12.00	0.00

Appendix E

A Retrospective Trust-Region Method for Unconstrained Optimization

This appendix presents the paper of Bastin, Malmedy, Mouffe, Toint and Tomanos (2009).

E.1 Introduction

Trust-region methods are well-known techniques in nonlinear nonconvex programming, whose concept has matured over more than thirty years (for an extensive coverage, see Conn et al., 2000). In such methods, one considers a model m_k of the objective function which is assumed to be adequate in a "trust region", which is a neighbourhood of the current iterate x_k. This neighbourhood is often represented by a ball in some norm, whose radius Δ_k is then updated from iteration k to iteration $k+1$ by considering how well m_k predicts the objective function value at iterate x_{k+1}. In retrospect, this might seem unnatural since the new radius Δ_{k+1} will determine the region in which a possibly updated model m_{k+1} is expected to predict the value of the objective function around x_{k+1}. Our aim in this paper is to propose a variant of the trust-region algorithm that determines Δ_{k+1} according to how well m_{k+1} predicts the value of the objective function at x_k, thereby synchronizing the radius update with the change in models.

The new method is motivated by applications in adaptive techniques which exploit the information made available during the optimization process in order to vary the accuracy of the objective function computation. These techniques typically appear in the context of a noisy objective function, where noise reduction can be achieved but at a significant cost. A first trust-region method with dynamic accuracy is described in Section 10.6 of Conn et al. (2000). The main idea there is to impose a model reduction larger than some multiple of the noise evaluated at both the current and candidate iterates. A cheaper nonmonotone approach has been developed in the context of nonlinear stochastic programming by Bastin, Cirillo and Toint (2006*a*), (see also Bastin, Cirillo and Toint, 2006*b*) more specifically for the minimization of sample average approximations (Shapiro, 2003) relying on Monte-Carlo sampling, a method

also known as sample-path optimization (Robinson, 1996). The main difference with respect to the work of Conn *et al.* is that it allows a reduction of the model smaller than the noise level. In both cases, the size of the model reduction is the main component to decide on the desired accuracy of the objective function: the adaptive mechanism is thus applied on the basis of past information, at the previous iterate, rather that at the current one. Our new proposal is then motivated by the hope of improving these techniques because the most relevant information on the model's quality at the current iterate would be used, instead of at the previous iterate.

This paper explores the theoretical properties and practical numerical potential of the new trust-region algorithm. We introduce the new method in Section 2, and study its convergence in the next section. Section 4 presents preliminary numerical experience on standard nonlinear problems. We conclude and examine perspectives for future research in Section 5.

E.2 A retrospective trust-region algorithm

We consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{E.1}$$

where f is a twice-continuously differentiable objective function which maps $\mathbb{R}^n$ into $\mathbb{R}$ and is bounded below. Trust-region methods are iterative processes, which, given a starting point x_0, construct a sequence $(x_k)_{k\geq 0}$ of iterates hopefully converging to a solution of (E.1). At each iteration k, a twice-continuously differentiable model m_k is defined which we trust inside a (typically Euclidean) ball $\mathcal{B}_k$ of radius $\Delta_k > 0$ centred at the current iterate x_k, called the *trust region*. A step s_k is then computed by (approximately) minimizing the model m_k inside the trust region $\mathcal{B}_k$. The trial point $x_k + s_k$ is then accepted as the next iterate x_{k+1} if and only if ρ_k, the ratio

$$\rho_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

of achieved reduction (in the objective function) to predicted reduction (in its local model m_k), is larger than a small positive constant η_1 (iteration k is then called *successful*). In the classical framework, the trust-region radius is updated at the end of each iteration: it is left unchanged or increased if the trial point is accepted (that is if $\rho_k \geq \eta_1$), and decreased otherwise. In this case, the new value Δ_{k+1} is chosen in the interval $[\gamma_0\|s_k\|, \gamma_1\|s_k\|]$ for some constants $0 < \gamma_0 < \gamma_1 < 1$. When ρ_k is negative, a quadratic fit of the model is used (as in Conn et al., 2000, p. 783), to determine a tentative new radius whose purpose is to ensure that the next iteration is very successful in the sense that $\rho_{k+1} \geq \eta_2$ for some $\eta_2 \in (\eta_1, 1)$. This value is given by $\theta_k\Delta_k$, where

$$\theta_k \stackrel{\text{def}}{=} \frac{(1-\eta_2)\langle \nabla_x f(x_k), s_k\rangle}{(1-\eta_2)[f(x_k) + \langle \nabla_x f(x_k), s_k\rangle] + \eta_2 m_k(x_k + s_k) - f(x_k + s_k)}. \tag{E.2}$$

Our new algorithm differs in that the trust-region radius is updated after each successful iteration k (that is at the beginning of iteration $k+1$) on the basis of the *retrospective* ratio

$$\tilde{\rho}_{k+1} \stackrel{\text{def}}{=} \frac{f(x_{k+1}) - f(x_{k+1} - s_k)}{m_{k+1}(x_{k+1}) - m_{k+1}(x_{k+1} - s_k)} = \frac{f(x_k) - f(x_k + s_k)}{m_{k+1}(x_k) - m_{k+1}(x_k + s_k)}$$

of achieved to predicted changes, while continuing to use ρ_k to decide whether the trial iterate may be accepted. Our method therefore distinguishes the two roles played by ρ_k in the classical algorithm: that of deciding acceptance of the trial iterate and that of determining the radius update. It also explicitely takes into account that m_{k+1}, not m_k, is used within the trust region of radius Δ_{k+1}. Thus, when the iterate has first been accepted, that is when $\rho_k \geq \eta_1$, we compute this radius by either increasing the current radius or leaving it unchanged if $\tilde{\rho}_k \geq \tilde{\eta}_1$ or decrease it otherwise. In this last case, it is again chosen in the interval $[\gamma_0\|s_k\|, \gamma_1\|s_k\|]$. Moreover, when $\tilde{\rho}_k$ is negative, a quadratic fit of the model is used as above to determine a tentative new radius which will make the next iteration very successful in the sense that $\tilde{\rho}_{k+1} \geq \tilde{\eta}_2$ for some $\tilde{\eta}_2 \in (\tilde{\eta}_1, 1)$. This value is given by $\tilde{\theta}_{k+1}\Delta_k$, where

$$\tilde{\theta}_{k+1} \stackrel{\text{def}}{=} \frac{-(1-\tilde{\eta}_2)\langle \nabla_x f(x_{k+1}), s_k\rangle}{(1-\tilde{\eta}_2)[f(x_{k+1}) - \langle \nabla_x f(x_{k+1}), s_k\rangle] + \tilde{\eta}_2 m_{k+1}(x_k) - f(x_k)}. \tag{E.3}$$

Notice that $\tilde{\theta}_{k+1}$ uses the gradient at the new point, rather than the old one as in (E.2).

This leads to the retrospective trust-region method described as Algorithm E.2.1, in which we leave the precise definitions of the model (at Step 1) and of "sufficient reduction" (at Step 3) for the next section.

E.3 Convergence theory

We now investigate the convergence properties of our algorithm. Since it can be considered as a variant of the basic trust-region method of Conn et al. (2000), we expect similar results and significant similarities in their proofs. In what follows, we have attempted to be explicit on the assumptions and properties, but to refer to Chapter 6 of this reference whenever possible.

Our assumptions are identical to those used for the basic trust-region method.

A.1 The Hessian of the objective function $\nabla_{xx} f$ is uniformly bounded, i.e. there exists a positive constant κ_{ufh} such that, for all $x \in \mathbb{R}^n$,

$$\|\nabla_{xx} f(x)\|_\infty \leq \kappa_{\text{ufh}}.$$

A.2 The model m_k is first-order coherent with the function f at each iteration x_k, i.e. their values and gradients are equal at x_k for all k:

$$m_k(x_k) = f(x_k) \quad \text{and} \quad g_k \stackrel{\text{def}}{=} \nabla_x m_k(x_k) = \nabla_x f(x_k).$$

A.3 The Hessian of the model $\nabla_{xx} m_k$ is uniformly bounded, i.e. there exists a constant $\kappa_{\text{umh}} \geq 1$ such that, for all $x \in \mathbb{R}^n$ and for all k,

$$\|\nabla_{xx} m_k(x)\|_\infty \leq \kappa_{\text{umh}} - 1.$$

A.4 The decrease on the model m_k is at least as much as a fraction of that obtained at the Cauchy point; i.e. there exists a constant $\kappa_{\text{mdc}} \in (0, 1)$ such that, for all k,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\|_\infty \min\left[\frac{\|g_k\|_\infty}{\beta_k}, \Delta_k\right]$$

with $\beta_k \stackrel{\text{def}}{=} 1 + \max_{x \in \mathcal{B}_k} \|\nabla_{xx} m_k(x)\|_\infty$.

Note that A.4 specifies the notion of "sufficient reduction" used in Step 3 of our algorithm, while the choice of m_k in Step 1 is limited by A.2 and A.3. We also note that $s_k \neq 0$ whenever $g_k \neq 0$ because of A.4.

E.3.1 Convergence to First-Order Critical Points

In this section, we prove that the retrospective trust-region algorithm is globally convergent to first-order critical points, in the sense that every limit point x_* of the sequence of iterates (x_k) produced by the algorithm E.2.1 satisfies

$$\nabla_x f(x_*) = 0$$

irrespective of the choice of the starting point x_0 and initial trust-region radius Δ_0.

We first give a bound on the error between the true objective function f and its current model m_k at the previous iterate x_{k-1}.

Theorem E.3.1 *Suppose that A.1–A.3 hold. Then we have that*

$$|f(x_k) - m_{k-1}(x_k)| \leq \kappa_{\text{ubh}} \Delta_{k-1}^2 \tag{E.8}$$

and, if iteration $k-1$ is successful, that

$$|f(x_{k-1}) - m_k(x_{k-1})| \leq \kappa_{\text{ubh}} \Delta_{k-1}^2 \tag{E.9}$$

where

$$\kappa_{ubh} \stackrel{\text{def}}{=} \max[\kappa_{ufh}, \kappa_{umh}]. \tag{E.10}$$

Proof. The bound (E.8) directly results from Theorem 6.4.1 in Conn et al. (2000). We thus only prove (E.9). Because the objective function and the model are C^2 functions, we may apply the mean value theorem on the objective function f and on the model m_k, and obtain from $x_{k-1} = x_k - s_{k-1}$ that

$$\begin{aligned} f(x_{k-1}) &= f(x_k) - \langle s_{k-1}, \nabla_x f(x_k) \rangle + \tfrac{1}{2} \langle s_{k-1}, \nabla_{xx} f(\xi_k) s_{k-1} \rangle & \text{(E.11)} \\ m_k(x_{k-1}) &= m_k(x_k) - \langle s_{k-1}, g_k \rangle + \tfrac{1}{2} \langle s_{k-1}, \nabla_{xx} m_k(\zeta_k) s_{k-1} \rangle & \text{(E.12)} \end{aligned}$$

for some ξ_k, ζ_k in the segment $[x_{k-1}, x_k]$.

Because of A.2, the objective function f and the model m_k have the same value and gradient at x_k. Thus, subtracting (E.12) from (E.11) and taking absolute values yields that

$$\begin{aligned} |f(x_{k-1}) - m_k(x_{k-1})| &= \tfrac{1}{2} |\langle s_{k-1}, \nabla_{xx} f(\xi_k) s_{k-1} \rangle - \langle s_{k-1}, \nabla_{xx} m_k(\zeta_k) s_{k-1} \rangle| \\ &\leq \tfrac{1}{2} [|\langle s_{k-1}, \nabla_{xx} f(\xi_k) s_{k-1} \rangle| + |\langle s_{k-1}, \nabla_{xx} m_k(\zeta_k) s_{k-1} \rangle|] \\ &\leq \tfrac{1}{2} (\kappa_{\text{ufh}} + \kappa_{\text{umh}} - 1) \|s_{k-1}\|^2 \\ &\leq \tfrac{1}{2} (\kappa_{\text{ufh}} + \kappa_{\text{umh}} - 1) \Delta_{k-1}^2, & \text{(E.13)} \end{aligned}$$

where we successively used the triangle inequality, the Cauchy-Schwarz inequality, the induced matrix norm properties, A.1, A.3, and the fact that $x_k \in \mathcal{B}_{k-1}$ implies that $\|s_{k-1}\| \leq \Delta_{k-1}$. So (E.9) clearly holds. □

Thus the analog of Theorem 6.4.1 of Conn et al. (2000) holds in our case, where we replace the forward difference $f(x_{k+1}) - m_k(x_{k+1})$ by its retrospective variant $f(x_{k-1}) - m_k(x_{k-1})$.

As our new ratio $\tilde{\rho}_k$ uses the reduction in m_k instead of the reduction in m_{k-1}, we are interested in a bound on their difference, which is provided by this next result.

Lemma E.3.2 *Suppose that A.1–A.3 hold. Then we have that, for every successful iteration $k-1$,*

$$|\,[m_{k-1}(x_{k-1}) - m_{k-1}(x_k)] - [m_k(x_{k-1}) - m_k(x_k)]\,| \leq 2\kappa_{ubh}\Delta_{k-1}^2. \tag{E.14}$$

Proof. Using the model differentiability, we apply the mean value theorem on the model m_{k-1}, and we obtain that

$$m_{k-1}(x_k) \;=\; m_{k-1}(x_{k-1}) + \langle s_{k-1}, g_{k-1}\rangle + \tfrac{1}{2}\,\langle s_{k-1}, \nabla_{xx} m_{k-1}(\psi_{k-1}) s_{k-1}\rangle \tag{E.15}$$

for some ψ_{k-1} in the segment $[x_{k-1}, x_k]$. Remember that (E.12) in the previous proof gives that

$$m_k(x_{k-1}) \;=\; m_k(x_k) - \langle s_{k-1}, g_k\rangle + \tfrac{1}{2}\,\langle s_{k-1}, \nabla_{xx} m_k(\zeta_k) s_{k-1}\rangle \tag{E.16}$$

for some ζ_k in the segment $[x_{k-1}, x_k]$. Substituting (E.15) and (E.16) inside the left-hand side of (E.14), and using A.3, the triangle inequality, the Cauchy-Schwarz inequality, and the induced matrix norm properties yield that

$$\begin{aligned}
&|\,[m_{k-1}(x_{k-1}) - m_{k-1}(x_k)] - [m_k(x_{k-1}) - m_k(x_k)]\,| \\
&\quad= \left|-\langle s_{k-1}, g_{k-1} - g_k\rangle - \tfrac{1}{2}\left(\langle s_{k-1}, \nabla_{xx} m_{k-1}(\psi_{k-1}) s_{k-1}\rangle + \langle s_{k-1}, \nabla_{xx} m_k(\zeta_k) s_{k-1}\rangle\right)\right| \\
&\quad\leq \|s_{k-1}\| \cdot \|g_{k-1} - g_k\| + \kappa_{\rm umh}\|s_{k-1}\|^2.
\end{aligned} \tag{E.17}$$

Now observe that, because of A.2, $\|g_{k-1} - g_k\| = \|\nabla_x f(x_{k-1}) - \nabla_x f(x_k)\|$. We then apply the mean value theorem on $\nabla_x f$ and obtain that

$$\nabla_x f(x_k) = \nabla_x f(x_{k-1}) + \int_0^1 \nabla_{xx} f(x_{k-1} + \alpha s_{k-1}) s_{k-1}\, d\alpha. \tag{E.18}$$

Thus the Cauchy-Schwarz inequality, and A.1 give that

$$\|g_{k-1} - g_k\| \leq \int_0^1 \|\nabla_{xx} f(x_{k-1} + \alpha s_{k-1})\| \cdot \|s_{k-1}\|_\infty\, d\alpha \leq \int_0^1 \kappa_{\rm ufh}\|s_{k-1}\|_\infty\, d\alpha = \kappa_{\rm ufh}\|s_{k-1}\|_\infty. \tag{E.19}$$

Substituting this bound in (E.17), we obtain that

$$|\,[m_{k-1}(x_{k-1}) - m_{k-1}(x_k)] - [m_k(x_{k-1}) - m_k(x_k)]\,| \leq (\kappa_{\rm ufh} + \kappa_{\rm umh})\|s_{k-1}\|^2 = 2\kappa_{\rm ubh}\Delta_{k-1}^2$$

where we finally use (E.10), and the fact that $x_k \in \mathcal{B}_{k-1}$. □

We conclude from this result that the denominators in the expression of $\tilde{\rho}_k$ and ρ_{k-1} differ by a quantity which is of the same order as the error between the model and the objective function. Using this observation, we are now capable of showing that the iteration must be successful if the radius is sufficiently small compared to the gradient, and also that the trust-region radius has to increase in this case.

Theorem E.3.3 *Suppose that A.1–A.4 hold. Suppose furthermore that* $g_k \neq 0$ *and that*

$$\Delta_{k-1} \leq \min\left[1-\eta_1, \frac{(1-\tilde{\eta}_2)}{(3-2\tilde{\eta}_2)}\right] \frac{\kappa_{mdc}}{\kappa_{ubh}} \|g_{k-1}\|. \tag{E.20}$$

Then iteration $k-1$ *is successful and*

$$\Delta_k \geq \Delta_{k-1}. \tag{E.21}$$

Proof. We first apply Theorem 6.4.2 of Conn et al. (2000) to deduce that iteration $k-1$ is successful and thus that $x_k = x_{k-1} + s_{k-1} \neq x_{k-1}$. Observe now that the constants $\tilde{\eta}_2$ and $\kappa_{\rm mdc}$ lie in the interval $(0,1)$, which implies that

$$\frac{(1-\tilde{\eta}_2)}{(3-2\tilde{\eta}_2)} < \frac{1}{2} < 1 \quad \text{and thus} \quad \kappa_{\rm mdc}\frac{(1-\tilde{\eta}_2)}{(3-2\tilde{\eta}_2)} < 1. \tag{E.22}$$

The conditions (E.20), (E.22), and (E.10), combined with the definition of β_{k-1} in A.4 imply that

$$\Delta_{k-1} < \frac{\|g_{k-1}\|}{\kappa_{\rm ubh}} < \frac{\|g_{k-1}\|}{\beta_{k-1}}. \tag{E.23}$$

As a consequence, A.4 immediately gives that

$$m_{k-1}(x_{k-1}) - m_{k-1}(x_k) \geq \kappa_{\rm mdc}\|g_{k-1}\| \min\left[\frac{\|g_{k-1}\|}{\beta_{k-1}}, \Delta_{k-1}\right] = \kappa_{\rm mdc}\|g_{k-1}\|\Delta_{k-1}. \tag{E.24}$$

On the other hand, we may apply Lemma E.3.2 and use the triangle inequality to obtain that

$$\begin{aligned} &|m_{k-1}(x_{k-1}) - m_{k-1}(x_k)| - |m_k(x_{k-1}) - m_k(x_k)| \\ &\quad \leq \; \big|\big[m_{k-1}(x_{k-1}) - m_{k-1}(x_k)\big] - \big[m_k(x_{k-1}) - m_k(x_k)\big]\big| \\ &\quad \leq \; 2\kappa_{\rm ubh}\Delta_{k-1}^2 \end{aligned}$$

and therefore, with (E.24), that

$$\begin{aligned} |m_k(x_{k-1}) - m_k(x_k)| &\geq |m_{k-1}(x_{k-1}) - m_{k-1}(x_k)| - 2\kappa_{\rm ubh}\Delta_{k-1}^2 \\ &\geq \kappa_{\rm mdc}\|g_{k-1}\|\Delta_{k-1} - 2\kappa_{\rm ubh}\Delta_{k-1}^2. \end{aligned} \tag{E.25}$$

Now (E.20) implies that $(3-2\tilde{\eta}_2)\kappa_{\rm ubh}\Delta_{k-1} \leq (1-\tilde{\eta}_2)\kappa_{\rm mdc}\|g_{k-1}\|_\infty$ and thus that

$$(1-\tilde{\eta}_2)(\kappa_{\rm mdc}\|g_{k-1}\|_\infty - 2\kappa_{\rm ubh}\Delta_{k-1}) \geq \kappa_{\rm ubh}\Delta_{k-1} > 0. \tag{E.26}$$

We finally may apply Theorem E.3.1 and deduce from A.2, (E.9), (E.25) and (E.26) that

$$|\tilde{\rho}_k - 1| = \left|\frac{f(x_{k-1}) - m_k(x_{k-1})}{m_k(x_{k-1}) - m_k(x_k)}\right| \leq \frac{\kappa_{\rm ubh}\Delta_{k-1}}{\kappa_{\rm mdc}\|g_{k-1}\| - 2\kappa_{\rm ubh}\Delta_{k-1}} \leq 1-\tilde{\eta}_2. \tag{E.27}$$

Therefore, $\tilde{\rho}_k \geq \tilde{\eta}_2$ and (E.6) then ensures that (E.21) holds. □

It is therefore guaranteed that the trust-region radius can not be decreased indefinitely if the current iterate is not near critically. This is ensured by the next theorem.

Theorem E.3.4 *Suppose that A.1–A.4 hold. Suppose furthermore that there exists a constant κ_{lbg} such that $\|g_k\| \geq \kappa_{lbg}$ for all k. Then there is a constant κ_{lbd} such that*

$$\Delta_k \geq \kappa_{lbd} \tag{E.28}$$

for all k.

Proof. The proof is the same as for Theorem 6.4.3 in Conn et al. (2000) except that

$$\kappa_{\rm lbd} = \min \left[1 - \eta_1, \frac{(1 - \tilde{\eta}_2)}{(3 - 2\tilde{\eta}_2)} \right] \frac{\gamma_1 \kappa_{\rm mdc} \kappa_{\rm lbg}}{\kappa_{\rm ubh}}.$$

□

From here on, the proof for the basic trust region applies without change. We first deduce the global convergence of the algorithm to first-order critical points when it generates only finitely many successful iterations.

Theorem E.3.5 *Suppose that A.1–A.4 hold. Suppose furthermore that there are only finitely many successful iterations. Then $x_k = x_*$ for all sufficiently large k and x_* is first-order critical.*

Proof. The same argument as in Theorem 6.4.4 in Conn et al. (2000) may be applied since the radius update is identical to that of the basic trust region method for unsuccessful iterations. □

Finally, the next two results ensure the global convergence of the algorithm to first-order critical points, by showing in a first step that at least one accumulation point of the iterates sequence is first-order critical.

Theorem E.3.6 *Suppose that A.1–A.4 hold. Then one has that*

$$\liminf_{k \to \infty} \|\nabla_x f(x_k)\| = 0. \tag{E.29}$$

Proof. See Theorem 6.4.5 in Conn et al. (2000). □

As for the basic trust-region method, this can be extended to show that all limit points are first-order critical.

Theorem E.3.7 *Suppose that A.1–A.4 hold. Then one has that*

$$\lim_{k \to \infty} \|\nabla_x f(x_k)\| = 0. \tag{E.30}$$

Proof. See Theorem 6.4.6 in Conn et al. (2000). □

E.3.2 Convergence to Second-Order Critical Points

We now investigate the possibility to exploit second-order information on the objective function, with the aim of ensuring convergence to second-order critical points, i.e. points x_* such that

$$\nabla_x f(x_*) = 0 \quad \text{and} \quad \nabla_{xx} f(x_*) \text{ is positive semidefinite.}$$

Of course, we need to clarify what we precisely mean by "second-order information". We therefore introduce the following additional assumptions:

A.5 The model is asymptotically second-order coherent with the objective function near first-order critical points, i.e.

$$\lim_{k\to\infty} \|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| = 0 \text{ whenever } \lim_{k\to\infty} \|g_k\| = 0.$$

A.6 The Hessian of every model m_k is Lipschitz continuous, that is, there exists a constant κ_{lch} such that, for all k,

$$\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\|_\infty \leq \kappa_{\text{lch}} \|x - y\|_\infty$$

for all $x, y \in \mathcal{B}_k$.

A.7 If the smallest eigenvalue τ_k of the Hessian of the model m_k at x_k is negative, then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min(\tau_k^2, \Delta_k^2)$$

for some constant $\kappa_{\text{sod}} \in (0, \frac{1}{2})$.

These assumptions are identical to those used in Sections 6.5 and 6.6 of Conn et al. (2000) for the basic trust-region method. In fact, the second-order convergence properties of the retrospective trust-region method also turn out to be exactly the same as those of the basic trust-region method, and their proofs can essentially been borrowed from this case, with the exception of Lemma 6.5.3. We therefore need to present a proof of that particular result for the new method. As we indicate below, all other results generalize without change and we only mention them for the sake of clarity.

In our analog of Lemma 6.5.3, we assume that the model reduction is eventually significant in the sense that it is at least of the same order as the error between the model and the objective function. We then show that the trust-region radius becomes asymptotically irrelevant if the steps tend to zero.

Lemma E.3.8 *Suppose that A.1–A.3, and A.5 hold. Suppose also that there exists a sequence (k_i) and a constant $\kappa_{mqd} > 0$ such that*

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{mqd} \|s_{k_i}\|^2 > 0 \tag{E.31}$$

for all i sufficiently large. Finally, suppose that

$$\lim_{i\to\infty} \|s_{k_i}\| = 0.$$

Then iteration k_i is successful and

$$\tilde{\rho}_{k_i+1} \geq \tilde{\eta}_2 \;\; \textit{and} \;\; \Delta_{k_i+1} \geq \Delta_{k_i} \tag{E.32}$$

for i sufficiently large.

Proof. We first apply Lemma 6.5.3 of Conn et al. (2000) to deduce that every iteration k_i is successful for i sufficiently large. Now, consider k_i one such iteration. The equations (E.11) and (E.12) imply that for some ξ_{k_i+1} and ζ_{k_i+1} in the segment $[x_{k_i}, x_{k_i+1}]$,

$$\begin{aligned} |\tilde{\rho}_{k_i+1} - 1| &= \left| \frac{f(x_{k_i}) - m_{k_i+1}(x_{k_i})}{m_{k_i+1}(x_{k_i}) - m_{k_i+1}(x_{k_i+1})} \right| \\ &= \left| \frac{\langle s_{k_i}, \nabla_{xx} f(\xi_{k_i+1}) s_{k_i} \rangle - \langle s_{k_i}, \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1}) s_{k_i} \rangle}{- \langle s_{k_i}, g_{k_i+1} \rangle + \frac{1}{2} \langle s_{k_i}, \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1}) s_{k_i} \rangle} \right| \\ &\leq \frac{\|s_{k_i}\|_\infty^2 \cdot \|\nabla_{xx} f(\xi_{k_i+1}) - \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1})\|_\infty}{\left| - \langle s_{k_i}, g_{k_i+1} \rangle + \frac{1}{2} \langle s_{k_i}, \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1}) s_{k_i} \rangle \right|} \end{aligned} \tag{E.33}$$

where we also used the Cauchy-Schwarz inequality. By substituting $g_{k_i+1} = \nabla_x f(x_{k_i+1})$ (because of A.2) with its expression in (E.18), the denominator D of the latter fraction can be rewritten as

$$D = \left| - \left\langle s_{k_i}, g_{k_i} + \int_0^1 \nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) s_{k_i} \, d\alpha \right\rangle + \tfrac{1}{2} \langle s_{k_i}, \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1}) s_{k_i} \rangle \right|.$$

Then, replacing $- \langle s_{k_i}, g_{k_i} \rangle$ by its expression in (E.15), we obtain

$$\begin{aligned} D &= \Big| m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i+1}) + \tfrac{1}{2} \langle s_{k_i}, \nabla_{xx} m_{k_i}(\psi_{k_i}) s_{k_i} \rangle \\ &\qquad + \tfrac{1}{2} \langle s_{k_i}, \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1}) s_{k_i} \rangle - \left\langle s_{k_i}, \int_0^1 \nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) s_{k_i} \, d\alpha \right\rangle \Big| \end{aligned}$$

for some ψ_{k_i} in the segment $[x_{k_i}, x_{k_i+1}]$. The triangle inequality, properties of the integral, (E.31), and Cauchy-Schwarz inequality give therefore the following lower bound on D:

$$\begin{aligned} D &\geq |m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i+1})| \\ &\quad - \tfrac{1}{2} \Big| \left\langle s_{k_i}, \int_0^1 [\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i}(\psi_{k_i})] s_{k_i} \, d\alpha \right\rangle \\ &\qquad + \left\langle s_{k_i}, \int_0^1 [\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1})] s_{k_i} \, d\alpha \right\rangle \Big| \\ &\geq \kappa_{\text{mqd}} \|s_{k_i}\|_\infty^2 - \tfrac{1}{2} \|s_{k_i}\|_\infty \int_0^1 \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i}(\psi_{k_i})\| \cdot \|s_{k_i}\|_\infty \, d\alpha \\ &\qquad - \tfrac{1}{2} \|s_{k_i}\|_\infty \int_0^1 \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1})\| \cdot \|s_{k_i}\|_\infty \, d\alpha \\ &\geq \|s_{k_i}\|_\infty^2 (\kappa_{\text{mqd}} - \tfrac{1}{2} \epsilon_i) \end{aligned} \tag{E.34}$$

where

$$\epsilon_i \stackrel{\text{def}}{=} \int_0^1 \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i}(\psi_{k_i})\| \, d\alpha + \int_0^1 \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i+1}(\zeta_{k_i+1})\| \, d\alpha.$$

The triangle inequality now implies that

$$\begin{aligned} \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} m_{k_i}(\psi_{k_i})\|_\infty &\leq \|\nabla_{xx} f(x_{k_i} + \alpha s_{k_i}) - \nabla_{xx} f(x_{k_i})\|_\infty \\ &\quad + \|\nabla_{xx} f(x_{k_i}) - \nabla_{xx} m_{k_i}(x_{k_i})\|_\infty + \|\nabla_{xx} m_{k_i}(x_{k_i}) - \nabla_{xx} m_{k_i}(\psi_{k_i})\|_\infty \end{aligned} \tag{E.35}$$

and, similarly, that

$$\|\nabla_{xx}f(x_{k_i}+\alpha s_{k_i})-\nabla_{xx}m_{k_i+1}(\zeta_{k_i+1})\|_\infty \leq \|\nabla_{xx}f(x_{k_i}+\alpha s_{k_i})-\nabla_{xx}f(x_{k_i+1})\|_\infty + \|\nabla_{xx}f(x_{k_i+1})-\nabla_{xx}m_{k_i+1}(x_{k_i+1})\|_\infty + \|\nabla_{xx}m_{k_i+1}(x_{k_i+1})-\nabla_{xx}m_{k_i+1}(\zeta_{k_i+1})\|_\infty. \tag{E.36}$$

Since we now observe that

$$\|(x_{k_i}+\alpha s_{k_i})-x_{k_i}\|_\infty \leq \|s_{k_i}\|_\infty, \qquad \|\psi_{k_i}-x_{k_i}\|_\infty \leq \|s_{k_i}\|_\infty,$$
$$\|(x_{k_i}+\alpha s_{k_i})-x_{k_i+1}\|_\infty \leq \|s_{k_i}\|_\infty, \qquad \|\zeta_{k_i+1}-x_{k_i+1}\|_\infty \leq \|s_{k_i}\|_\infty,$$

we may deduce that both

$$\|\nabla_{xx}f(x_{k_i}+\alpha s_{k_i})-\nabla_{xx}m_{k_i}(\psi_{k_i})\| \text{ and } \|\nabla_{xx}f(x_{k_i}+\alpha s_{k_i})-\nabla_{xx}m_{k_i+1}(\zeta_{k_i+1})\|$$

converge to zero with $\|s_{k_i}\|$ because the first and third terms of the right-hand side of (E.35) and (E.36) tend to zero by continuity of the objective function's and model's Hessians, and because the middle term in the right-hand side of these inequalities also converges to zero because of A.5 and Theorem E.3.7. As a consequence, $\epsilon_i \leq \kappa_{\rm mqd}$ when i is sufficiently large, and therefore, combining (E.33) and (E.34), and using the triangle inequality, we obtain

$$\begin{aligned} |\tilde{\rho}_{k_i+1}-1| &\leq \frac{2}{\kappa_{\rm mqd}}\|\nabla_{xx}f(\xi_{k_i+1})-\nabla_{xx}m_{k_i+1}(\zeta_{k_i+1})\| \\ &\leq \frac{2}{\kappa_{\rm mqd}}\Big[\|\nabla_{xx}f(\xi_{k_i+1})-\nabla_{xx}f(x_{k_i+1})\| \\ &\qquad + \|\nabla_{xx}f(x_{k_i+1})-\nabla_{xx}m_{k_i+1}(x_{k_i+1})\| \\ &\qquad + \|\nabla_{xx}m_{k_i+1}(x_{k_i+1})-\nabla_{xx}m_{k_i+1}(\zeta_{k_i+1})\|\Big] \end{aligned} \tag{E.37}$$

By the same reasoning as for (E.35)–(E.36), the right-hand side of (E.37) tends to zero when i goes to infinity, and $\tilde{\rho}_{k_i+1}$ therefore tends to 1. It is thus larger than $\tilde{\eta}_2 < 1$ for i sufficiently large and (E.32) follows. □

As in Lemma 6.5.4 of Conn et al. (2000), we may apply this result to the entire sequence of iterates and deduce that all iterations are eventually successful and the trust-region radius bounded away from zero.

From here on, the theory in Conn et al. (2000) generalizes without significant change, yielding the following results.

Theorem E.3.9 *Suppose that A.1–A.5 hold and that x_{k_i} is a subsequence of the iterates generated by Algorithm RTR converging to a first-order critical point x_* where the Hessian of the objective function $\nabla_{xx}f(x_*)$ is positive definite. Suppose furthermore that $s_k \neq 0$ for all k sufficiently large. Then the complete sequence of iterates converges to x_*, all iterations are eventually very successful, and the trust-region radius Δ_k is bounded away from zero.*

Proof. See Theorem 6.5.5 in Conn et al. (2000). □

We now proof that if the sequence of iterates remains in a compact set, then the existence of at least one second-order critical accumulation point is guaranteed.

Theorem E.3.10 *Suppose that A.1–A.7 hold and that all iterates remain in some compact set. Then there exists at least one limit point x_* of the sequence of iterates x_k produced by Algorithm RTR, which is second-order critical.*

Proof. See Theorem 6.6.5 in Conn et al. (2000). □

By just strengthening the radius update rule by requiring that

$$\text{if } \tilde{\rho}_k \geq \tilde{\eta}_2 \text{ and } \Delta_k \leq \Delta_{\max}, \text{ then } \Delta_{k+1} \in [\gamma_3 \Delta_k, \gamma_4 \Delta_k] \tag{E.38}$$

for some $\gamma_4 \geq \gamma_3 > 1$ and some $\Delta_{\max} > 0$, we moreover obtain the second-order criticality of any limit point of the sequence of iterates generated by Algorithm RTR.

Theorem E.3.11 *Suppose that A.1–A.7, and (E.38) hold and let x_* be any limit point of the sequence of iterates. Then x_* is a second-order critical point.*

Proof. See Theorem 6.6.8 in Conn et al. (2000). □

Thus the retrospective trust-region algorithm shares all the (interesting) convergence properties of the basic trust-region method under the same assumptions. We conclude this theory section by noting that the above convergence results are still valid if one replaces the Euclidean norm by any (possibly iteration dependent) uniformly equivalent norm, thereby allowing problem scaling and preconditioning.

E.4 Preliminary numerical experience

We now consider the numerical behaviour of the new algorithm, in comparison with the basic trust-region algorithm BTR (see page 116 of Conn et al. (2000)). We test both algorithms on all of the 146 unconstrained problems of the CUTEr collection (see Gould, Orban and Toint, 2003). For the problems whose dimension may be changed, we chose a reasonably small value in order not to overload the CUTEr interface with MATLAB. The starting points are the standard ones provided by the CUTEr library.

For the basic algorithm, the trust-region radius update was implemented by using the rule proposed in Conn et al. (2000), p. 783:

$$\Delta_{k+1} = \begin{cases} \max[\gamma_2 \|s_k\|, \Delta_k] & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \gamma_1 \|s_k\| & \text{if } \rho_k \in [0, \eta_1), \\ \min[\gamma_1 \|s_k\|, \max[\gamma_0, \theta_k] \Delta_k] & \text{if } \rho_k < 0, \end{cases}$$

where γ_0 is fixed at 0.0625, γ_1 at 0.25, γ_2 at 2.5, η_1 at 0.05 and η_2 at 0.9 and where θ_k is given by (E.2). To avoid biasing the comparison, we have decided to make as few adaptations as possible to that rule in our retrospective variant (i.e. Step 2 in Algorithm E.2.1). Thus, if iteration k is unsuccessful, i.e. $\rho_k < \eta_1$ and consequently $x_k = x_{k+1}$, we also decrease the trust-region using the above rule. If, on the contrary, iteration k is successful, i.e. $\rho_k \geq \eta_1$, the trust-region is updated using the procedure described in Step 2 of Algorithm E.2.1 where we choose the same values as above for γ_0, γ_1 and γ_2, and take $\tilde{\eta}_1 = \eta_1 = 0.05$ and $\tilde{\eta}_2 = \eta_2 = 0.9$. The model was

chosen, in both cases, to be the exact Taylor's series truncated to second-order, and the minimizer of the model inside the trust-region, was computed either exactly using the Moré-Sorensen algorithm (see Moré and Sorensen, 1983) or approximately using the Steihaug-Toint algorithm (see Steihaug, 1983, Toint, 1981). In this case, the conjugate gradient iterations are stopped if the trust-region boundary is met or as soon as the models' gradient satisfies the condition

$$\|\nabla_x m_k(x_k + s)\| \leq \min[0.1, \|\nabla_x m_k(x_k)\|^{1/2}]\|\nabla_x m_k(x_k)\|.$$

We considered that the iterative process converged when the Euclidean norm of the gradient became smaller than 10^{-5}. Failure was declared if the algorithm did not converge within the maximum number of 50 000 iterations.

We chose to compare the number of iterations to achieve convergence instead of the CPU time or number of function evaluations. Indeed, the cost per iteration is the same for both algorithms and they both evaluate the objective function once per iteration and compute one gradient at every successful iteration. Moreover, timings in MATLAB are often difficult to interpret.

All runs were performed in Matlab v. 7.1.0.183 (R14) Service Pack 3 on a 3.2 Ghz Intel single-core processor computer with 2 GB of RAM. Figure E.1 represents the comparison by a performance profile of the number of iterations of the two algorithms. Performance profiles give, for every $\sigma \geq 1$, the proportion $p(\sigma)$ of test problems on which each considered algorithmic variant has a performance within a factor σ of the best (see Dolan and Moré, 2002, for a more complete discussion). In this figure, we have only kept the problems for which both algorithms converged to the same local solution (we excluded `BIGGS6`, `BROYDN7D`, `CHAINWOO`, `FLETCHBV`, `LOGHAIRY`, `MEYER3`, `NONCVXU2`, `NONCVXUN`, `SENSORS`, `TOINTGSS` and `VIBRBEAM`). If the subproblem is solved approximately, both algorithms failed on `PALMER1C`, `SBRYBND`, `SCOSINE`, `SCURLY10`, `SCURLY20` and `SCURLY30`. Moreover, RTR failed on `FLETCBV3`, which was solved by BTR. On the other hand, if the subproblem is solved exactly, both algorithms failed on `FLETCBV3` and BTR failed on `SCOSINE`, which was solved by RTR. Note also the number of iterations needed to reach convergence with the RTR algorithm on the highly nonconvex `HUMPS` and `LOGHAIRY` problems is much higher than for the BTR algorithm. The complete numerical results are given in Appendix E.6.

Our results show that the retrospective algorithm performs as well as the classical one and is just as reliable if the trust-region subproblem is solved approximately. However, if the problem size or structure allows an exact solution, the retrospective algorithm is then significantly more efficient (the improvement is typically of only a few iterations, but is very consistent) and just as reliable. A detailed analysis of our results shows that RTR is in general slightly more conservative than BTR in that it tends to take marginally shorter steps. However, this does not seem to alter performance in a negative way. In particular the longer steps of BTR often result in a larger proportion of unsuccesful iterations (this may be deduced from the result table since the number of unsuccessful iterations is given by the difference between the number of iterations and the number of gradient evaluations). We also note that the choice of an accurate minimization of Newton's model in the trust region also appears to be considerably more efficient than an approximate one, at least in terms of the number of iterations needed for convergence, irrespective of the choice between BTR and RTR. As a consequence, the retrospective variant is clearly at its best when the cost of evaluating the objective function and gradient dominates

that of the overall iteration. Additional test not reported here also indicate that both algorithms are essentially undistinguishable when quasi-Newton approximations (SR1 or BFGS) are used instead of the true Hessian. This is perhaps not surprising since the corresponding variants, which use exact solutions of approximate models, may also be interpreted as using approximate solutions of exact models.

E.5 Conclusion and perspectives

We have introduced a natural variant of the basic trust-region algorithm, where the most recent model information is exploited to update the trust-region radius. We have also shown that limit points of sequences of iterates produced by the new algorithm are second-order critical points for the minimization problem. Our preliminary numerical experiments indicate that the method is advantageous when the model is good and its quality exploited by an accurate subproblem solution. Moreover this advantage is obtained at essentially zero cost.

As indicated in the introduction, this new method is especially interesting for adaptive techniques for noisy functions. The potential of the new approach is to exploit the most recent information on the noise to improve numerical performance. Research along this line is ongoing.

Other applications of the same idea are also possible across the wide class of trust-region methods, constrained and unconstrained.

E.6 Appendix

Here is the set of results from our tests. For each problem, we report its number of variables (n), the number of iterations ($iter$), the number of gradient evaluations ($\#g$) and the best objective function value found (f). The symbol $>$ indicates that the iteration limit (fixed at 100 000) was exceeded. The columns “LS” contains a star for least-squares problems.

Algorithm E.2.1: Retrospective trust-region algorithm (RTR)

Step 0: Initialisation. An initial point x_0 and initial trust-region radius $\Delta_0 > 0$ are given. The constants η_1, $\tilde{\eta}_1$, $\tilde{\eta}_2$, γ_0, γ_1 and γ_2 are also given and satisfy $0 < \eta_1 < 1$, $0 < \tilde{\eta}_1 \leq \tilde{\eta}_2 < 1$ and $0 < \gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$. Compute $f(x_0)$ and set $k = 0$.

Step 1: Model definition. Select a twice-continuously differentiable model m_k defined in $\mathcal{B}_k$.

Step 2: Retrospective trust-region radius update. If $k = 0$, go to Step 3. If $x_k = x_{k-1}$, then choose

$$\Delta_k = \begin{cases} \gamma_1 \|s_{k-1}\| & \text{if } \rho_{k-1} \in [0, \eta_1), \\ \min[\gamma_1 \|s_{k-1}\|, \max[\gamma_0, \theta_{k-1}]\Delta_{k-1}] & \text{if } \rho_{k-1} < 0, \end{cases} \tag{E.4}$$

where θ_{k-1} is defined in (E.2). Else, define

$$\tilde{\rho}_k = \frac{f(x_{k-1}) - f(x_k)}{m_k(x_{k-1}) - m_k(x_k)} \tag{E.5}$$

and choose

$$\Delta_k = \begin{cases} \max[\gamma_2 \|s_{k-1}\|, \Delta_{k-1}] & \text{if } \tilde{\rho}_k \geq \tilde{\eta}_2, \\ \Delta_{k-1} & \text{if } \tilde{\rho}_k \in [\tilde{\eta}_1, \tilde{\eta}_2), \\ \gamma_1 \|s_{k-1}\| & \text{if } \tilde{\rho}_k \in [0, \tilde{\eta}_1), \\ \min[\gamma_1 \|s_{k-1}\|, \max[\gamma_0, \tilde{\theta}_k]\Delta_{k-1}] & \text{if } \tilde{\rho}_k < 0, \end{cases} \tag{E.6}$$

where $\tilde{\theta}_k$ is defined in (E.3).

Step 3: Step calculation. Compute a step s_k that "sufficiently reduces the model" m_k and such that $x_k + s_k \in \mathcal{B}_k$.

Step 4: Acceptance of the trial point. Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \tag{E.7}$$

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$ and compute $\nabla_x f(x_{k+1})$; otherwise define $x_{k+1} = x_k$. Increment k by 1 and go to Step 1.

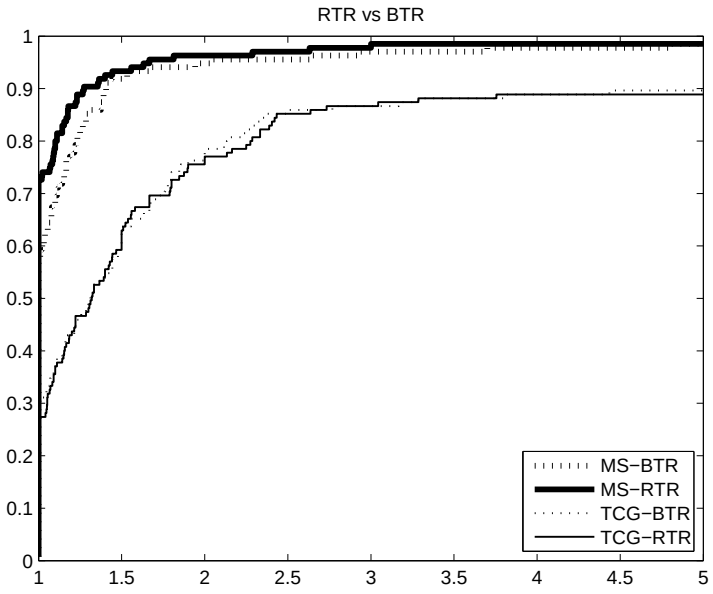


Figure E.1: Performance profile comparing the number of iterations of the RTR and BTR algorithms

			Moré-Sorensen						Steihaug-Toint					
			BTR			RTR			BTR			RTR		
Name	LS	n	iter	#g	f	iter	#g	f	iter	#g	f	iter	#g	f
AKIVA		2	6	7	6.1660e+00	6	7	6.1660e+00	8	9	6.1660e+00	8	9	6.1660e+00
ALLINITU		4	7	8	5.7444e+00	7	8	5.7444e+00	5	6	5.7444e+00	5	6	5.7444e+00
ARGLINA	*	200	5	6	2.0000e+02	5	6	2.0000e+02	5	6	2.0000e+02	5	6	2.0000e+02
ARWHEAD		100	5	6	6.5947e−14	5	6	6.5947e−14	5	6	0.0000e+00	5	6	0.0000e+00
BARD	*	3	9	9	8.2149e−03	9	9	8.2149e−03	13	13	8.2149e−03	13	13	8.2149e−03
BDQRTIC	*	100	10	11	3.7877e+02	10	11	3.7877e+02	13	14	3.7877e+02	13	14	3.7877e+02
BEALE	*	2	9	9	1.9232e−16	8	8	4.5813e−14	7	8	7.3194e−12	7	8	7.3194e−12
BIGGS6	*	6	6094	4585	2.4268e−01	6021	4685	2.4268e−01	149	135	8.9467e−09	149	138	1.6487e−07
BOX3	*	3	7	8	1.5192e−11	7	8	1.5192e−11	8	9	2.3841e−15	8	9	2.3841e−15
BRKMCC		2	2	3	1.6904e−01	2	3	1.6904e−01	3	4	1.6904e−01	3	4	1.6904e−01
BROWNAL	*	200	24	20	5.3204e−23	32	27	1.2675e−15	5	6	1.4731e−09	5	6	1.4731e−09
BROWNBS	*	2	29	29	0.0000e+00	29	29	0.0000e+00	51	52	0.0000e+00	55	56	0.0000e+00
BROWNDEN	*	4	10	11	8.5822e+04	10	11	8.5822e+04	11	12	8.5822e+04	11	12	8.5822e+04
BROYDN7D		100	24	21	3.9739e+01	23	21	3.9771e+01	35	31	3.9660e+01	31	27	3.9660e+01
BRYBND	*	100	17	13	2.0687e−28	12	12	1.4121e−23	11	12	2.8661e−17	11	12	2.8661e−17
CHAINWOO	*	100	53	44	1.0000e+00	50	45	1.0000e+00	300	228	5.5035e+01	162	141	3.2191e+01
CHNROSNB	*	50	57	48	1.8917e−13	54	50	2.4837e−21	78	61	6.7337e−14	64	59	2.2256e−15
CLIFF		2	27	28	1.9979e−01	27	28	1.9979e−01	30	31	1.9979e−01	30	31	1.9979e−01
COSINE		100	6	7	-9.9000e+01	6	7	-9.9000e+01	10	10	-9.9000e+01	10	10	-9.9000e+01
CRAGGLVY		202	15	16	6.6741e+01	15	16	6.6741e+01	16	17	6.6741e+01	16	17	6.6741e+01
CUBE	*	2	37	31	9.3052e−12	35	31	1.9212e−15	44	38	1.2297e−12	42	37	1.7564e−13
CURLY10	*	50	9	10	-5.0158e+03	9	10	-5.0158e+03	18	18	-5.0158e+03	18	18	-5.0158e+03
CURLY20	*	50	8	9	-5.0158e+03	8	9	-5.0158e+03	18	18	-5.0158e+03	18	18	-5.0158e+03
CURLY30	*	50	13	13	-5.0158e+03	13	13	-5.0158e+03	17	16	-5.0158e+03	20	19	-5.0158e+03
DECONVU	*	61	25	19	1.9290e−10	19	16	1.7251e−08	22	19	3.9035e−08	22	20	3.9966e−08
DENSCHNA		2	5	6	2.2139e−12	5	6	2.2139e−12	5	6	1.2000e−15	5	6	1.2000e−15
DENSCHNB	*	2	4	5	3.3850e−16	4	5	3.3850e−16	6	7	7.9948e−14	6	7	7.9948e−14
DENSCHNC	*	2	10	11	2.1777e−20	10	11	2.1777e−20	9	10	1.8423e−13	9	10	1.8423e−13
DENSCHND	*	3	37	33	1.1392e−08	38	34	1.1392e−08	30	31	1.3753e−08	30	31	1.3753e−08
DENSCHNE	*	3	9	10	8.7102e−19	9	10	8.7102e−19	16	16	4.4587e−19	15	16	7.3809e−13
DENSCHNF	*	2	6	7	6.5132e−22	6	7	6.5132e−22	6	7	6.5132e−22	6	7	6.5132e−22
DIXMAANA		150	7	8	1.0000e+00	7	8	1.0000e+00	9	10	1.0000e+00	9	10	1.0000e+00
DIXMAANB		150	11	11	1.0000e+00	11	11	1.0000e+00	9	10	1.0000e+00	9	10	1.0000e+00
DIXMAANC		150	11	11	1.0000e+00	11	11	1.0000e+00	10	11	1.0000e+00	10	11	1.0000e+00
DIXMAAND		150	14	13	1.0000e+00	14	13	1.0000e+00	11	12	1.0000e+00	11	12	1.0000e+00
DIXMAANE		150	10	10	1.0000e+00	11	11	1.0000e+00	11	11	1.0000e+00	11	12	1.0000e+00
DIXMAANF		150	15	14	1.0000e+00	14	13	1.0000e+00	12	13	1.0000e+00	12	13	1.0000e+00
DIXMAANG		150	15	14	1.0000e+00	15	14	1.0000e+00	13	14	1.0000e+00	13	14	1.0000e+00
DIXMAANH		150	18	16	1.0000e+00	19	17	1.0000e+00	14	15	1.0000e+00	14	15	1.0000e+00

			Moré-Sorensen						Steihaug-Toint					
			BTR			RTR			BTR			RTR		
Name	LS	n	iter	#g	f	iter	#g	f	iter	#g	f	iter	#g	f
DIXMAANI		150	14	14	1.0000e+00	16	16	1.0000e+00	13	14	1.0000e+00	13	14	1.0000e+00
DIXMAANJ		150	25	21	1.0000e+00	18	16	1.0000e+00	18	17	1.0000e+00	19	18	1.0000e+00
DIXMAANK		150	23	20	1.0000e+00	19	17	1.0000e+00	22	20	1.0000e+00	20	19	1.0000e+00
DIXMAANL		150	23	20	1.0000e+00	25	22	1.0000e+00	15	16	1.0000e+00	15	16	1.0000e+00
DIXON3DQ		100	4	5	1.1710e−29	4	5	1.1710e−29	8	9	0.0000e+00	8	9	0.0000e+00
DJTL		2	105	71	-8.9515e+03	104	74	-8.9515e+03	231	161	-8.9515e+03	253	183	-8.9515e+03
DQDRTIC		100	5	6	2.3990e−28	5	6	2.3990e−28	9	10	1.7453e−17	9	10	1.7453e−17
DQRTIC		100	29	30	2.8059e−08	29	30	2.8059e−08	29	30	3.5899e−08	29	30	3.5899e−08
EDENSCH		100	19	18	6.0328e+02	20	19	6.0328e+02	17	18	6.0328e+02	17	18	6.0328e+02
EG2		100	3	4	-9.8947e+01	3	4	-9.8947e+01	3	4	-9.8947e+01	3	4	-9.8947e+01
EIGENALS	*	110	20	21	5.0766e−21	20	20	1.1113e−12	23	23	1.0531e−12	23	23	8.3333e−13
EIGENBLS	*	110	134	107	4.2412e−15	69	63	3.1853e−17	164	142	3.7937e−13	167	153	1.3427e−12
ENGVAL1		100	9	10	1.0909e+02	9	10	1.0909e+02	11	12	1.0909e+02	11	12	1.0909e+02
ENGVAL2	*	3	13	14	9.7152e−17	13	14	9.7152e−17	24	24	5.2007e−15	24	24	1.1952e−15
ERRINROS	*	50	56	48	3.9904e+01	52	47	3.9904e+01	85	79	3.9904e+01	75	72	3.9904e+01
EXPFIT	*	2	7	6	2.4051e−01	7	6	2.4051e−01	13	12	2.4051e−01	16	14	2.4051e−01
EXTROSNB	*	100	1281	1182	1.8373e−08	487	468	3.1722e−07	566	516	1.5784e−06	643	624	7.1530e−07
FLETCBV2		100	2	3	-5.1401e−01	2	3	-5.1401e−01	3	4	-5.1401e−01	3	4	-5.1401e−01
FLETCBV3		50	>	>	-3.5073e+02	>	>	-3.3920e+02	30878	30541	-1.3860e+03	>	>	-1.0286e+03
FLETCHBV		10	460	453	-2.1502e+06	1203	1151	-2.0203e+06	127	118	-2.3674e+06	257	257	-2.1109e+06
FLETCHCR		100	231	200	1.7096e−19	164	162	2.6432e−19	347	264	1.2049e−14	194	180	7.8105e−18
FMINSRF2		121	35	31	1.0000e+00	30	25	1.0000e+00	95	91	1.0000e+00	70	60	1.0000e+00
FMINSURF		121	32	27	1.0000e+00	23	19	1.0000e+00	102	98	1.0000e+00	70	59	1.0000e+00
FREUROTH	*	100	9	10	1.1965e+04	9	10	1.1965e+04	14	15	1.1965e+04	14	15	1.1965e+04
GENHUMPS	*	10	10402	9802	3.7851e−12	11624	10931	4.3255e−13	5083	4434	6.3997e−13	7075	6449	2.7198e−14
GENROSE	*	100	107	88	1.0000e+00	90	83	1.0000e+00	130	116	1.0000e+00	123	113	1.0000e+00
GENROSEB		500	460	369	1.0000e+00	327	325	1.0000e+00	585	505	1.0000e+00	498	473	1.0000e+00
GROWTHLS	*	3	96	78	1.0040e+00	79	72	1.0040e+00	183	172	1.0040e+00	171	163	1.0040e+00
GULF	*	3	30	28	1.7991e−17	32	30	3.6188e−14	40	38	3.4547e−13	44	43	3.2415e−09
HAIRY		2	64	57	2.0000e+01	116	107	2.0000e+01	96	84	2.0000e+01	91	86	2.0000e+01
HATFLDD	*	3	20	20	6.6151e−08	20	20	6.6151e−08	18	18	6.6937e−08	18	18	6.6937e−08
HATFLDE	*	3	21	21	5.1204e−07	20	20	5.1204e−07	17	17	5.1204e−07	17	17	5.1204e−07
HEART6LS	*	6	667	642	4.4113e−26	1039	1019	2.1192e−24	1528	1498	7.2910e−13	1593	1583	1.5966e−12
HEART8LS	*	8	112	95	4.6362e−17	102	88	1.7507e−13	152	143	2.0524e−20	159	154	3.8145e−14
HELIX	*	3	11	11	5.6587e−23	8	8	4.9599e−13	20	19	7.7395e−15	15	14	1.8475e−15
HIELOW		3	11	10	8.7417e+02	8	8	8.7417e+02	13	12	8.7417e+02	12	11	8.7417e+02
HILBERTA		2	3	4	2.0543e−33	3	4	2.0543e−33	3	4	1.8551e−30	3	4	1.8551e−30
HILBERTB		10	3	4	1.8835e−29	3	4	1.8835e−29	7	8	2.2225e−14	7	8	2.2225e−14
HIMMELBB		2	10	9	5.1740e−16	10	8	1.2423e−20	19	19	1.7548e−11	19	19	1.7548e−11

			Moré-Sorensen						Steihaug-Toint					
			BTR			RTR			BTR			RTR		
Name	LS	n	iter	#g	f	iter	#g	f	iter	#g	f	iter	#g	f
HIMMELBF	*	4	276	274	3.1857e+02	94	92	3.1857e+02	358	356	3.1857e+02	353	315	3.1857e+02
HIMMELBG		2	5	6	9.0327e−12	5	6	9.0327e−12	7	7	1.7308e−15	7	7	1.7308e−15
HIMMELBH		2	4	5	-1.0000e+00	4	5	-1.0000e+00	4	5	-1.0000e+00	4	5	-1.0000e+00
HUMPS	*	2	2690	2503	1.0977e−12	6856	6604	2.4027e−13	2606	2243	6.0915e−14	6265	6038	6.5371e−11
JENSMP		2	9	10	1.2436e+02	9	10	1.2436e+02	9	10	1.2436e+02	9	10	1.2436e+02
KOWOSB	*	4	11	10	3.0780e−04	11	10	3.0780e−04	12	12	3.0780e−04	12	11	3.0780e−04
LIARWHD	*	100	12	13	5.5677e−14	12	13	5.5677e−14	14	15	2.4677e−15	14	15	2.4677e−15
LOGHAIRY		2	2734	2676	1.8232e−01	9091	8167	1.8232e−01	4871	4132	5.1277e+00	7612	6953	1.8232e−01
MANCINO	*	100	14	15	1.5058e−21	16	16	4.0607e−19	20	21	1.4487e−21	20	21	1.4487e−21
MARATOSB		2	699	673	-1.0000e+00	680	667	-1.0000e+00	1882	1726	-1.0000e+00	1547	1493	-1.0000e+00
MEXHAT		2	32	30	-4.0010e−02	31	30	-4.0010e−02	19	20	-4.0010e−02	19	20	-4.0010e−02
MEYER3	*	3	481	441	8.7946e+01	416	381	8.7946e+01	686	680	8.8511e+01	693	688	8.8186e+01
MODBEALE		200	10	11	7.8240e−21	10	11	7.8240e−21	14	15	3.1114e−15	14	15	3.1114e−15
MOREBV	*	100	1	2	7.8870e−10	1	2	7.8870e−10	138	139	2.1401e−07	138	139	2.1401e−07
MSQRTALS	*	100	20	18	2.6765e−17	19	17	7.4695e−10	20	19	4.0318e−11	20	19	4.0318e−11
MSQRTBLS	*	100	16	14	1.8855e−17	16	14	9.4179e−14	21	20	4.1329e−14	21	20	4.1329e−14
NONCVXU2		100	53	47	2.3183e+02	49	41	2.3241e+02	45	40	2.3241e+02	41	34	2.3241e+02
NONCVXUN		100	42	38	2.3168e+02	41	36	2.3285e+02	44	40	2.3168e+02	41	34	2.3227e+02
NONDIA	*	100	6	7	1.4948e−18	6	7	1.4948e−18	10	11	6.5982e−15	10	11	6.5982e−15
NONDQUAR		100	15	16	2.6991e−09	15	16	2.6991e−09	110	84	2.1978e−06	97	86	1.9731e−06
OSBORNEA	*	5	37	32	5.4649e−05	30	27	5.4649e−05	64	59	5.4718e−05	82	79	5.4649e−05
OSBORNEB	*	11	21	19	4.0138e−02	21	19	4.0138e−02	22	22	4.0138e−02	22	22	4.0138e−02
OSCIPATH		8	2035	1734	1.7473e−05	2015	1804	1.4813e−05	3020	2625	3.3662e−05	2670	2488	4.3935e−05
PALMER1C		8	7	8	9.7605e−02	7	8	9.7605e−02	>	>	9.7653e−02	>	>	9.7653e−02
PALMER1D		7	7	8	6.5267e−01	7	8	6.5267e−01	23	24	6.5267e−01	23	24	6.5267e−01
PALMER2C		8	6	7	1.4369e−02	6	7	1.4369e−02	3161	3162	1.4370e−02	3161	3162	1.4370e−02
PALMER3C		8	6	7	1.9538e−02	6	7	1.9538e−02	1784	1785	1.9539e−02	1784	1785	1.9539e−02
PALMER4C		8	7	8	5.0311e−02	7	8	5.0311e−02	1538	1539	5.0312e−02	1538	1539	5.0312e−02
PALMER5C	*	6	5	6	2.1281e+00	5	6	2.1281e+00	9	10	2.1281e+00	9	10	2.1281e+00
PALMER6C	*	8	7	8	1.6387e−02	7	8	1.6387e−02	165	166	1.6388e−02	165	166	1.6388e−02
PALMER7C	*	8	9	10	6.0199e−01	9	10	6.0199e−01	6810	5734	6.0199e−01	4456	3946	6.0199e−01
PALMER8C	*	8	8	9	1.5977e−01	8	9	1.5977e−01	197	198	1.5977e−01	197	198	1.5977e−01
PENALTY1	*	100	45	44	9.0249e−04	45	44	9.0249e−04	44	41	9.0260e−04	48	44	9.0249e−04
PENALTY2	*	100	19	20	9.7096e+04	19	20	9.7096e+04	19	20	9.7096e+04	19	20	9.7096e+04
PFIT1LS	*	3	325	287	1.5734e−16	294	280	3.0857e−15	365	350	4.8505e−07	384	379	4.3509e−07
PFIT2LS	*	3	114	98	3.6218e−15	90	84	3.4229e−20	133	128	1.9620e−08	161	158	7.5351e−09
PFIT3LS	*	3	144	125	4.4639e−19	126	116	3.6432e−14	222	211	1.2519e−08	226	221	2.4788e−09
PFIT4LS	*	3	241	218	3.4144e−20	232	223	8.8142e−23	401	390	6.1391e−10	495	491	7.1420e−10
POWELLSG		4	15	16	4.6333e−09	15	16	4.6333e−09	15	16	1.2731e−08	15	16	1.2731e−08

			Moré-Sorensen						Steihaug-Toint					
			BTR			RTR			BTR			RTR		
Name	LS	n	iter	#g	f	iter	#g	f	iter	#g	f	iter	#g	f
POWER		100	24	25	1.1818e−09	24	25	1.1818e−09	25	26	1.6694e−09	25	26	1.6694e−09
QUARTC		100	29	30	2.8059e−08	29	30	2.8059e−08	29	30	3.5899e−08	29	30	3.5899e−08
ROSENBR	*	2	30	26	7.1488e−15	28	26	6.0210e−13	34	30	2.8234e−14	34	31	5.7977e−11
S308	*	2	13	12	7.7320e−01	13	12	7.7320e−01	9	10	7.7320e−01	9	10	7.7320e−01
SBRYBND	*	100	46	37	2.5620e−22	46	37	9.1262e−15	>	>	2.6525e+01	>	>	2.5463e+01
SCHMVETT		100	4	5	-2.9400e+02	4	5	-2.9400e+02	6	7	-2.9400e+02	6	7	-2.9400e+02
SCOSINE		100	>	>	-9.8840e+01	97	90	-9.9000e+01	>	>	-9.7311e+01	>	>	-9.3382e+01
SCURLY10	*	100	39	35	-1.0032e+04	46	42	-1.0032e+04	>	>	-1.0013e+04	>	>	-1.0013e+04
SCURLY20	*	100	34	30	-1.0032e+04	37	33	-1.0032e+04	>	>	-1.0032e+04	>	>	-1.0032e+04
SCURLY30	*	100	35	31	-1.0032e+04	35	31	-1.0032e+04	>	>	-1.0022e+04	>	>	-1.0021e+04
SENSORS	*	100	21	21	-1.9668e+03	24	23	-1.9668e+03	20	20	-2.0250e+03	24	22	-2.0250e+03
SINEVAL	*	2	53	46	1.9744e−25	58	52	3.3812e−36	107	93	3.6189e−18	80	73	1.4447e−21
SINQUAD		100	9	10	-4.0056e+03	9	10	-4.0056e+03	14	14	-4.0056e+03	11	12	-4.0056e+03
SISSER		2	12	13	1.0658e−08	12	13	1.0658e−08	12	13	1.2144e−08	12	13	1.2144e−08
SNAIL		2	61	61	9.3702e−13	59	60	1.2117e−14	72	72	8.6160e−17	62	63	3.6402e−18
SPARSINE		100	37	27	9.3794e−16	30	22	2.8734e−16	10	11	1.7155e−15	10	11	1.7155e−15
SPARSQUR		100	16	17	1.4795e−08	16	17	1.4795e−08	16	17	1.9872e−08	16	17	1.9872e−08
SPMSRTLS	*	100	14	13	1.2592e−13	12	11	6.1356e−12	13	13	4.6661e−14	13	13	4.6661e−14
SROSENBR	*	100	6	7	8.8993e−28	6	7	8.8993e−28	8	9	2.6078e−19	8	9	2.6078e−19
TOINTGOR		50	9	10	1.3739e+03	9	10	1.3739e+03	11	12	1.3739e+03	11	12	1.3739e+03
TOINTGSS		100	17	15	1.0102e+01	13	13	1.0204e+01	12	12	1.0102e+01	12	12	1.0102e+01
TOINTPSP		50	22	20	2.2556e+02	30	28	2.2556e+02	47	38	2.2556e+02	58	50	2.2556e+02
TQUARTIC	*	100	14	13	2.6771e−24	15	13	1.4965e−17	15	15	5.3087e−15	15	15	5.3087e−15
VARDIM		200	29	30	2.9081e−24	29	30	2.9081e−24	29	30	2.0682e−25	29	30	2.0682e−25
VAREIGVL	*	50	15	13	4.7122e−09	16	14	1.3553e−10	13	14	2.2712e−10	13	14	2.2712e−10
VIBRBEAM	*	8	49	39	1.7489e+00	51	40	1.7489e+00	668	669	1.5645e−01	960	956	1.5645e−01
WATSON	*	12	14	14	8.1544e−07	13	13	3.9067e−08	12	13	1.5973e−07	12	13	1.5973e−07
WOODS	*	4	52	44	4.6408e−15	53	47	5.1563e−17	69	59	2.0670e−13	60	54	3.8275e−17
YFITU	*	3	54	48	6.6863e−13	50	46	6.6700e−13	85	77	2.2960e−08	79	75	1.0173e−08

Appendix F

Traduction française des parties-clés de la thèse

F.1 Introduction

L'optimisation nonlinéaire est une discipline des mathématiques appliquées dont le but est d'optimiser des fonctions nonlinéaires. En pratique, on cherche le minimum d'une fonction de coût $f(.)$, appelée *fonction objectif*, qui peut être soumise à des contraintes. Une façon traditionnelle d'écrire un problème d'optimisation est la suivante

$$\min_{x \in \mathcal{F}} f(x), \tag{F.1}$$

où $f(.)$ est une fonction continue qui peut être nonlinéaire et où $\mathcal{F}$ désigne l'ensemble admissible. Ce problème admet en général une solution globale, mais aussi parfois des solutions locales, c'est-à-dire des points qui minimisent la fonction objectif au moins sur l'intersection entre le domaine admissible et une (potentiellement petite) boule ouverte. Dans ce travail, nous nous concentrons sur la recherche de solutions locales de (F.1). De plus, nous nous intéressons au cas où $\mathcal{F}$ est un ensemble de contraintes de bornes, c'est-à-dire

$$\mathcal{F} = \{x \in \mathbb{R}^n \,|\, l \leq x \leq u\},$$

avec $l, u \in \mathbb{R}^n$ et où les inégalités sont interprétées composante par composante. Dans ce cas, les conditions suivantes sont suffisantes pour assurer qu'un vecteur x^* est une solution locale exacte du problème (F.1) :

$$\begin{array}{c} [\nabla_x f(x_*)]_i = 0 \text{ pour tout } i \notin \mathcal{A}(x_*), \\ \nabla_{xx} f(x_*) \text{ définie positive,} \end{array} \tag{F.2}$$

où $\nabla_x f(.)$ est le gradient de $f(.)$, où $\nabla_{xx} f(.)$ est sa matrice hessienne et où

$$\mathcal{A}(\tilde{x}) = \left\{ i \in \{1, ..., n\} \;|\; \begin{array}{rclcrcl} [\tilde{x}]_i & = & [l]_i & \text{et} & [\nabla_x f(\tilde{x})]_i & > & 0 \\ & & & \text{ou} & & & \\ [\tilde{x}]_i & = & [u]_i & \text{et} & [\nabla_x f(\tilde{x})]_i & > & 0 \end{array} \right\}$$

est l'ensemble des *contraintes actives liées* en $\tilde{x}$. En pratique, on cherche seulement des *points critiques du premier ordre* de (F.1), c'est-à-dire des points qui satisfont seulement la première ligne de (F.2). Pour résoudre ce problème, on utilise généralement des *méthodes itératives*. Ces algorithmes produisent une séquence de points,

nommés itérés, en partant d'un premier candidat donné x_0 et jusqu'à ce que la solution approchée soit suffisamment proche de la solution critique du premier ordre. Les algorithmes itératifs sont arrêtés lorsque (F.2) est suffisamment proche d'être satisfait et, par exemple, dans le cas où $\mathcal{F} = \mathbb{R}^n$, le critère d'arrêt est simplement

$$||\nabla_x f(x_*)|| \leq \epsilon,$$

où ϵ est une constante donnée. Nous recommandons Kelley (1999) pour une discussion au sujet d'autres définitions possibles pour les critères d'arrêts. Ce critère d'arrêt doit être adapté dans le cadre de contraintes de bornes, comme nous le verrons plus loin. Deux classes principales de méthodes itératives sont généralement utilisées pour résoudre les problèmes d'optimisation nonlinéaire avec et sans contraintes de bornes (voir Nocedal and Wright (1999)) : les méthodes de recherche linéaire (voir Zhu et al. (1997) et Hager and Zhang (2004) parmi beaucoup d'autres) et les méthodes de région de confiance (Conn et al. (1996) ou Gould et al. (2002), par exemple).

A chaque itération, les méthodes de recherche linéaire sélectionnent une *direction de descente*, définie comme une direction le long de laquelle la fonction de coût peut être diminuée. Un pas est alors calculé le long de cette direction, partant de l'itéré courant et dont la longueur est choisie de sorte que le pas génère une décroissance dans la fonction objectif. Cette longueur de pas peut être choisie comme le minimiseur exact de $f(.)$ le long de la direction de descente choisie. Dans ce cas, la méthode est appelée *recherche linéaire exacte*, mais cette technique n'est pas toujours très efficace en pratique. Nous pouvons aussi avoir recours à une *recherche linéaire inexacte* et appliquer les célèbres conditions d'Armijo et de Goldstein pour définir un pas raisonable (voir, par exemple, Dennis and Schnabel (1983) ou Moré and Thuente (1994)). Même si ces méthodes fonctionnent relativement bien, dans le cadre de cette thèse nous nous sommes focalisés sur les méthodes de région de confiance, moins sensibles au mauvais conditionnement de la matrice hessienne et à la nonconvexité du problème. Si le lecteur souhaite plus d'informations au sujet des différentes méthodes pour la résolution de problèmes d'optimisation nonlinéaires, nous recommandons l'excellente introduction de Gould and Leyffer (2003) sur ce sujet.

Les méthodes de région de confiance sont parmi les méthodes les plus populaires et les plus efficaces pour l'optimisation nonlinéaire et sont supportées par une théorie considérable (voir Conn et al. (2000) pour une couverture plus étendue du sujet). Ces méthodes procèdent de manière itérative en minimisant un modèle de la fonction objectif dans une région, définie dans une norme spécifique, où le modèle est présumé fiable. Elles insistent sur le fait que chaque pas doit réaliser une décroissance minimale, connue sous le nom de condition de Cauchy. Elles adaptent le rayon de la région de confiance et choisissent d'accepter ou de refuser le pas en accord avec la décroissance relative de la fonction objectif par rapport à la décroissance du modèle. Cependant, telles quelles, ces méthodes n'exploitent pas la structure des problèmes. Notre objectif est donc d'explorer différentes alternatives pour exploiter cette structure dans les fréquentes situations où le problème peut être à la base décomposé en une hiérarchie de modèles avec des degrés d'approximation variés. Le récent intérêt pour la conception de surfaces, l'assimilation de données pour la prévision météorologique (Fisher 1998) ou le contrôle optimal de systèmes décrits par des équations différentielles partielles a été la motivation principale de cette nouvelle tendance en recherche. En outre, d'autres applications telles que la graduation multidimensionnelle (multi-dimensional scaling) (Bronstein et al. 2005) ou les shémas de

quantification (Emilianenko 2005) ont aussi suscité des questions similaires. Dans de tels problèmes, on considère typiquement une (fine) discrétisation d'un problème de dimension infinie qui fournit une approximation suffisamment bonne de la solution. Cependant, nous avons souvent aussi accès à des discrétisations plus grossières qui décrivent encore le problème raisonablement bien, et qui peuvent donc être utilisés pour améliorer l'efficacité de la solution numérique sur une discrétisation fine.

Dans le cadre de la résolution numérique de systèmes linéaires provenant d'équations différentielles partielles, des techniques ont été développées sous le nom de *méthodes multigrilles* qui exploitent le cas où la hiérarchie de problèmes provient d'une discrétisation multiniveaux d'un problème continu sous-jacent. Ce domaine actif de recherche, défriché par Brandt (1977), est basé sur une double observation : d'une part il existe des méthodes de résolution itératives (appelées *lisseurs*) très efficaces pour réduire les composants oscillants, à hautes fréquences, de l'erreur mais qui sont potentiellement inefficaces pour réduire ses composants lisses, à basses fréquences (les méthodes de Jacobi et Gauss-Seidel en sont des exemples représentatifs). D'autre part, la définition des composants à hautes fréquences est intrinsèquement liée à la grille de discrétisation puisque plus la grille est fine, plus la fréquence représentable sur cette grille est élevée. Les méthodes multigrilles procèdent alors en utilisant des lisseurs pour réduire les composants oscillants de l'erreur sur une grille fine, et considèrent les composants lisses restants de cette grille fine comme des composants oscillants sur une grille plus grossière. De façon générale, ces derniers peuvent être éliminés en utilisant des lisseurs sur la grille plus grossière. De plus, cette technique peut être appliquée récursivement afin d'éliminer l'ensemble des différentes fréquences de l'erreur. Un des principaux attraits des méthodes multigrilles pour systèmes linéaires bien réglées est que leur quantité de travail croît seulement linéairement avec la taille du problème, caractéristique cruciale pour la résolution de problèmes avec un très grand nombre de variables. Nous conseillons au lecteur les excellents livres de Briggs et al. (2000) et Trottenberg et al. (2001) pour une couverture significative de cette classe d'algorithmes remarquablement efficaces.

L'idée d'exploiter la structure hiérarchique des problèmes en optimisation est beaucoup plus récente. Certains auteurs ont proposé des méthodes qui tiennent compte des hiérarchies multiniveaux comme Fisher (1998), Nash (2000), Lewis and Nash (2002, 2005), et Oh et al. (2005). Kornhuber (1994, 1996, 1997) a aussi développé une méthode de ce type dans le contexte des éléments finis pour les problèmes non-lisses, convexes et soumis à des contraintes de bornes. La convergence de ces méthodes multigrilles est assurée par la minimisation successive suivant les directions des coordonnées générées par les lisseurs de type Gauss-Seidel, évitant de ce fait le besoin d'une globalisation explicite. D'un autre côté, Gratton et al. (2008*b*) ont proposé un algorithme récursif de région de confiance en norme Euclidienne pour les problèmes multiniveaux généraux de minimisation non-convexe sans contraintes. L'intérêt principal de leur proposition est de fournir le premier cadre globalement convergent pour l'application de mécanismes de type multigrilles géométriques à cette classe de problèmes. De plus, les expérimentations numériques initiales réalisées avec cet algorithme sont très prometteuses (voir Gratton et al. (2006*a*)) et motivent une analyse plus approfondie des méthodes de ce type.

Bien qu'il soit théoriquement satisfaisant et acceptable en pratique, le choix de la norme Euclidienne pour la définition de la région de confiance n'est pas sans incon-

vénients. Premièrement, et essentiellement pour ce travail, les régions de confiance Euclidiennes ne conviennent pas au traitement des problèmes avec contraintes de bornes parce que l'intersection de la région de confiance (une boule en norme Euclidienne) avec le domaine admissible pour les bornes (une boite) a une structure plus complexe que, par exemple, une simple boite. De plus, la combinaison des itérations de lissage de type Gauss-Seidel avec une région de confiance Euclidienne n'est pas naturelle car les pas de lissage considèrent les variables une à une et sont, de ce fait, alignés avec les axes de coordonnées. En outre, des complications plus techniques proviennent du fait que, dans la proposition de Gratton et al. (2008*b*), le pas réalisé à un niveau grossier doit en même temps être inclus dans la région de confiance du niveau courant et être tel que sa prolongation au(x) niveau(x) plus fin(s) soit incluе dans la(les) région(s) de confiance du(des) niveau(x) plus fin(s). Comme discuté dans Gratton et al. (2008*b*), cette double requête implique l'utilisation de préconditionneurs coûteux ainsi qu'une technique spéciale pour mettre à jour le rayon de la région de confiance, ce qui limite parfois inefficacement la longueur du pas.

Pour s'adapter aux contraintes de bornes et éviter ces difficultés techniques, un algorithme multiniveaux alternatif peut être défini en utilisant la norme infinie pour la définition de la région de confiance. Le premier but de cette thèse est de décrire cet algorithme pour l'optimisation avec contraintes de bornes, ce qui est fait en début de chapitre 2. En sus, l'algorithme ne nécessite aucun préconditionneur imposé et est beaucoup moins restrictif pour les pas des niveaux grossiers que son prédécesseur dans le cas sans contraintes. De plus, les itérations de lissage qui explorent les directions alignées avec les axes de coordonnées sont bien adaptés à la forme de boite de l'ensemble admissible déterminé par l'intersection entre la région de confiance et l'ensemble $\mathcal{F}$ des contraintes.

Malheureusement, la théorie de convergence présentée dans Gratton et al. (2008*b*, 2006*b*) ne peut pas être appliquée à ce nouvel algorithme sans modification significative, non seulement à cause de la possible présence de bornes, mais aussi parce que l'algorithme analysé dans ces références est lui-même très dépendant du choix de la norme Euclidienne. Le second objectif de la thèse, réalisé dans la seconde partie du Chapitre 2, est donc de prouver la convergence globale du nouvel algorithme vers des points critiques du premier ordre, c'est-à-dire la convergence depuis des points de départ arbitraires vers des points limites satisfaisant les conditions d'optimalité du premier ordre.

Comme prévu, l'algorithme et la théorie présentés ici s'appliquent aussi, avec de petites adaptations, aux ensembles d'équations nonlinéaires. En effet, une des techniques les plus courantes dans ce domaine est de considérer la minimisation d'une norme continue des résidus, ce qui peut alors être vu comme un problème de minimisation sans contraintes, dont la solution donne les racines désirées lorsque les résidus convergent vers zéro. En conséquence, l'algorithme multiniveaux proposé s'applique aussi à la solution multiniveaux d'équations nonlinéaires, tout comme la preuve de convergence globale associée.

Nous nous intéressons aussi à l'identification des contraintes actives par l'algorithme, c'est-à-dire à déterminer quelles contraintes d'inégalité sont en fait des égalités à la solution exacte. Dans le cas des problèmes soumis à des contraintes convexes résolus par un algorithme de région de confiance dont le solveur interne est basé sur le pas de Cauchy généralisé, Conn et al. (1993) ont prouvé que l'identification des con-

traintes actives est réalisée au bout d'un nombre fini d'itérations. En conséquence, dans la dernière partie du Chapitre 2, nous montrons que la théorie d'identification des contraintes actives présentée dans cette référence peut être en fait étendue sans trop de difficultés à toute méthode de région de confiance dont le solveur interne respecte une condition de décroissance suffisante, de même qu'à l'usage de la norme infinie dans la définition de la région de confiance. Dans notre contexte, ce résultat implique que si des itérations de type Gauss-Seidel sont utilisées pour calculer les pas dans un algorithme de région de confiance pour l'optimisation avec bornes, et s'il est démontré que ce solveur interne respecte la condition de décroissance suffisante requise, alors l'algorithme identifie l'ensemble actif à la solution exacte après un nombre fini d'itérations.

De plus, cette propriété peut aussi s'appliquer à une variante de l'algorithme de région de confiance multiniveaux, lorsqu'il est autorisé à exploiter la structure multiniveaux à chaque pas, mais uniquement sur les variables pour lesquelles aucune contrainte n'est active. Néanmoins, ce résultat théorique n'a pas d'effet positif significatif sur les résultats numériques.

Etant donné que nous travaillons dans un contexte de méthodes itératives pour la résolution de problèmes soumis à des contraintes de bornes, nous nous sommes aussi intéressés à la conception d'un critère d'arrêt ayant un sens particulier pour notre algorithme. Beaucoup d'algorithmes avec contraintes de bornes définissent leur critère d'arrêt comme la norme de la projection de l'opposé du gradient sur les contraintes (voir, par exemple, Zhu et al. (1994), Lin and Moré (1999), Hager and Zhang (2006) et Xu and Burke (2007)). Cependant, un autre critère est plus souvent utilisé parmi les utilisateurs de méthodes de région de confiance, qui a été introduit pour la première fois par Conn et al. (1993), et qui a pour propriété d'être une approximation du premier ordre de la décroissance maximale qui peut être obtenue dans la direction opposée au gradient. Cette propriété est d'un intérêt tout particulier pour les méthodes de région de confiance à cause de l'importance qu'elles donnent à la décroissance réalisée à chaque pas.

Lorsqu'ils sont exactement égaux à zéro, ces critères d'arrêts sont équivalents aux conditions suffisantes d'optimalité du premier ordre. Cependant, il n'est pas approprié de les utiliser dans le cas où les données sont approximatives, ce qui est le cas lorsqu'on travaille sur des problèmes discrétisés. D'un autre côté, un critère d'arrêt adapté à la résolution de problèmes approchés à déjà été conçu dans le cadre linéaire. En conséquence, dans le Chapitre 3 nous avons décidé de suivre une approche d'*analyse de l'erreur inverse* et d'observer si cela mène à des critères d'arrêt déjà connus dans le cadre de l'optimisation nonlinéaire avec contraintes de bornes. Cette technique consiste à supposer que la solution approchée courante résout exactement un problème proche du problème original, et à mesurer la distance entre les deux problèmes plutôt que la distance entre les deux solutions. Cette technique est bien connue et a été intensivement étudiée dans le cadre de l'algèbre linéaire (voir Rigal and Gaches (1967), Cox and Higham (1998), Golub and Van Loan (1983), Chaitin-Chatelin and Fraysse (1996) ou Higham (1996)), mais c'est la première fois que l'analyse de l'erreur inverse est utilisée pour concevoir un critère d'arrêt pour l'optimisation nonlinéaire avec contraintes de bornes. S'il est décidé d'arrêter l'algorithme lorsque l'erreur inverse est inférieure à un certain seuil, notre approche a pour avantage que ce seuil peut être déterminé comme une fonction des incertitudes que l'on connait sur le gra-

dient, la fonction objectif ou les contraintes de borne. Il est en effet inutile d'essayer de réduire la distance entre le problème original et celui qui lui est proche lorsqu'elle est inférieure à ces incertitudes. A la fin du Chapitre 3, nous vérifierons que le critère d'arrêt défini par cette analyse de l'erreur inverse respecte toutes les propriétés nécessaires à la convergence de notre algorithme multiniveaux de région de confiance.

Pour terminer, nous définissons dans le Chapitre 4 un algorithme concret où les solveurs internes sont spécifiés et certains choix d'options, laissés imprécisés dans l'algorithme théorique, sont décrits. Nous appliquons cette implémentation particulière de la méthode sur quelques problèmes tests représentatifs de grande taille sans contraintes ou soumis à des contraintes de bornes sur lesquels nous réalisons une première série de tests afin de déterminer des valeurs par défaut appropriées pour les paramètres de la méthode. Nous utilisons ensuite cette configuration optimale pour comparer notre algorithme avec d'autres méthodes utilisées dans ce domaine et illustrer l'efficacité des méthodes de région de confiance multiniveaux. Pour terminer, nous comparons le comportement numérique des différents critères d'arrêt, en particulier le critère habituel pour les méthodes de région de confiance et celui conçu par le biais de l'analyse de l'erreur inverse.

Des conclusions et perspectives sont discutées dans le Chapitre 5

F.2 Un algorithme de région de confiance multiniveaux en norme infinie pour l'optimisation avec contraintes de bornes

Dans ce chapitre, après avoir rappelé les concepts de base de l'optimisation nonlinéaire, nous introduisons les idées principales qui définissent le nouvel algorithme multiniveaux, nous prouvons sa convergence depuis tout point de départ arbitraire et nous étendons la théorie d'identification des contraintes actives pour les méthodes de région de confiance à l'usage de solveurs internes satisfaisant une condition de décroissance suffisante.

F.2.1 Concepts préliminaires

Nous considérons le problème d'optimisation suivant

$$\min f(x), \tag{F.3}$$

où f est une fonction objectif deux fois continuement différentiable de $\mathbb{R}^n$ dans $\mathbb{R}$ et qui est bornée infrieurement. Nous sommes intéressés par trouver une solution critique du premier ordre x_* de (F.3) dans le sens $[\nabla_x f(x_*)]_j = 0$ pour tout j, où $[v]_j$ représente la $j^{\text{ième}}$ composante d'un vecteur v. Un moyen très classique de résoudre ce problème est d'appliquer la méthode de Newton. C'est une méthode itérative dans le sens où, étant donné un point initial x_0, elle produit une séquence d'itérés $\{x_k\}$. Lors d'une itération k, partant de x_k, la méthode approxime la fonction objectif $f(x)$ autour de x_k par son approximation de Taylor du second ordre. Chaque pas produit par la méthode de Newton est le résultat de la minimisation de ce modèle de Taylor

$$s_k = \min_s f(x_k) + \nabla_x f(x_k)^T s + \frac{1}{2} s^T \nabla_x^2 f(x_k) s,$$

où $\nabla_x f(.)$ est le gradient de $f(.)$, $\nabla_x^2 f(.)$ est sa matrice hessienne et v^T dénote la transposée d'un vecteur v. Cette expression est équivalente à $\nabla_x f(x_k)^T + \nabla_x^2 f(x_k) s_k = 0$ et $\nabla_x^2 f(x_k) > 0$. En conséquence, l'itéré suivant est donné par

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

L'algorithme est arrêté dès que le gradient est suffisamment proche de zéro dans le sens de $||\nabla_x f(x_k)|| < \epsilon$, où le gradient est mesuré dans une norme appropriée et où ϵ est la tolérance choisie. La méthode de Newton est localement quadratiquement convergente sous certaines conditions de régularité de $f(.)$ à la solution x^*. En d'autres mots, cette méthode peut être très rapide près de la solution mais peut ne pas trouver de solution si la minimisation est commencée trop loin de celle-ci. Pour palier ce problème, nous pouvons nous intéresser aux méthodes de région de confiance. Ce sont des méthodes très connues et très efficaces pour résoudre des problèmes d'optimisation nonlinéaire et ce pour deux raisons principales. Premièrement, elles sont *globalement convergentes*, ce qui signifie qu'elles trouvent un point critique du premier ordre à partir de tout point de départ admissible x_0. Leur second avantage est qu'elles se réduisent à la méthode de Newton lorsqu'elles sont suffisamment proches de la solution et, en conséquence, ont une *convergence quadratique*. Regardons à présent de manière plus approfondie le fonctionnement de l'algorithme de région de confiance basique.

A chaque itération k, l'algorithme construit un modèle m_k de la fonction objectif autour de l'itéré courant x_k, qui est généralement une approximation quadratique de $f(x)$. Il définit aussi une *région de confiance* $\mathcal{B}_k$ centrée en x_k et définie par son rayon $\Delta_k > 0$, dans laquelle le modèle est supposé fiable. Un pas s_k est alors calculé à l'intérieur de la région de confiance, qui induit une réduction suffisante du modèle. La fonction objectif est calculée au *point d'essai* (trial point) $x_k + s_k$ et ce candidat est accepté comme itéré suivant si et seulement si ρ_k, le ratio de la réduction effective de la fonction objectif sur la réduction prédite par son modèle est raisonnable (typiquement plus grande qu'une constante positive $\eta_1 \leq 1$). Le rayon de la région de confiance est finalement mis à jour : il est réduit si le point d'essai est rejeté et laissé inchangé ou augmenté si ρ_k est suffisamment grand. L'algorithme est arrêté dès que la norme du gradient est inférieure à une constante choisie, c'est-à-dire $||\nabla_x f(x_k)|| < \epsilon$. L'introduction de la région de confiance $\mathcal{B}_k$ assure le caractère globalement convergent de l'algorithme, tandis que la définition du modèle implique que m_k devient similaire à la fonction objectif lorsqu'on approche la solution et, en conséquence, la rayon de la région de confiance finit par tendre vers l'infini, de sorte que la méthode de région de confiance se réduit à la méthode de Newton. Nous renvoyons le lecteur à Conn et al. (2000) pour une couverture détaillée de ce sujet.

Nous considérons maintenant le problème d'optimisation avec contraintes

$$\min_{x \in \mathcal{F}} f(x), \tag{F.4}$$

où $\mathcal{F} = \{x \in \mathbb{R}^n | l \leq x \leq u\}$ est un ensemble de contraintes de bornes et où $l, u \in \mathbb{R}^n$ peuvent être infinis. Dans ce cas, chercher un point critique du premier ordre x_* de (F.4) revient à trouver x_* tel que

$$[\nabla_x f(x_*)]_j = 0 \quad \text{for all } j \notin \mathcal{A}(x_*), \tag{F.5}$$

où $\mathcal{A}(x) = \mathcal{A}^-(x) \cup \mathcal{A}^+(x)$ est l'ensemble des *contraintes actives liées* avec

$$\begin{array}{rcl} \mathcal{A}^-(x) = \{j \in \{1, ..., n\} \mid [x]_j & = & [l]_j \quad \text{and} \quad [\nabla_x f(x)]_j > 0\} \\ \mathcal{A}^+(x) = \{j \in \{1, ..., n\} \mid [x]_j & = & [u]_j \quad \text{and} \quad [\nabla_x f(x)]_j < 0\}. \end{array}$$

Dans ce contexte, l'algorithme de région de confiance sans contraintes peut être facilement adapté pour devenir l'algorithme F.2.1. Cependant, quelques commentaires sont nécessaires. Nous définissons premièrement une *mesure de criticalité* $\chi_k = \chi(x_k)$ qui doit être égale à zéro lorsqu'elle est évaluée à la solution exacte x_* et qui est utilisée comme un critère d'arrêt conçu pour l'optimisation avec contraintes de bornes. Les mesures de criticalité habituelles sont, par exemple, $\chi_k^{out,2} = \|\text{Proj}_{\mathcal{F}}(x_k - \nabla_x f(x_k)) - x_k\|_2$ où $\text{Proj}_{\mathcal{F}}$ est la projection orthogonale sur la boite $\mathcal{F}$, ou $\chi_k^{tr} = |\min_{x_k + d \in \mathcal{F}, \|d\|_\infty \leq 1} \langle \nabla_x f(x_k), d \rangle|$ (voir, par exemple, Conn et al. (2000)). Le choix de la définition la plus adaptée n'est pas évident et sera discuté en détails au Chapitre 3. Un second point à préciser est que nous avons choisi de définir la région de confiance en norme infinie $\mathcal{B} = \{s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_k\}$ pour rendre plus facile son intersection avec l'ensemble des contraintes de bornes du problème original. Finalement, le modèle choisi est

$$m_k(x_k + s_k) = f(x_k) + g_k^T s_k + \frac{1}{2} s_k^T H_k s_k, \tag{F.6}$$

où $g_k = \nabla_x f(x_k)$ et où H_k est une approximation symétrique $n \times n$ de $\nabla_x^2 f(x_k)$, et la condition de décroissance suffisante, connue sous le nom de condition de Cauchy modifiée, est donnée par

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{red} \chi_k \min \left[\frac{\chi_k}{\beta_k}, \Delta_k, 1 \right], \tag{F.7}$$

où $\kappa_{dcp} \in (0, \frac{1}{2})$ et où $\beta_k = 1 + \|H_k\|$. Malgré son caractère apparemment technique, cette exigence n'est pas démesurément restrictive et peut être garantie pour des algorithmes pratiques, comme décrits par exemple dans la Section 12.2.1 de Conn et al. (2000), ou dans les Sections 4.1.1 et B.1 de cette thèse.

Algorithm F.2.1: BTR(x_0, g_0, ϵ)

Etape 0: Initialisation. Calculer $f(x_0)$, définir $\mathcal{B}_0 = \{x_0 + s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_0\}$ et poser $k = 0$.

Etape 1: Calcul du pas. Calculer un pas $s_k \in \mathcal{B}_k$ qui réduit suffisamment le modèle m_k défini par (F.6) au sens de (F.7). Définir $\delta_k = m_k(x_k) - m_k(x_k + s_k)$.

Etape 2: Acceptation du point d'essai. Calculer $f(x_k + s_k)$ et $\rho_k = [f(x_k) - f(x_k + s_k)]/\delta_k$. Si $\rho_k \geq \eta_1$, alors définir $x_{k+1} = x_k + s_k$; sinon, définir $x_{k+1} = x_k$.

Etape 3: Test d'arrêt. Calculer g_{k+1} et χ_{k+1}. Si $\chi_{k+1} \leq \epsilon$, alors retourner la solution approchée $x_* = x_{k+1}$.

Etape 4: Mise-à-jour de la région de confiance. Poser

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, +\infty) & \text{si } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{si } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{si } \rho_k < \eta_1, \end{cases}$$

où $0 < \eta_1 < \eta_2 < 1$ et $0 < \gamma_1 < \gamma_2 < 1$. Définir $\mathcal{B}_{k+1} = \{x_{k+1} + s \in \mathbb{R}^n | \, ||s||_\infty \leq \Delta_{k+1}\}$, incrémenter k d'une unité et aller à l'étape 1.

F.2.2 Le problème et l'algorithme

Lorsque une hiérarchie de descriptions du problème (F.4) est connue, nous souhaitons exploiter ces différentes descriptions en suivant l'idée de Gratton et al. (2008*b*). Pour être plus précis, supposons qu'une collection de fonctions $\{f_i\}_{i=0}^r$ est disponible, chaque f_i étant une fonction deux fois continuement différentiable de $\mathbb{R}^{n_i}$ dans $\mathbb{R}$ (avec $n_i \geq n_{i-1}$). Nous supposons que $n_r = n$ et $f_r(x) = f(x)$ pour tout $x \in \mathbb{R}^n$, ce qui redonne le problème original. Nous faisons aussi l'hypothèse que f_i est "plus coûteux" à minimiser que f_{i-1} pour chaque $i = 1, \ldots, r$. Cela peut être le cas si les f_i représentent des discrétisations de plus en plus fines d'un même objectif en dimension

infinie. Pour fixer la terminologie, nous appellerons un i particulier un *niveau*. Nous utilisons le premier indice i dans tous les indices ultérieurs pour dénoter une quantité correspondant au $i^{\text{ème}}$ niveau, allant du plus grossier ($i = 0$) au plus fin ($i = r$) (ce qui signifie, en particulier, lorsqu'il est appliqué à un vecteur, que ce vecteur appartient à $\mathbb{R}^{n_i}$). Une relation doit évidemment exister entre les variables de deux fonctions successive de la collection $\{f_i\}_{i=0}^r$. Nous supposons donc que, pour tout $i = 1, \ldots, r$, il existe un opérateur linéaire de rang plein R_i de $\mathbb{R}^{n_i}$ dans $\mathbb{R}^{n_{i-1}}$ (la restriction) et un autre opérateur de rang plein P_i de $\mathbb{R}^{n_{i-1}}$ dans $\mathbb{R}^{n_i}$ (la prolongation) tels que

$$\sigma_i P_i = R_i^T,$$

pour une constante $\sigma_i > 0$ où P_i et R_i sont interprétés comme la restriction et la prolongation entre une grille fine et une grille grossière. Ces hypothèses sont communes à un certain nombre d'approches multiniveaux en optimisation (Fisher (1998), Nash (2000), Gratton et al. (2008*b*)) ou dans la résolution de systèmes nonlinéaires d'équations (voir Briggs et al. (2000) et les références qui y sont citées). Afin de simplifier les notations, et parce que c'est souvent le cas en pratique, nous supposons, sans perte de généralité, que $\|R_i\|_\infty = 1$ pour tout i (puisque nous pouvons choisir $\sigma_i = 1/\|P_i\|_\infty$).

Lorsque le problème a deux niveaux (r et $r-1$), l'idée principale est d'utiliser f_{r-1} comme un modèle pour $f_r = f$ dans le voisinage de l'itéré courant $x_{r,k}$, qui soit moins coûteux que le modèle quadratique de Taylor au niveau r. Nous minimisons alors le modèle f_{r-1} (potentiellement nonquadratique) *en utilisant un algorithme de région de confiance* au niveau $r-1$, dont l'itération ℓ possède sa propre région de confiance en forme de boite $\mathcal{B}_{r-1,\ell}$. Cette minimisation est réalisée dans un ensemble de contraintes héritées du niveau r et en partant du point initial $x_{r-1,0} = R_r x_{r,k}$, jusqu'à ce qu'un minimiseur contraint approximé $x_{r-1,*}$ soit trouvé. Le pas résultant est alors prolongé vers le niveau r en calculant

$$s_{r,k} = P_r(x_{r-1,*} - x_{r-1,0}).$$

La difficulté principale est de spécifier la forme des contraintes héritées du niveau supérieur. Premièrement, il est préférable que l'ensemble admissible résultant (au niveau inférieur) soit une boite afin de préserver la cohérence et l'efficacité de l'algorithme à travers les niveaux. Nous souhaitons aussi garantir la caractère admissible *pour le niveau du dessus* du point d'essai prolongé $x_{r,k} + s_{r,k}$ par rapport aux contraintes de bornes. Finalement, nous voudrions assurer que ce point candidat reste à l'intérieur de la région de confiance du niveau du dessus $\mathcal{B}_{r,k}$. Malheureusement, *la prolongation de la restriction d'une boite du niveau r vers le niveau r n'est généralement pas inclue dans la boite de départ.*

Nous sommes donc forcés de modifier notre technique de représentation des boites du niveau du dessus au niveau du bas si nous insistons sur le fait que leur prolongation doit satisfaire les contraintes représentées par la boite du niveau supérieur. Ceci est fortement désirable pour la boite $\mathcal{F}_r$ du niveau du dessus définissant les contraintes de bornes originales du problème car nous souhaitons préserver le caractère admissible à tous les niveaux. D'un autre côté, nous pourrions accepter un peu de flexibilité pour les boites grossières correspondant à la région de confiance du dessus $\mathcal{B}_{r,k}$, parce qu'on s'attend à ce qu'un pas dont la norme est proportionnelle à la taille de la région de confiance soit suffisant pour assurer la convergence (même si il n'y a pas

d'inclusion stricte) sans être excessivement restrictif. Cela nous mène donc à une stratégie en deux temps, où nous représentons séparément au niveau inférieur, d'une part les contraintes de bornes d'une façon qui garantisse le caractère admissible du pas prolongé, et d'autre part la région de confiance du niveau du dessus, peut-être de manière moins stricte. Si $\mathcal{F}_{r-1}$ représente les contraintes de bornes au niveau grossier et $\mathcal{A}_{r-1}$ représente la version grossière des contraintes de la région de confiance du niveau supérieur, alors le pas à l'itération ℓ de la minimisation au niveau grossier doit être inclu dans la boite

$$\mathcal{W}_{r-1,\ell} \stackrel{\text{def}}{=} \mathcal{F}_{r-1} \cap \mathcal{A}_{r-1} \cap \mathcal{B}_{r-1,\ell}.$$

Nous détaillons ci-dessous la façon dont $\mathcal{F}_{r-1}$ et $\mathcal{A}_{r-1}$ sont calculées.

Si plus de deux niveaux sont disponibles ($r > 1$), la même technique peut être appliquée récursivement, le processus se terminant au niveau 0, où il n'existe aucun modèle plus grossier et où le modèle de Taylor est donc systématiquement utilisé. Considérons à présent les détails du processus dans ce cadre plus général. Nous considérons l'itération k au niveau i et nous supposons que $x_{i,k}$ est un itéré dans la minimisation de f_i à l'intérieur d'une itération q au niveau $i+1$ où f_i à été choisie comme modèle de f_{i+1} (c'est-à-dire l'itération $(i+1, q)$ est une *itération récursive*).

Nous commençons par considérer la représentation des bornes du problème aux niveaux grossiers. Au niveau i, nous définissons

$$\mathcal{F}_i \stackrel{\text{def}}{=} \{x \mid l_i \leq x \leq u_i\} \tag{F.8}$$

le domaine admissible "restreint", où

$$[l_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \frac{1}{\|P_{i+1}\|_\infty} \max_{t=1,\ldots,n_{i+1}} \left\{ \begin{array}{lll} [l_{i+1} - x_{i+1,q}]_t & \text{when} & [P_{i+1}]_{tj} > 0 \\ [x_{i+1,q} - u_{i+1}]_t & \text{when} & [P_{i+1}]_{tj} < 0 \end{array} \right\} \tag{F.9}$$

et

$$[u_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \frac{1}{\|P_{i+1}\|_\infty} \min_{t=1,\ldots,n_{i+1}} \left\{ \begin{array}{lll} [u_{i+1} - x_{i+1,q}]_t & \text{when} & [P_{i+1}]_{tj} > 0 \\ [x_{i+1,q} - l_{i+1}]_t & \text{when} & [P_{i+1}]_{tj} < 0 \end{array} \right\} \tag{F.10}$$

pour $j = 1, \ldots, n_i$. L'idée derrière cette généralisation de la définition de Gelman and Mandel (1990), formulée à l'origine pour des opérateurs de prolongation plus spécifiques[(1)], est d'utiliser la structure de P_{i+1} pour calculer un ensemble de bornes grossières $\mathcal{F}_i$ *dans le but de garantir le caractère admissible de sa prolongation au niveau fin*, c'est-à-dire

$$l_{i+1} \leq x_{i+1} + P_{i+1}(l_i - x_i) \leq x_{i+1} + P_{i+1}(u_i - x_i) \leq u_{i+1}$$

[(1)] La formulation originale est restreinte au cas où $||P_{i+1}||_\infty \leq 1$ et $P_{i+1} > 0$, et est donnée par

$$[l_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \max_{t=1,\ldots,n_{i+1}:[P_{i+1}]_{tj}>0} [l_{i+1} - x_{i+1,q}]_t,$$

$$[u_i]_j \stackrel{\text{def}}{=} [x_{i,0}]_j + \max_{t=1,\ldots,n_{i+1}:[P_{i+1}]_{tj}>0} [x_{i+1,q} - u_{i+1}]_t.$$

Nous étendons cette définition pour couvrir les opérateurs de prolongation avec $||P_{i+1}||_\infty > 1$ et aussi pour gérer les élément négatifs de P_{i+1} (comme pour l'interpolation cubique, par exemple), ce qui impose de tenir compte à la fois des contraintes de bornes supérieure et inférieure dans la définition des contraintes de bornes supérieure et inférieure du niveau grossier.

pour tout $x_{i+1} \in \mathcal{F}_{i+1}$, pour tout $x_i \in \mathcal{F}_i$. Cette propriété est prouvée par le Lemme 2.3.2 ci-dessous.

Nous nous tournons maintenant vers la représentation grossière de la région de confiance du dessus. Au niveau i, nous définissons aussi

$$\mathcal{A}_i = \{x \mid v_i \leq x \leq w_i\}, \tag{F.11}$$

la restriction des contraintes de la région de confiance héritées des niveaux r à $i+1$ à travers $x_{i+1,q}$, et calculées par le biais de l'opérateur de restriction R_{i+1}. Les $j^{\text{ième}}$ composantes de v_i et w_i sont données par

$$\begin{aligned} [v_i]_j \quad = \quad & \sum_{u=1,[R_{i+1}]_{ju}>0}^{n_{i+1}} [R_{i+1}]_{ju}[\max(v_{i+1}, x_{i+1,q} - \Delta_{i+1,q}e)]_u \\ & + \sum_{u=1,[R_{i+1}]_{ju}<0}^{n_{i+1}} [R_{i+1}]_{ju}[\min(w_{i+1}, x_{i+1,q} + \Delta_{i+1,q}e)]_u \end{aligned} \tag{F.12}$$

et

$$\begin{aligned} [w_i]_j \quad = \quad & \sum_{u=1,[R_{i+1}]_{ju}>0}^{n_{i+1}} [R_{i+1}]_{ju}[\min(w_{i+1}, x_{i+1,q} + \Delta_{i+1,q}e)]_u \\ & + \sum_{u=1,[R_{i+1}]_{ju}<0}^{n_{i+1}} [R_{i+1}]_{ju}[\max(v_{i+1}, x_{i+1,q} - \Delta_{i+1,q}e)]_u, \end{aligned} \tag{F.13}$$

où $e \in \mathbb{R}^n$ est un vecteur dont les composantes sont toutes égales à 1 (et où nous définissons $v_r = -\infty$ et $w_r = +\infty$ pour être cohérents). Notons que, comme autorisé par la précédente discussion, le choix d'utiliser R_i pour restreindre ces bornes implique que les itérés récursifs au niveau i ne sont plus nécessairement inclus dans le région de confiance du niveau i mais ne peuvent pas en être éloignés non plus. En effet, en rappelant que $||R_i||_\infty = 1$ pour $i = 1, ..., r$, nous avons que

$$\|x_{i,k+1} - x_{i,k}\|_\infty \leq \|P_i\|_\infty \|x_{i-1,*} - x_{i-1,0}\|_\infty \leq 2\|P_i\|_\infty \Delta_{i,k}, \tag{F.14}$$

où la dernière inégalité est prouvée par le Lemme 2.3.3 ci-dessous.

Si la région de confiance du niveau i autour de l'itéré $x_{i,k}$ est définie par

$$\mathcal{B}_{i,k} = \{x_{i,k} + s \in \mathbb{R}^{n_i} \mid \|s\|_\infty \leq \Delta_{i,k}\},$$

nous devons alors trouver un pas $s_{i,k}$ qui réduise suffisamment un modèle de f_i à l'intérieur de la région

$$\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{A}_i \cap \mathcal{B}_{i,k}.$$

Observons que l'ensemble $\mathcal{W}_{i,k}$ peut à la fois être vu comme $\mathcal{W}_{i,k} = \mathcal{L}_i \cap \mathcal{B}_{i,k}$, l'intersection entre un domaine dépendant du niveau $\mathcal{L}_i \stackrel{\text{def}}{=} \mathcal{F}_i \cap \mathcal{A}_i$ et une région de confiance $\mathcal{B}_{i,k}$ dépendant de l'itération, ou comme $\mathcal{W}_{i,k} = \mathcal{F}_i \cap \mathcal{S}_{i,k}$, l'intersection de $\mathcal{F}_i$, l'ensemble admissible pour les contraintes rigides avec $\mathcal{S}_{i,k} \stackrel{\text{def}}{=} \mathcal{A}_i \cap \mathcal{B}_{i,k}$ l'ensemble admissible pour les contraintes souples. Ce dernier ensemble peut être interprété comme une région de confiance "composite" qui inclut toutes les contraintes imposées par les régions de confiance des niveaux i et supérieurs. Notons que tous les ensembles impliqués sont des boites, ce qui rend leur représentation et leur intersection simples à calculer.

Lorsque $\mathcal{W}_{i,k}$ est connu, nous choisissons alors un modèle pour f_{i+1} entre

$$m_{i+1,q}(x_{i+1,q} + s_{i+1}) = f_{i+1}(x_{i+1,q}) + \langle g_{i+1,q}, s_{i+1}\rangle + \tfrac{1}{2}\langle s_{i+1}, H_{i+1,q}s_{i+1}\rangle, \quad \text{(F.15)}$$

la série de Taylor tronquée habituelle pour f_{i+1} (avec $g_{i+1,q} = \nabla_x f_{i+1}(x_{i+1,q})$ et $H_{i+1,q}$ étant une approximation symétrique de $\nabla^2_x f_{i+1}(x_{i+1,q})$), et sa représentation grossière f_i. Dans le second cas, nous supposons que f_{i+1} et son modèle grossier f_i sont *cohérents au premier ordre*, c'est-à-dire que $g_{i,0} = R_{i+1}g_{i+1,q}$. Cette hypothèse n'est pas restrictive étant donné que nous pouvons toujours choisir un modèle de f_{i+1} qui soit cohérent au premier ordre en ajoutant un terme de correction du premier ordre à f_i, comme dans

$$f_i(x_{i,0} + s_i) + \langle R_{i+1}g_{i+1,q} - \nabla_x f_i(x_{i,0}), s_i\rangle.$$

Si le modèle f_i est choisi (ce qui n'est possible que si $i > 0$), la détermination du pas consiste alors à résoudre (approximativement) le problème grossier avec contraintes de bornes suivant

$$\min_{x_{i,0}+\tilde{s}_i \in \mathcal{L}_i} f_i(x_{i,0} + \tilde{s}_i). \quad \text{(F.16)}$$

Cette minimisation produit un pas s_i tel que $f_i(x_{i,0} + s_i) < f_i(x_{i,0})$ qui doit alors être transféré au niveau $i + 1$ par prolongation, c'est-à-dire $s_{i+1} = P_{i+1}s_i$. Notons que

$$\langle g_{i+1,q}, s_{i+1}\rangle = \langle g_{i+1,q}, P_{i+1}s_i\rangle = \frac{1}{\sigma_{i+1}}\langle R_{i+1}g_{i+1,q}, s_i\rangle. \quad \text{(F.17)}$$

Comme la décroissance de f_i réalisée par s_i peut être approximée au premier ordre par $f_i(x_{i,0}) - f_i(x_{i,0} + s_i) \approx \langle g_{i,0}, s_i\rangle = \langle R_{i+1}g_{i+1,q}, s_i\rangle$, la décroissance du modèle au niveau $i + 1$, lorsque les pas sont réalisés au niveau i, est calculée comme $[f_i(x_{i,0}) - f_i(x_{i,0} + s_i)]/\sigma_{i+1}$, en utilisant (F.17).

Mais cela a-t-il toujours un sens d'utiliser le modèle du niveau grossier ? La réponse dépend évidemment du bénéfice attendu de la solution de (F.16). Dans Gratton et al. (2008*b*), il était suffisant de tester si $\|g_{i,0}\|_2 = \|R_{i+1}g_{i+1,q}\|_2$ est assez grand par rapport à $\|g_{i+1,q}\|_2$. Cependant, cette mesure de criticalité est inadéquate dans notre contexte car (F.16) est maintenant un problème avec contraintes de bornes. Dans ce chapitre, nous supposons que nous utilisons une mesure de criticalité $\chi_{i+1,q}$ conçue pour l'optimisation avec contraintes de bornes définie pour chaque $x_{i+1,q} \in \mathcal{L}_{i+1}$. Le choix de cette mesure sera discuté dans le Chapitre 3. Ensuite, si la restriction du problème au départ de l'itéré non-critique $x_{i+1,q}$ du niveau $i + 1$ vers le niveau i n'est pas encore critique au premier ordre, c'est-à-dire si

$$\chi_{i,0} \geq \kappa_\chi \chi_{i+1,q}, \quad \text{(F.18)}$$

pour une constante $\kappa_\chi \in (0, \max\{1, \sigma_i\})$, nous pouvons poursuivre à ce niveau grossier. Dans le cas contraire, la récursion n'a pas d'intérêt et nous devrions plutôt choisir (F.15).

Une fois que nous avons décidé de résoudre approximativement (F.16), nous devons aussi décider de ce que nous entendons par "approximativement". Nous avons choisi de terminer la minimisation au niveau r si $\chi_{r,k} \leq \epsilon_r$, pour une certain $\epsilon_r > 0$ et, dans l'esprit de (F.18), de terminer la minimisation à un niveau grossier i à l'itéré (i, p) dès que l'inégalité

$$\chi_{i,p} < \epsilon_i \stackrel{\text{def}}{=} \kappa_\chi \epsilon_{i+1},$$

est satisfaite. Nous définissons alors $x_{i,*} = x_{i,p}$, $s_i = x_{i,*} - x_{i,0}$ et $s_{i+1,q} = P_{i+1}s_i$.

Si, d'un autre côté, nous décidons, à l'itération $(i+1,q)$, d'utiliser le modèle de Taylor $m_{i+1,q}$ donné par (F.15), un pas $s_{i+1,q}$ est alors calculé, produisant une décroissance suffisante de la valeur de ce modèle dans le sens usuel de ce terme pour les méthodes de régions de confiance avec contraintes convexes (définies ici par l'ensemble $\mathcal{L}_{i+1}$), c'est-à-dire $s_{i+1,q}$ est tel qu'il satisfait

$$m_{i+1,q}(x_{i+1,q}) - m_{i+1,q}(x_{i+1,q} + s_{i+1,q}) \geq \kappa_{\text{red}}\chi_{i+1,q} \min\left[\frac{\chi_{i+1,q}}{\beta_{i+1,q}}, \Delta_{i+1,q}, 1\right],$$

pour une constante $\kappa_{\text{red}} \in (0, \frac{1}{2})$ et $\beta_{i+1,q} \stackrel{\text{def}}{=} 1 + \|H_{i+1,q}\|_{\infty,1}$ où la norme $\|.\|_{\infty,1}$ est définie pour toutes les matrices M par $\|M\|_{\infty,1} \stackrel{def}{=} \max_{x\neq 0}\left\{\frac{\|Mx\|_1}{\|x\|_\infty}\right\}$. Malgré son apparence technique, cette condition, connue sous le nom de condition de Cauchy modifiée, n'est pas exagérément restrictive et peut être garantie pour des algorithmes pratiques, comme décrits par exemple dans la Section 12.1 de Conn et al. (2000).

Nous explicitons formellement notre algorithme sur la page suivante comme l'Algorithme RMTR$_\infty$. Il utilise les constantes $0 < \eta_1 \leq \eta_2 < 1$ et $0 < \gamma_1 \leq \gamma_2 < 1$ et Δ_i^{s} $(i = 0, \ldots, r)$.

Algorithm F.2.2: RMTR$_\infty$ $(i, x_{i,0}, g_{i,0}, \chi_{i,0}, \mathcal{F}_i, \mathcal{A}_i, \epsilon_i)$

Etape 0: Initialisation. Calculer $f_i(x_{i,0})$. Poser $k = 0$ et

$$\mathcal{L}_i = \mathcal{F}_i \cap \mathcal{A}_i \quad \text{et} \quad \mathcal{W}_{i,0} = \mathcal{L}_i \cap \mathcal{B}_{i,0},$$

où $\mathcal{B}_{i,0} = \{x_{i,0} + s \in \mathbb{R}^{n_i} \,|\, \|s\|_\infty \leq \Delta_{i,0} = \Delta_i^{\mathrm{s}}\}$.

Etape 1: Choix du modèle. Si $i = 0$, aller à l'étape 3. Sinon, calculer $\mathcal{L}_{i-1}$ et $\chi_{i-1,0}$. Si (F.18) n'est pas satisfait, aller à l'étape 3. Sinon, choisir d'aller à l'étape 2 ou à l'étape 3.

Etape 2: Calcul du pas récursif. Appeler l'Algorithme

$$\mathbf{RMTR}_\infty(i-1, R_i x_{i,k}, R_i g_{i,k}, \chi_{i-1,0}, \mathcal{F}_{i-1}, \mathcal{A}_{i-1}, \kappa_\chi \epsilon_i),$$

qui renvoie une solution approximée $x_{i-1,*}$ de (2.17). Définir ensuite $s_{i,k} = P_i(x_{i-1,*} - R_i x_{i,k})$, poser $\delta_{i,k} = \frac{1}{\sigma_i}[f_{i-1}(R_i x_{i,k}) - f_{i-1}(x_{i-1,*})]$ et aller à l'étape 4.

Etape 3: Calcul du pas de Taylor. Choisir $H_{i,k}$ et calculer un pas $s_{i,k} \in \mathbb{R}^{n_i}$ qui réduise suffisamment le modèle $m_{i,k}$ donné par (2.16) au sens de (2.21) et tel que $x_{i,k} + s_{i,k} \in \mathcal{W}_{i,k}$. Poser $\delta_{i,k} = m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k})$.

Etape 4: Acceptation du point d'essai. Calculer $f_i(x_{i,k} + s_{i,k})$ et

$$\rho_{i,k} = [f_i(x_{i,k}) - f_i(x_{i,k} + s_{i,k})]/\delta_{i,k}. \tag{F.19}$$

Si $\rho_{i,k} \geq \eta_1$, alors définir $x_{i,k+1} = x_{i,k} + s_{i,k}$; sinon, définir $x_{i,k+1} = x_{i,k}$.

Etape 5: Test d'arrêt. Calculer $g_{i,k+1}$ et $\chi_{i,k+1}$. Si $\chi_{i,k+1} \leq \epsilon_i$ ou $x_{i,k+1} \notin \mathcal{A}_i$, alors retouner la solution approximée $x_{i,*} = x_{i,k+1}$.

Etape 6: Mise à jour de la région de confiance. Poser

$$\Delta_{i,k+1} \in \begin{cases} [\Delta_{i,k}, +\infty) & \text{si } \rho_{i,k} \geq \eta_2, \\ [\gamma_2 \Delta_{i,k}, \Delta_{i,k}] & \text{si } \rho_{i,k} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}] & \text{si } \rho_{i,k} < \eta_1, \end{cases} \tag{F.20}$$

et $\mathcal{W}_{i,k+1} = \mathcal{L}_i \cap \mathcal{B}_{i,k+1}$ où

$$\mathcal{B}_{i,k+1} = \{x_{i,k+1} + s \in \mathbb{R}^{n_i} \,|\, \|s\|_\infty \leq \Delta_{i,k+1}\}.$$

Incrémenter k de 1 et aller à l'étape 1.

Quelques commentaires sont à présent nécessaires pour une compréhension totale de l'algorithme.

1. Lors de l'étape d'initialisation, Δ_i^{s} est le rayon initial de la région de confiance locale et dépend uniquement du niveau.

2. Le test de la valeur de i au début de l'étape 1 sert à identifier le niveau le plus grossier auquel la récursion n'est plus possible. Dans ce cas, une itération de Taylor est le seul choix possible.

3. En conséquence de la discussion précédant l'équation (F.14), $x_{i,k+1}$ peut ne pas appartenir à la région de confiance composite $\mathcal{A}_i$ lorsque le pas $s_{i,k}$ est calculé par le biais d'une itération récursive. Cependant, comme indiqué précédemment, nous souhaitons limiter la longueur du pas au niveau $i+1$ à un multiple de la taille de la région de confiance. Etant donné (F.14) et la définition de $\mathcal{A}_i$, nous pouvons atteindre cet objectif en arrêtant les itérations au niveau i dès que l'itéré sort de la région de confiance composite $\mathcal{A}_i$. Ceci explique le second test d'arrêt à l'étape 5 de l'algorithme.

4. La différence entre la formule de restriction (F.8)-(F.10) pour les contraintes rigides et (F.11)-(F.13) pour les contraintes souples nécessite de passer séparément $\mathcal{A}_i$ et $\mathcal{F}_i$ à l'algorithme au niveau i, puisqu'il est nécessaire de calculer $\mathcal{L}_i$ indépendamment à chaque niveau.

5. Le problème original (F.4) est résolu en appelant RMTR_∞ depuis un niveau $r+1$ virtuel auquel on suppose que la région de confiance est infinie.

6. S'il n'y a qu'un seul niveau ($r=0$), alors RMTR_∞ se réduit à l'application de l'Algorithme F.2.1 décrit précédemment.

Comme souvent dans le cadre des méthodes de région de confiance, les itérations auxquelles $\rho_{i,k} \geq \eta_1$ sont appelées *réussies* et même *très réussies* si $\rho_{i,k} \geq \eta_2$. Lors de ces itérations, le point d'essai $x_{i,k} + s_{i,k}$ est accepté comme le nouvel itéré et le rayon de la région de confiance correspondante peut être agrandi. Si l'itération est *ratée*, le point d'essai est rejeté et le rayon de la région de confiance est réduit.

F.2.3 Convergence de l'algorithme

Cet algorithme est prouvé globallement convergent, c'est-à-dire que

$$\lim_{k\to\infty} \chi_{i,k} = 0$$

peu importe le choix du point de départ. Les détails de la démonstration sont disponibles dans la version complète de la thèse en anglais.

F.2.4 Identification des contraintes actives

Dans une seconde partie, la théorie de Conn et al. (1993) sur l'identification des contraintes actives par un algorithme de région de confiance dont le solveur interne est basé sur la calcul du pas de Cauchy est étendue à toute méthode de région de confiance dont le solveur interne respecte la condition de Cauchy modifiée, ou condition

de décroissance suffisante. Cela nous permet, entre autres, d'affirmer qu'une méthode de région de confiance qui utilise une technique de lissage de type Gauss-Seidel pour le calcul des pas identifie l'ensemble des contraintes actives en un nombre fini d'itérations. Par conséquent, si toutes les contraintes qui sont actives avant la réalisation d'un pas grossier sont gelées pendant le calcul de celui-ci, l'algorithme RMTR_∞ présenté dans ce chapitre identifie lui-aussi l'ensemble des contraintes actives à la solution exacte en un nombre fini d'itérations. La démonstration est détaillée dans la version complète de la thèse en anglais.

F.3 Critères d'arrêt pour l'optimisation avec contraintes de bornes

Dans ce chapitre, nous regardons de manière approfondie la définition d'un critère d'arrêt approprié à l'optimisation nonlinéaire avec contraintes de bornes. Plus précisément, nous nous intéressons au fait de relier les mesures de criticalité utilisées comme critères d'arrêt pour les algorithmes d'optimisation avec contraintes de bornes et l'analyse d'*erreur inverse ("backward error")* provenant de l'algèbre linéaire. Cela nous mènera finalement à considérer le problème d'erreur inverse du point de vue de l'optimisation multicritères. Nous montrons à la fin du chapitre que les différentes mesures discutées dans ce chapitre satisfont les hypothèses de convergence de RMTR_∞ décrites dans le Chapitre 2.

Nous sommes intéressés par la résolution d'un problème du type

$$\min_{\mathcal{F}} f(x), \tag{F.21}$$

où $\mathcal{F} = \{x \in \mathbb{R}^n | l \leq x \leq u\}$ est un ensemble de contraintes de bornes et $l, u \in \mathbb{R}^n$. Nous définissons l'ensemble des *contraintes actives liées*, pour tout $x \in \mathcal{F}$, comme $\mathcal{A}(x) = \mathcal{A}^-(x) \cup \mathcal{A}^+(x)$ avec

$$\begin{array}{rcl} \mathcal{A}^-(x) = \{j \in \{1,...,n\} \mid [x]_j & = & [l]_j \quad \text{et} \quad [\nabla_x f(x)]_j > 0\} \\ \mathcal{A}^+(x) = \{j \in \{1,...,n\} \mid [x]_j & = & [u]_j \quad \text{et} \quad [\nabla_x f(x)]_j < 0\}. \end{array}$$

Dans ce contexte, si $[\nabla_x f(x_*)]_j = 0$ pour tout $j \notin \mathcal{A}(x_*)$, alors x_* est un point critique du premier ordre de (F.21).

Nous considérons des méthodes itératives d'optimisation qui produisent une suite d'itérés x_k qui converge vers une solution du premier ordre x_* du problème à résoudre. Cependant, cette séquence peut être infinie. Nous sommes donc intéressés par l'identification d'un bon moment pour arrêter l'algorithme, afin d'avoir une solution approchée raisonablement fiable. Une façon d'exprimer ce problème consiste à arrêter les itérations lorsque l'itéré courant x_k est tel que

$$||x_k - x_*|| < \epsilon,$$

où ϵ est la tolérance que nous acceptons sur la distance entre la solution approchée et la solution critique du premier ordre. Cependant, nous ne connaissons généralement pas la solution exacte puisque nous sommes précisément en train de la chercher. En conséquence, nous préférons considérer l'erreur inverse, qui remplace la question *A quelle distance de la solution se trouve l'itéré courant x_k ?* par *S'il existe un problème de minimisation (P) tel que x_k est une de ses solutions au premier ordre, à quelle distance du problème original (F.21) se trouve (P) ?* Dans le contexte de l'analyse d'erreur inverse, nous arrêtons l'algorithme à une itération k lorsque x_k est un point critique du premier ordre d'une version perturbée du problème original (F.21):

$$\min_{l+\Delta l \leq x \leq u+\Delta u} f(x) + \Delta f + \Delta g^T x,$$

et lorsque les perturbations $\Delta l, \Delta u, \Delta f, \Delta g \in \mathbb{R}^n$ sont suffisamment petites. La condition suffisante d'optimalité au premier ordre implique $[\nabla_x f(x_k) + \Delta g]_j = 0$ pour tout $j \notin \mathcal{A}_\Delta(x_k)$, où $\mathcal{A}_\Delta(x) = \mathcal{A}^-_\Delta(x) \cup \mathcal{A}^+_\Delta(x)$ avec

$$\begin{array}{l} \mathcal{A}^-_\Delta(x) = \{j \in \{1,...,n\} \mid [x]_j = [l]_j + [\Delta l]_j \text{ et } [\nabla_x f(x) + \Delta g]_j > 0\} \\ \mathcal{A}^+_\Delta(x) = \{j \in \{1,...,n\} \mid [x]_j = [u]_j + [\Delta u]_j \text{ et } [\nabla_x f(x) + \Delta g]_j < 0\}. \end{array}$$

La valeur de Δf n'apparaissant pas dans cette condition suffisante, nous pouvons poser $\Delta f = 0$ sans perte de généralité. Finalement, nous cherchons

$$y \stackrel{def}{=} (\Delta g\,;\,\Delta l\,;\,\Delta u) \in \mathcal{Y}_k$$

tel que

$$\mathcal{Y}_k \stackrel{def}{=} \{(\Delta g\,;\,\Delta l\,;\,\Delta u) \in \mathbb{R}^{3n} : [\nabla_x f(x_k) + \Delta g]_j = 0 \text{ pour tout } j \notin \mathcal{A}_\Delta(x_k)\}$$

et où y est un vecteur composé des trois vecteurs de perturbation $\Delta g, \Delta l$ et Δu, réunis en un long vecteur $y = (\Delta g\,;\,\Delta l\,;\,\Delta u)$. Uune norme produit doit en outre être définie sur l'espace des perturbations pour tous les vecteurs du type y. L'algorithme est alors arrêté si

$$\inf_{y \in \mathcal{Y}_k} \parallel y = (\Delta g\,;\,\Delta l\,;\,\Delta u) \parallel < \epsilon(\epsilon_l, \epsilon_u, \epsilon_g),$$

où $\epsilon_l, \epsilon_u, \epsilon_g \in \mathbb{R}$ sont des tolérances choisies qui représentent dans la plupart des cas un ordre de grandeur correspondant à la précision de calcul de g, l et u. De plus, notons que cet infimum est en fait un minimum. En effet, lorsqu'on regarde $\mathcal{Y}_k$, nous voyons qu'il est égal au produit cartésien, sur tous les $j = 1, ..., n$, d'ensembles $[\mathcal{Y}_k]_j$ qui contiennent toutes les solutions possibles pour $\{y\}_j \stackrel{not.}{=} ([\Delta g]_j\,;\,[\Delta l]_j\,;\,[\Delta u]_j)$. Les ensembles $[\mathcal{Y}_k]_j$ sont tous composés par l'union de deux produits cartésiens de trois éléments : nous avons nécessairement, pour tout j,

$$[\Delta g]_j = [-\nabla_x f(x_k)]_j \text{ ou } \begin{cases} [\Delta_l]_j = [x_k - l]_j & \text{if } [\nabla_x f(x_k) + \Delta g]_j > 0 \\ [\Delta_u]_j = [x_k - u]_j & \text{if } [\nabla_x f(x_k) + \Delta g]_j < 0 \end{cases}$$

tandis que les autres composantes de $\{y\}_j$ peuvent prendre n'importe quelle valeur dans $\mathbb{R}^n$ à condition que $l + \Delta l \leq x_k \leq u + \Delta u$ et, en conséquence,

$$[\mathcal{Y}_k]_j = \left\{ \begin{array}{c} \left\{ ([-\nabla_x f(x_k)]_j; [\Delta l]_j; [\Delta u]_j) \;:\; \begin{array}{l} [\Delta l]_j, [\Delta u]_j \in \mathbb{R} \\ [l + \Delta l]_j \leq [x_k]_j \leq [u + \Delta u]_j \end{array} \right\} \\ \cup \\ \left\{ ([\Delta g]_j; [x_k - l]_j; [\Delta u]_j) \;:\; \begin{array}{l} [\Delta g]_j, [\Delta u]_j \in \mathbb{R} \\ [x_k]_j \leq [u + \Delta u]_j \\ [\nabla_x f(x_k) + \Delta g]_j > 0 \end{array} \right\} \\ \cup \\ \left\{ ([\Delta g]_j, [\Delta l]_j, [x_k - u]_j) \;:\; \begin{array}{l} [\Delta g]_j, [\Delta l]_j \in \mathbb{R} \\ [l + \Delta l]_j \leq [x_k]_j \\ [\nabla_x f(x_k) + \Delta g]_j < 0 \end{array} \right\}. \end{array} \right.$$

Finalement, nous en déduisons que $\mathcal{Y}_k$ est l'union d'un nombre fini de produits cartésiens d'ensembles fermés et qu'il est donc lui-même un ensemble fermé. Par conséquent, nous avons pour un certain $y_0 \in \mathcal{Y}_k$ (qui existe étant donné que $\mathcal{Y}_k$ n'est pas vide)

$$\begin{aligned} \inf_{y \in \mathcal{Y}_k} \parallel (\Delta g\,;\,\Delta l\,;\,\Delta u) \parallel &= \inf_{\substack{y \,\in\, \mathcal{Y}_k \\ ||y|| \,\leq\, ||y_0||}} \parallel (\Delta g\,;\,\Delta l\,;\,\Delta u) \parallel \\ &= \min_{\substack{y \,\in\, \mathcal{Y}_k \\ ||y|| \,\leq\, ||y_0||}} \parallel (\Delta g\,;\,\Delta l\,;\,\Delta u) \parallel \\ &= \min_{y \in \mathcal{Y}_k} \parallel (\Delta g\,;\,\Delta l\,;\,\Delta u) \parallel \end{aligned}$$

où la première et la troisième égalités viennent du fait que $y_0 \in \mathcal{Y}_k$ et de la définition de la fonction objectif, tandis que la deuxième égalité est satisfaite parce que l'ensemble des contraintes est fermé borné puisqu'il s'agit de l'intersection entre un ensemble fermé ($\mathcal{Y}_k$) et un ensemble borné ($\{y \in \mathcal{Y}_k : ||y|| \leq ||y_0||\}$). En conclusion, nous pouvons choisir de terminer l'algorithme dès que

$$\min_{y \in \mathcal{Y}_k} \parallel y = (\Delta g\,;\, \Delta l\,;\, \Delta u) \parallel < \epsilon(\epsilon_l, \epsilon_u, \epsilon_g).$$

Nous nous consacrons à présent à la recherche d'un $y \in \mathcal{Y}_k$ qui soit optimal pour

$$\chi_k \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} \parallel y = (\Delta g\,;\, \Delta l\,;\, \Delta u) \parallel < \epsilon(\epsilon_l, \epsilon_u, \epsilon_g),$$

dans le but de définir une mesure de criticalité générale basée sur l'analyse d'erreur inverse et qui soit intéressante comme critère d'arrêt. Dans ce travail, nous avons choisi de regarder plus précisément deux définitions de cette norme spécifique. La première norme que nous considérons est

$$\chi_k^{out} = \min_{y \in \mathcal{Y}_k} ||y = (\Delta g\,;\, \Delta l\,;\, \Delta u)||_{out} \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} (\alpha_g ||\Delta g||_g + \alpha_l ||\Delta l||_l + \alpha_u ||\Delta u||_u), \tag{F.22}$$

où $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ et où les normes sur $\Delta l, \Delta u, \Delta g$ sont des normes *monotones*, au sens de

$$\text{si} \quad \forall j \in \{1, ..., n\} \quad |[u]_j| \geq |[v]_j| \quad \text{alors} \quad ||u|| \geq ||v|| \qquad \forall u, v \in \mathbb{R}^n.$$

Le second choix est

$$\chi_k^{in} = \min_{y \in \mathcal{Y}_k} ||y = (\Delta g\,;\, \Delta l\,;\, \Delta u)||_{in} \stackrel{def}{=} \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\ ||_{glu}, \tag{F.23}$$

où $(\alpha_g, \alpha_l, \alpha_u) \in (0,1]^3$ et où la norme sur la somme est à nouveau une norme monotone. Il est simple de vérifier que autant $||.||_{in}$ que $||.||_{out}$ respectent les propriétés des normes. Notons en outre qu'elles sont toutes deux symétriques à cause de la présence des valeurs absolues et du caractère positif des poids α_g, α_l et α_u. Dans la suite de ce chapitre, nous appellerons $\mathcal{S}_k^{out}$ et $\mathcal{S}_k^{in}$ l'ensemble de toutes les solutions de (F.22) et (F.23), respectivement. Nous avons alors immédiatement la propriété suivante :

$$\mathcal{S}_k^{out} \subseteq \mathcal{Y}_k \text{ and } \mathcal{S}_k^{in} \subseteq \mathcal{Y}_k.$$

Nous nous restreignons au cas où $||.||_g, ||.||_l, ||.||_u$ et $||.||_{glu}$ sont des normes monotones car cette restriction va nous permettre de caractériser plus facilement la solution du problème et finalement d'en trouver une forme explicite. Notons finalement que si $||.||_g = ||.||_l = ||.||_u = ||.||_{glu} = ||.||_p$, avec $1 \leq p \leq \infty$, alors

$$\chi_k^{in,p} \leq \chi_k^{out,p}$$

où $\chi_k^{in,p} = \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\ ||_p$ et $\chi_k^{out,p} = \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_p + \alpha_l ||\Delta l||_p + \alpha_u ||\Delta u||_p$. En effet,

$$\begin{aligned}
\chi_k^{in,p} &= \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\ ||_p \\
&\leq \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g|\ ||_p + ||\alpha_l |\Delta l| + \alpha_u |\Delta u|\ ||_p \\
&\leq \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g|\ ||_p + ||\alpha_l |\Delta l|\ ||_p + ||\alpha_u |\Delta u|\ ||_p \\
&= \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_p + \alpha_l ||\Delta l||_p + \alpha_u ||\Delta u||_p,
\end{aligned}$$

où nous avons utilisé le fait que α_g, α_l et α_u sont positifs, ainsi que les propriétés des normes.

Définissons maintenant l'ensemble des *indices indécis*

$$\mathcal{U}_k = \{j \in \{1, ..., n\} \mid [\nabla_f(x_k)]_j \neq 0 \text{ et } j \notin \mathcal{A}(x_k)\}.$$

qui jouera un rôle important dans le résultat-clé qui caractérise $\mathcal{S}_k^{out}$ et $\mathcal{S}_k^{in}$. Le Lemme 3.1.1 montre que la solution optimale de (F.22), ainsi que la solution optimale de (F.23), se situe dans un ensemble spécifique $\mathcal{P}_k \subseteq \mathcal{Y}_k$ tel que le choix de chacune de ses composantes $\{y^*\}_j = ([\Delta g^*]_j\,;\,[\Delta l^*]_j\,;\,[\Delta u^*]_j)$ est indépendant du choix des autres. De plus, cette caractérisation de la solution optimale nous laisse uniquement le choix entre deux ensembles de valeurs explicites pour chaque $\{y\}_j = ([\Delta g]_j\,;\,[\Delta l]_j\,;\,[\Delta u]_j)$, $j \in \mathcal{U}_k$.

Une conséquence importante du Lemme 3.1.1 est que la minimisation de $y \in \mathcal{P}_k$ est *séparable* en j, ce qui est prouvé dans le Lemme 3.1.2.

Jusqu'à présent, et afin de garder un caractère général, nous n'avons pas spécifié le choix des normes $||.||_g, ||.||_l, ||.||_u$ et $||.||_{glu}$. Nous allons maintenant voir que le choix de certaines normes spécifiques mène à des expressions de χ_k^{out} et de χ_k^{in} qui sont en fait parmi les mesures de criticalité les plus répandues dans le cadre de l'optimisation avec contraintes de bornes. Considérons d'abord le problème (F.22)

$$\chi_k^{out} = \min_{y \in \mathcal{Y}_k} \alpha_g ||\Delta g||_g + \alpha_l ||\Delta l||_l + \alpha_u ||\Delta u||_u.$$

Dans ce cas, si $||.||_g = ||.||_l = ||.||_u = ||.||_1$, alors il est possible d'obtenir une valeur explicite de χ_k^{out}. Cela est montré dans le Théorème 3.2.2, où le résultat du Lemme 3.1.1 joue un rôle central. Même si nous n'avons pas prouvé que c'est impossible, trouver une solution explicite dans les autres cas n'est en tous cas pas chose aisée et n'a pas été fait dans ce travail. Ce théorème mène à un résultat important dans le cas où les poids de (F.22) sont choisis spécifiquement. En effet, dans le cas particuler où les poids sont tous égaux à 1, l'erreur inverse est $||\Gamma_k||_1$, où Γ_k est la projection de l'opposé du gradient sur l'ensemble admissible, c'est-à-dire un vecteur dont les composantes sont définies par

$$\begin{aligned}
[\Gamma_k]_j &= [\text{Proj}_{\mathcal{F}}(x_k - \nabla_x f(x_k)) - x_k]_j \\
&= \begin{cases}
[-\nabla_x f(x_k)]_j & \text{si} \begin{cases} [\nabla_x f(x_k)]_j > 0 \text{ et } |[\nabla_x f(x_k)]_j| \leq |[l]_j - [x_k]_j|, \\ [\nabla_x f(x_k)]_j < 0 \text{ et } |[\nabla_x f(x_k)]_j| \leq |[u]_j - [x_k]_j|, \end{cases} \\
[l]_j - [x_k]_j & \text{si } [\nabla_x f(x_k)]_j > 0 \text{ et } |[\nabla_x f(x_k)]_j| > |[l]_j - [x_k]_j|, \\
[u]_j - [x_k]_j & \text{si } [\nabla_x f(x_k)]_j < 0 \text{ et } |[\nabla_x f(x_k)]_j| > |[u]_j - [x_k]_j|.
\end{cases}
\end{aligned}$$

Dans un cas plus général, lorsque $\alpha_{lu} \stackrel{def}{=} \alpha_l = \alpha_u$, l'erreur inverse vaut $||\Gamma_k(\alpha_g, \alpha_{lu})||_1$, où

$$\Gamma_k(\alpha_g, \alpha_{lu}) \stackrel{def}{=} \alpha_{lu} \left(\text{Proj}_{\mathcal{F}}(x_k - \frac{\alpha_g}{\alpha_{lu}} \nabla_x f(x_k)) - x_k \right), \tag{F.24}$$

ce qui est formalisé dans le Corallaire 3.2.3. Nous sommes aussi intéressés par le fait de trouver une solution explicite au second problème d'erreur inverse (F.23)

$$\chi_k^{in} = \min_{y \in \mathcal{Y}_k} ||\alpha_g |\Delta g| + \alpha_l |\Delta l| + \alpha_u |\Delta u|\,||_{glu}.$$

Dans ce cas, si une norme p est choisie pour $||.||_{glu}$, avec $1 \leq p \leq \infty$, alors il est possible de trouver une valeur explicite de $\chi_k^{in,p}$, comme dans le cas précédent, ce qui est fait dans les Théorèmes 3.2.4 et 3.2.5. Notons qu'une conséquence directe de ces résultats est que dans le cas particulier où $p = 1$, nous avons $\chi_k^{in,1} = \chi_k^{out,1}$. De plus, lorsque $\alpha_l = \alpha_u$ ou lorsque tous les poids de F.23 sont égaux à un, ces résulats impliquent, comme dans le cas précédent que l'erreur inverse est la norme p de $\Gamma_k(\alpha_g, \alpha_{lu})$.

Nous pouvons aussi nous intéresser à une autre mesure de criticalité définie par

$$\chi_k^{tr} \stackrel{def}{=} \chi^{tr}(x_k) = \left| \min_{\substack{x_k + d \in \mathcal{F} \\ ||d||_\infty \leq 1}} \nabla_x f(x_k)^T d \right|. \tag{F.25}$$

Cette mesure a pour avantage d'être une approximation au premier ordre de la décroissance qui peut être obtenue dans la direction de plus forte pente (voir Conn et al. (1993)) et certains algorithmes, comme les méthodes de région de confiance par exemple, sont basés sur le fait d'obtenir une décroissance du même ordre que la décroissance de Cauchy. En conséquence, cette mesure de criticalité peut être particulièrement intéressante de ce point de vue. Néanmoins, χ_k^{tr} ne peut pas être liée aux erreurs inverse aussi simplement que $||\Gamma_k||$. Nous prouvons même dans le Lemme 3.2.7 que χ_k^{tr} n'es pas une erreur inverse, pour aucune norme produit. En conclusion, malgré le fait que χ_k^{tr} ait des propriétés intéressantes du point de vue de certains algorithmes spécifiques, elle n'est pas recommandable du point de vue de l'analyse d'erreur inverse.

Par ailleurs, le problème d'erreur inverse est de trouver la plus petite distance entre le problème original que nous souhaitons résoudre et le plus proche des problèmes perturbés que nous avons déjà résolus à l'itération k. Cette distance est traditionnellement mesurée par le biais d'une norme produit définie sur l'espace des perturbations Δg, Δl et Δu. Cependant, nous pourrions voir le problème autrement : au lieu de chercher la plus petite norme possible de Δg, Δl et Δu, nous pourrions prendre la norme des plus petits Δg, Δl et Δu. Cette seconde approche nous amène à considérer le problème sous l'angle de l'optimisation multicritères(voir Ehrgott (2005)) et, fait surprenant, cela ne donne pas nécessairement les mêmes résultats qu'avec l'approche classique. Plus précisément, nous montrons que toutes les solutions trouvées en minimisant la norme des perturbation (en utilisant χ_k^{out}) peuvent être atteintes par le biais de l'optimisation multicritères, mais que le contraire est faux.

Pour terminer, nous montrons que les mesures définies dans le chapitre satisfont toutes les propriétés requises pour démontrer la convergence de l'algorithme RMTR$_\infty$.

F.4 Expérimentations numériques

Dans ce chapitre, nous définissons dans un premier temps un algorithme pratique qui est utilisé pour sélectionner un ensemble de valeurs efficaces pour les paramètres de notre méthode. Cette version par défaut de l'algorithme est ensuite comparée à d'autres algorithmes classiques. Nous comparons enfin les différentes mesures de criticalité dont nous avons discuté dans le chapitre précédent.

F.4.1 Un algorithme pratique

Notre description de l'algorithme a laissé ouverts un certain nombre de choix pratiques. Le but de cette section est de fournir les détails manquants pour réaliser l'implémentation particulière utilisée pour les expérimentations numériques de ce chapitre. Ces options sont bien sûr liées à notre intérêt pour les problèmes discrétisés, où différents niveaux représentent différentes grilles de discrétisation, de la plus grossière à la plus fine.

F.4.2 Tests numériques

L'algorithme RMTR_∞ décrit précédemment a été codé en FORTRAN 95 par Dimitri Tomanos et toutes les expérimentations suivantes ont été réalisées sur un PC à processeur simple de 3 Ghz avec 2 Gbytes de mémoire RAM.

F.4.2.1 Problèmes tests

Nous avons considéré une série de problèmes de minimisation dans des espaces de dimension infinie et impliquant des opérateurs différentiels. Ces problèmes sont détaillés dans l'Annexe B. Les opérateurs différentiels sont discrétisés sur une succession de grilles régulières de manière à ce que la grille grossière de niveau $i-1$ soit définie en prenant un point sur 2 de la grille de niveau i : le rapport entre les pas de grille de deux niveaux successifs pour chaque direction de coordonnées est ainsi de 2. Les opérateurs de transfert de grille P_i sont définis de la même façon que dans le cas des multigrilles géométriques classiques, en utilisant les opérateurs d'interpolation. Les opérateurs de restriction R_i sont choisis tels que (2.6) soit vérifié.

Toutes les expérimentations discutées ci-après considèrent la solution d'un problème sur la grille la plus fine dont la taille, ainsi que les autres caractéristiques, peuvent être trouvées dans le tableau F.1. Les algorithmes sont arrêtés dès que la mesure de criticalité χ^{tr} du niveau le plus fin est inférieure à 10^{-3} pour tous les cas tests.

Notre stratégie de test, qui est discutée dans les paragraphes suivants, consiste dans un premier temps à établir une valeur par défaut adéquate pour les paramètres algorithmiques et dans un second temps, à comparer la méthode avec d'autres approches de référence.

F.4.2.2 Détermination de paramètres par défaut efficaces

Etant donné le nombre relativement important de paramètres de notre méthode, une discussion détaillée sur toutes les combinaisons possibles sort du cadre de cette section. Nous avons dès lors adopté la démarche suivante. Premièrement, nous avons fixé les paramètres pour lesquels un consensus est communément établi, à savoir

Nom du problème	n_r	r	Commentaire
DNT	511	8	1-D, quadratique
P2D	1046529	9	2-D, quadratique
P3D	250047	5	3-D, quadratique
DEPT	1046529	9	2-D, quadratique, (Minpack 2)
DPJB	1046529	9	2-D, quadratique, avec contraintes de bornes, (Minpack 2)
DODC	65025	7	2-D, convexe, (Minpack 2)
MINS-SB	1046529	9	2-D, convexe, conditions limites lisses.
MINS-OB	65025	7	2-D, convexe, conditions limites oscillantes.
MINS-DMSA	65025	7	2-D, convexe, (Minpack 2)
IGNISC	65025	7	2-D, convexe
DSSC	1046529	9	2-D, convexe, (Minpack 2)
BRATU	1046529	9	2-D, convexe, (Minpack 2)
MINS-BC	65025	7	2-D, convexe, avec contraintes de bornes
MEMBR	393984	9	2-D, convexe, frontière libre, avec contraintes de bornes
NCCS	130050	7	2-D, nonconvexe, conditions limites lisses.
NCCO	130050	7	2-D, nonconvexe, conditions limites oscillantes.
MOREBV	1046529	9	2-D, nonconvexe

Table F.1: Caractéristiques des problèmes tests

les paramètres de région de confiance η_1, η_2, γ_1 et γ_2 qui sont déterminés en (4.8), en accord avec Conn et al. (2000) et Gould et al. (2005). Le rayon de région de confiance $\Delta_{i,0}$ est fixé à 1, comme suggéré dans la section 17.2 de la première de ces références. Une seconde classe de paramètres ayant une influence négligeable sur les résultats numériques a ensuite été isolée. Ceux-ci sont: le choix d'activer le mécanisme de recherche linéaire (nous autorisons le backtracking si l'itération initiale est ratée et au plus un pas de recherche linéaire si l'itération est réussie et qu'elle détermine une direction "gradient-related" avec $\epsilon_{\text{gr}} = 0.01$), les paramètres ϵ_H et η_H de la méthode d'évaluation du Hessien (nous choisissons $\eta_H = 0.5$ et $\epsilon_H = 0.15$) et le degré d'interpolation de l'opérateur de prolongation (l'interpolation linéaire est utilisée dans le cas d'itérations récursives, l'interpolation cubique est utilisée lorsque la solution à un niveau grossier est prolongée comme point de départ de la grille fine suivante). Les paramètres restants de l'algorithme sont fondamentaux ou ont une influence significative sur les performances de la méthode. Nous consacrons le reste de cette discussion au choix d'une valeur optimale pour ces derniers.

Nous commençons par déterminer la combinaison optimale de ces paramètres. Nous avons à cet effet testé un grand nombre (192) de combinaisons possibles sur notre panel de 17 cas test et nous avons illustré les résultats de ces 3264 runs sur un graphique à nuage de points représentant le nombre d'évaluations de la fonction coût en fonction du temps CPU. Plus précisément, nous avons pris soin de mettre ces deux grandeurs à l'échelle en les divisant, pour chaque cas test, par la meilleure valeur obtenue sur le problème et ce, pour toutes les variantes algorithmiques. En omettant les variantes algorithmiques pour lesquelles le temps CPU dépassait 1000 secondes sur au moins un problème, nous avons ensuite tracé la moyenne de ces grandeurs réduites pour chaque variante algorithmique et ce, sur tous les cas test. Dans le premier de ces graphiques (Figures 4.2 and 4.3), les variantes pour lesquelles le modèle grossier

de Galerkin est choisi pour les itérations récursives sont représentées par des triangles alors que les variantes pour lesquelles le modèle du second ordre (4.6) est choisi sont représentées par des étoiles.

Nous observons une dispersion substantielle des résultats avec des options jusqu'à 15 fois moins performantes que d'autres. Les cas les plus défavorables (points situés dans le coin supérieur droit) correspondent à des combinaisons du modèle quadratique (4.6) avec un seul cycle de lissage et des petites valeurs de κ_χ. Le choix du modèle de Galerkin est manifestement le meilleur. Ceci est principalement dû au coût numérique moins élevé de ce modèle car il ne nécessite pas, contrairement au modèle du second ordre, l'évaluation d'une fonction et d'une matrice hessienne suivie d'une mise à jour matricielle pour chaque modèle. Même pour les problèmes tests pour lesquels le modèle (4.6) se révèle meilleur en termes de nombre d'itérations, l'avantage est perdu en temps CPU. Au vu de ces résultats, nous sélectionnons donc le modèle de Galerkin comme modèle par défaut et nous nous limiterons à ce cas pour la suite de l'étude.

Nous considérons à présent le nombre de cycles de lissage effectués à chaque itération de Taylor (à un niveau $i > 0$) et nous présentons nos résultats sur la Figure 4.4. Toutes les variantes algorithmiques (avec le modèle grossier de Galerkin) sont à nouveau représentées sous la même forme que pour la Figure 4.2, où différents symboles sont utilisés pour représenter les variantes ayant des nombres différents de cycles de lissage.

Une propriété importante de cette option est que le nombre d'évaluations de fonction diminue à mesure que le nombre de cycles augmente, car une évaluation est mieux exploitée si plus de cycles sont réalisés consécutivement. Cette corrélation se vérifie jusqu'à un certain niveau (probablement dépendant de la quadraticité de la fonction objectif) au-delà duquel tout cycle additionnel se révèle inefficace. Il est plus difficile d'établir une corrélation en terme de temps CPU, même si nos résultats indiquent choisir un petit nombre de cycles de lissage est rarement la bonne option. Le bon choix semble se trouver entre 2 et 7 cycles.

Choisir de bonnes valeurs pour κ_χ n'est pas aisé. Nous avons considéré 4 valeurs possibles (1/2, 1/4, 1/8, 1/16). Nous observons d'abord que pour les valeurs de κ_χ largement supérieures à 1/2, le caractère multigrille du problème n'est pas pleinement exploité car les itérations récursives deviennent peu fréquentes. A l'opposé, les valeurs inférieures à 1/16 sont également problématiques car l'apport des itérations récursives en terme d'optimisation devient négligeable et ce, bien que cette stratégie soit plus proche de la nature récursive non-conditionnelle des algorithmes multigrilles pour la résolution de systèmes linéaires. A l'issue de nos tests, la meilleure valeur obtenue est $\kappa_\chi = 1/2$ ou $\kappa_\chi = 1/4$, avec un léger avantage pour cette dernière (voir la Figure 4.5, construite sur le même principe que les figures précédentes).

Nous abordons à présent l'impact du type de cycle sur la performance, tel qu'illustré sur la Figure 4.6. Nous remarquons qu'indépendamment des autres paramètres algorithmiques, la performance obtenue pour les trois types de cycle considérés est excellente. En particulier, ceci indique que la méthode qui consiste à adapter automatiquement le nombre d'itérations au problème en fonction d'un critère d'arrêt dépendant uniquement du niveau courant est raisonnablement efficace. La méthode est néanmoins légèrement plus complexe et l'on pourrait en pratique souvent préférer des simples cycles en forme V.

Enfin, la Figure 4.7 illustre l'effet du choix du seuil de criticalité grossier entre (4.9)

(pas de min) et (4.10) (min). Elle indique que (4.10) est en général préférable bien que la performance reste mitigée.

En conclusion de cette analyse, nous avons décidé d'utiliser par défaut, le modèle de Galerkin, 7 cycles de lissage par itération de Taylor, une valeur de $\kappa_\chi = 1/4$, des itérations en forme V et le critère d'arrêt (4.10).

F.4.2.3 Performances de RMTR$_\infty$

Nous analysons à présent, sur notre série de 17 cas tests, la performance de l'algorithme récursif de région de confiance obtenu, en comparaison avec d'autres approches. Cette analyse est conduite en comparant quatre algorithmes :

- La méthode classique au niveau le plus fin ("all of finest": AF) qui est un algorithme de région de confiance de Newton standard (avec PTCG comme solveur de sous-problème) appliqué au niveau le plus fin et sans recours à des calculs aux niveaux grossiers.

- La méthode de raffinement de maillage ("mesh refinement": MR) où les problèmes discrétisés sont résolus successivement du niveau le plus grossier (niveau 0) au niveau le plus fin (niveau r) en utilisant la même méthode de région de confiance de Newton standard et où le point de départ au niveau $i+1$ est obtenu en prolongeant la solution obtenue au niveau i (en utilisant P_{i+1}).

- La méthode multiniveaux au départ du niveau le plus fin ("multilevel on finest": MF) où l'algorithme RMTR$_\infty$ est appliqué directement sur le niveau le plus fin.

- La méthode totalement multiniveaux ("full multilevel": FM) où l'algorithme RMTR$_\infty$ est appliqué successivement sur des discrétisations de plus en plus fines (des plus grossières aux plus fines) et où le point de départ au niveau $i+1$ est obtenu en prolongeant la solution obtenue au niveau i (en utilisant P_{i+1}).

Le profil de performance en temps CPU (voir Dolan and Moré, 2002) est présenté sur la Figure F.1 pour tous les cas tests et les quatre variantes. L'axe des ordonnées de ce profil représente la fraction du nombre total des problèmes que l'algorithme testé résout dans un multiple (représenté sur l'axe des abscisses) du meilleur temps CPU. En conséquence, la ligne en haut à gauche du graphe représente la variante la plus efficace, tandis que la ligne en haut à droite correspond à la plus fiable. La première comclusion est que la variante totalement multiniveaux (FM) est clairement plus performante que les autres, autant en termes d'efficacité que de fiabilité. La seconde observation est que la variante AF est, comme attendu, de loin la moins bonne. Les deux variantes restantes sont étonnament proches, et l'usage d'itérations récursives sur le niveau le plus fin a l'air d'avoir une efficacité similaire au fait d'optimiser sur des grilles de plus en plus fines. Ces observations sont confirmées par une analyse détaillée des résultats numériques complets présentés dans l'Annexe C.

Les conclusions du paragraphe précédant ne disent pas tout, et nous pourrions être intéressés de voir si le gain en performance est effectivement le résultat d'un gain en efficacité de type multigrilles. Pour répondre à cette question, nous comparons

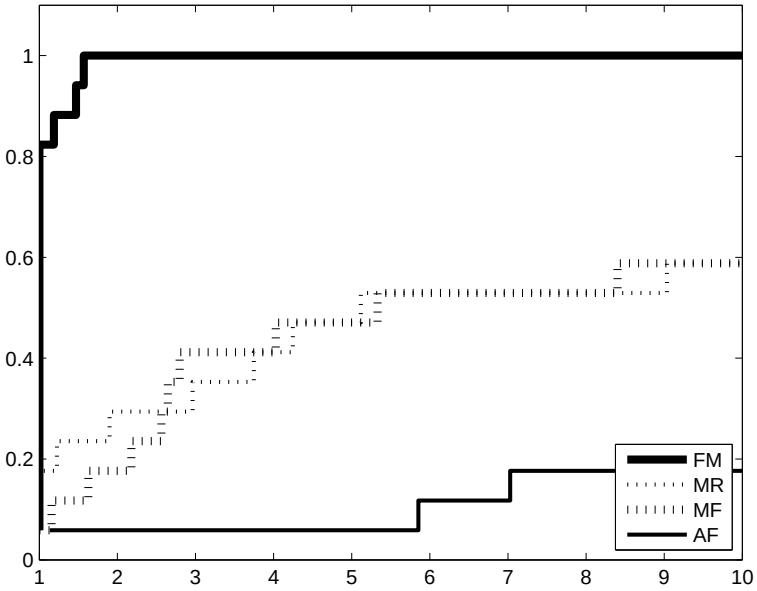


Figure F.1: Profil de performance pour le temps CPR avec les variantes AF, MF, MR et FM (17 problèmes tests).

à présent de façon détaillée les variantes MR et FM sur trois problèmes tests spécifiques sans contraintes (P2D, MINS-SB et NCCS), que nous considérons comme représentatifs des différentes classes de problèmes mentionnés dans la Table F.1.

La performance des algorithmes est illustrée pour chacun de ces problèmes par une figure montrant l'historique de la mesure de criticalité à l'échelle définie à la Section 4.1.7 lorsque les algorithmes MR (fine ligne) et FM (ligne en gras) sont utilisés. Dans ces figures, la ligne en tirets représente l'augmentation de la mesure de criticalité à l'échelle lorsqu'une solution est prolongée pendant l'application d'un processus de raffinement de maillage. De plus, et parce que les itérations aux niveaux grossiers sont considérablement moins coûteuses que celles réalisées au niveaux plus fins, nous avons choisi de représenter ces historiques comme des fonctions du *nombre équivalent d'iérations au niveau le plus fin*, donné par

$$q = \sum_{i=0}^{r} q_i \left(\frac{n_i}{n_r} \right), \tag{F.26}$$

où q_i est le nombre d'itérations au niveau i.

Nous considérons premièrement le problème de minimisation quadratique P2D. Comme ce problème est équivalent à résoudre un système linéaire d'équations, nous nous attendons à ce que l'algorithme FM exhibe un comportement de type multigrilles. En regardant la Figure 4.9, nous voyons que c'est effectivement le cas. Notons que FM est considérablement plus efficace que MR (d'un facteur proche de 100). Ce dernier résultat confirme que globalisation par le biais de la région de confiance n'altère pas la célèbre efficacité des méthodes multigrilles pour ce type de problèmes. Notons aussi que l'augmentation significative de la mesure de criticalité à l'échelle

lorsqu'une solution grossière est prolongée pour devenir un point de départ pour le niveau plus fin est due au fait que les composant oscillants de l'erreur ne peuvent pas être représentés aux niveaux grossiers et ne peuvent donc pas être réduits à ces niveaux.

Les mêmes conclusions semblent s'appliquer lorsque nous considérons les Figures 4.10 et 4.11, où les mêmes algorithmes ont été testés sur MINS-SB et NCCS, respectivement. Ceci est remarquable puisque les problèmes sont à présent plus généraux et ne correspondent plus à des systèmes linéaires d'équations (MINS-SB est non-quadratique) ou à des problèmes elliptiques (NCCS est non-convexe).

Une caractéristique importante de l'algorithme de région de confiance classique est que sa convergence est accélérée lorsque la région de confiance devient inactive (car l'algorithme se réduit alors à la méthode de Newton et converge donc quadratiquement à condition que le modèle de Taylor du second ordre (2.16) soit choisi). Les itérations auxquelles la région de confiance est active ont été indiquées dans les figures ci-dessus par un petit cercle (notons qu'elles correspondent souvent à une décroissance non-monotone de la mesure de criticalité à l'échelle). Nous observons que de telles itérations ne se produisent pas pour MR et FM sur P2D, mais aussi que la convergence s'accélère pour toutes les méthodes dès que la région de confiance devient inactive, même si ce taux est au plus linéaire pour les méthodes multiniveaux.

Nous évaluons finalement l'algorithme RMTR_∞ sur les problèmes avec contraintes de bornes suivants: DPJB, MINS-BC et MEMBR. Les résultats pour ces problèmes sont présentés sur les Figures 4.12 à 4.14

Nous notons tout d'abord que sur les trois problèmes, un gain en temps CPU d'un facteur excédant 10 est typiquement obtenu lorsqu'on considère la variante multigrilles. Ensuite, observons que la performance relative des deux algorithmes considérés est assez similaire à celle analysée pour les problèmes non-contraints, au moins pour DPJB et MEMBR. Pour ce dernier problème, la figure indique que des gains d'efficacité supplémentaires pourraient être obtenus par un réglage plus minutieux de la précision du critère d'arrêt aux niveaux 5, 6 et 7. A nouveau, la contrainte de la région de confiance est principalement inactive sur ces exemples. Ceci est à l'opposé de ce que nous constatons pour MINS-SB où elle joue un rôle important, sauf à l'asymptote (comme attendu par la théorie des régions de confiance).

Pour terminer, dans la version complète de la thèse en anglais, une comparaison numérique est réalisée entre deux mesures de criticalités définies au Chapitre 3.

F.4.3 Conclusion

Nous avons présenté une implémentation de l'algorithme multiniveaux de région de confiance pour les problèmes avec contraintes de bornes RMTR_∞, ainsi qu'une expérimentation numérique sur des problèmes tests multiniveaux. Un choix approprié des paramètres de l'algorithme a été identifié pour ces problèmes, menant a un bon compromis entre efficacité et fiabilité. L'algorithme par défaut résultant a été comparé à d'autres techniques d'optimisation, telles que le raffinement de maillage et la solution directe obtenue sur le niveau le plus fin.

Nous sommes conscients que plus d'expérimentations numériques sur un plus large spectre d'applications sont nécessaires, cependant l'expérience numérique obtenue jusqu'ici est très encourageante. D'autres comparaisons avec d'autres propositions, telles que celles de Kornhuber (1994,1996), seraient aussi désirables.

F.5 Conclusion

Nous avons tout d'abord construit un algorithme de région de confiance multiniveaux en norme infinie qui gère l'information multiniveaux de problèmes provenant d'une discrétisation, tout autant que la possible présence de contraintes de bornes. Cet algorithme a été inspiré par la méthode décrite par Gratton, Sartenaer and Toint (2008*b*), et adapté à l'usage de la norme infinie dans la définition de la région de confiance ainsi qu'au traitement des contraintes de bornes. Il suit l'idée des méthodes multigrilles pour la résolution de systèmes linéaires afin d'éliminer progressivement les basses et hautes fréquences de l'erreur en combinant l'usage des niveaux de discrétisation grossiers et celui d'une technique de lissage. Les caractéristiques principales du nouvel algorithme ont été détaillées, en particulier la manière dont les discrétisation grossières sont exploitées pour construire un modèle de la fonction objectif qui soit moins coûteux que l'approximation de Taylor (généralement utilisée par les méthodes de région de confiance), l'introduction d'une condition de descente qui indique si l'usage d'un modèle grossier peut être bénéfique ou pas, ou encore les différentes façons dont sont traitées les contraintes suivant qu'elles proviennent des contraintes de bornes du problème original ou de la méthode de région de confiance elle-même.

Par la suite, un algorithme général a été présenté formellement, laissant un certain nombre de choix ouverts, tels que les solveurs utilisés pour le calcul des pas non-grossiers (aussi appellés pas de Taylor), la définition des opérateurs de transfert qui permettent le passage d'information à travers les différents niveaux de discrétisation, ou encore la définition de la mesure de criticalité utilisée comme critère d'arrêt. Cet algorithme général a été prouvé globallement convergent, ce qui signifie qu'il converge à partir de tout point de départ (admissible), sous de raisonables hypothèses.

De plus, nous avons étendu la théorie de Conn et al. (1993) au sujet de l'identification des contraintes actives, dans le sens où nous avons prouvé que toutes les méthodes de région de confiance pour la résolution de problèmes d'optimisation soumis à des contraintes convexes dont le solveur interne respecte une condition de décroissance suffisante du modèle (la condition de Cauchy modifiée) identifient l'ensemble complet des contraintes actives de la solution après un nombre fini d'itérations. Nous prouvons aussi plus tard que, dans le cadre des contraintes de bornes, le lisseur qui est utilisé pour calculer les pas non-récursifs à tous les niveaux sauf le plus grossier respecte cette condition de Cauchy modifiée. En outre, nous pouvons forcer le gradient conjugué projeté tronqué (PTCG) à converger exactement au niveau le plus grossier. En conséquence, l'ensemble correct des contraintes actives est identifié en un nombre fini de pas par une adaptation de notre algorithme de région de confiance multiniveaux pour laquelle les contraintes actives sont uniquement identifiées par les pas de Taylor tandis que les contraintes qui sont actives juste avant qu'un pas récursif ne soit calculé sont gelées pour le calcul de ce pas.

Dans le second chapitre, nous nous sommes interessés au fait de trouver un critère d'arrêt pour notre algorithme qui soit adapté à la résolution de problèmes nonlinéaires avec contraintes de bornes. Plus important, et parce que notre intérêt réside dans les problèmes discrétisés, nous avons recherché un critère d'arrêt qui soit adapté au cas où des incertitudes sur le problème (comme des erreurs de discrétisation) sont connues. Pour cette raison, et confortés par l'étendue de la théorie qui existe dans le cas linéaire, nous avons suivi une approche d'analyse de l'erreur inverse. Cela consiste traditionnellement à supposer que l'approximation courante de la solution du

problème original est la solution exacte d'un problème proche de celui-ci et à mesurer la distance entre ces deux problèmes dans une norme appropriée. L'application de cette technique à notre cas a mené, pour certaines définitions spécifiques de la norme, à un critère d'arrêt bien connu pour l'optimisation nonlinéaire avec contraintes de bornes, à savoir la norme de la projection de l'opposé du gradient sur l'ensemble admissible. De plus, nous avons prouvé qu'un second critère d'arrêt, souvent utilisé dans les méthodes de région de confiance, ne correspond pas à une erreur inverse, quelle que soit la norme choisie. D'un autre côté, ce second critère fait s'arrêter l'algorithme lorsque plus aucune décroissance significative ne peut être atteinte sur un modèle du premier ordre de la fonction objectif, ce qui peut s'avérer être un avantage dans certaines situations. Malgré leur nombreuses différences, il a été prouvé que ces deux critères respectent toutes les conditions requises pour la convergence de l'algorithme de région de confiance multiniveaux. De plus, nous avons prouvé que l'algorithme général est indépendant de la taille du pas de discrétisation (mesh-independant) pour une classe précise de problèmes, à condition que le second critère d'arrêt (conçu pour les méthodes de région de confiance) soit choisi.

Dans notre contexte (où il s'agit de trouver un point critique du premier ordre d'un problème d'optimisation nonlinéaire soumis à des contraintes de bornes), la norme utilisée dans l'analyse traditionnelle de l'erreur inverse prend en compte la distance entre les deux gradient et entre les deux ensembles admissibles. Il est possible d'agir sur des poids afin d'insister plus sur la réduction de l'erreur sur le gradient ou sur les bornes, par exemple en accord avec les incertitudes que nous pouvons connaitre sur ces quantités. Néanmoins, nous pouvons aussi être intéressés par le fait d'imposer que la distance entre les gradient et celle entre les ensembles admissibles soient toutes les deux indépendamment réduites (par exemple plus petites que leur incertitudes respectives). Ce point de vue nous a menés à considérer le problème du calcul de l'erreur inverse comme un problème d'optimisation multicritères. Nous avons prouvé que certaines solutions du problème d'erreur inverse lorsqu'il est vu sous l'angle de l'optimisation multicritères peuvent être impossibles à atteindre par l'approche habituelle (avec l'usage d'une norme). En d'autres mots, définir un critère d'arrêt inspiré par la version multicritères de l'erreur inverse pourrait mener à arrêter l'algorithme sur une solution approchée impossible à atteindre par un critère d'arrêt traditionnel. En conséquence, nous pourrions réfléchir à explorer la pertinence de ces solutions dans des situations concrètes.

La dernière partie de la thèse a été consacrée aux expérimentations numériques. Nous avons tout d'abord spécifié un algorithme concret, où les itérations de Taylor et récursives sont alternées, où une technique de lissage est utilisée pour calculer les pas aux itérations de Taylor à tous les niveaux sauf le plus grossier tandis que l'algorithme PTCG est utilisé pour le calcul du pas au niveau restant, le plus grossier (notons que l'algorithme général autorise l'utilisation de l'algorithme PTCG à tous les niveaux). Des implémentations concrètes ont été testées pour différents paramètres algorithmiques, tels que les opérateurs de transfert, la définition du modèle grossier ou la valeur de la constante dans la condition de descente, par exemple. Nous avons aussi expliqué brièvement certaines charactéristiques de l'algorithme telles que la façon dont les matrices hessiennes sont calculées ou les circonstances dans lesquelles une recherche linéaire est autorisée. Nous avons choisi la mesure de criticalité conçue pour les algorithmes de région de confiance comme critère d'arrêt pour ces tests numériques.

Cependant, nos expérimentations ont montré que les deux mesures peuvent se comporter très différemment lors du processus itératif pour certains problèmes.

Toutes les expérimentations ont été réalisées sur un panel de problèmes tests représentatif. La première partie de nos tests numériques a été contrée sur la recherche d'une combinaison optimale des paramètres de la méthode afin de définir un ensemble raisonnable de valeurs par défaut. Nous retenons qu'un modèle grossier du second ordre de type Galerkin est certainement recommandable, et que le meilleur nombre de cycles de lissage décroît lorsque la nonlinéarité du problème augmente. Nous avons ensuite utilisé les valeurs sélectionnées pour comparer l'algorithme de région de confiance multiniveaux à des méthodes concurrentes dans ce domaine. Ces comparaisons ont montré un avantage significatif pour notre méthode, autant en termes d'efficacité que de robustesse.

Ces résultats numériques sont réellement encourageants et font croître l'intérêt pour le développement des méthodes de cette sorte. En particulier, nous pensons à l'extension de l'algorithme au traitement de contraintes générales, ce qui pourrait être géré, dans un premier temps, en introduisant une fonction de pénalité dans le cadre d'une méthode de Lagrangien augmenté. De plus, l'algorithme multiniveaux que nous avons présenté ici est encore basé sur les ingrédients des méthodes multigrilles géométriques, et il pourrait être intéressant de développer une adaptation de la méthode basée sur les techniques algébriques. Néanmoins, tout comme pour les méthodes multigrilles algébriques pour la résolution de systèmes linéaires, le calcul de nouveaux opérateurs de transfert à chaque itération est extrêmement coûteux et nous devrions penser à des moyens de contourner ce problème.

En conclusion, cette thèse s'inscrit dans l'intérêt croissant pour les méthodes multiniveaux au sein de la communauté d'optimisation nonlinéaire. En effet, cette communauté est de plus en plus confrontée à des problèmes de dimension infinie comportant des opérateurs d'intégration et/ou provenant d'équations différentielles partielles où une hiérarchie de fonctions de coût et de contraintes est naturellement disponible. Dans ce cadre, les résultats présentés ici constituent une nouvelle manifestation de ce que dès que la nature sous-jascente du problème (ici, l'aspect discrétisé) peut être prise en compte, des algorithmes extraordinairement efficaces peuvent être conçus. De plus, des critères d'arrêts significatifs peuvent être définis pour ces problèmes lorsque des incertitudes dues à leur nature sont connues.

Bibliography

B. M. Averick and J. J. Moré. The Minpack-2 test problem collection. Technical Report ANL/MCS-TM-157, Mathematics and Computer Science, Argonne National Laboratory, Argonne, Illinois, USA, 1991.

N. S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *Computational Mathematics and Mathematical Physics*, **6**(5), 101–135, 1966.

F. Bastin, C. Cirillo, and Ph. L. Toint. An adaptive Monte-Carlo algorithm for computing mixed logit estimators. *Computational Management Science*, **3**(1), 55–80, 2006*a*.

F. Bastin, C. Cirillo, and Ph. L. Toint. Application of an adaptive Monte-Carlo algorithm to mixed logit estimation. *Transportation Research B*, **40**(7), 577–593, 2006*b*.

F. Bastin, V. Malmedy, M. Mouffe, Ph. L. Toint, and D. Tomanos. A retrospective trust-region method for unconstrained optimization. *Mathematical Programming, Series A*, **(to appear)**, 2009.

A. Borzi and K. Kunisch. A globalisation strategy for the multigrid solution of elliptic optimal control problems. *Optimization Methods and Software*, **21**(3), 445–459, 2006.

A. Brandt. Multi-level adaptative solutions to boundary value problems. *Mathematics of Computation*, **31**(138), 333–390, 1977.

W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, USA, 2nd edn, 2000.

M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh. A multigrid approach for multi-dimensional scaling. Talk at the 12th Copper Mountain Conference on Multigrid Methods, 2005.

Françoise Chaitin-Chatelin and Valerie Fraysse. *Lectures on Finite Precision Computations*. SIAM, Philadelphia, USA, 1996.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, **25**(182), 433–460, 1988. See also same journal 26:764–767, 1989.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*. Number 17 *in* 'Springer Series in Computational Mathematics'. Springer Verlag, Heidelberg, Berlin, New York, 1992.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization. *Mathematical Programming, Series A*, **73**(1), 73–110, 1996.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 01 *in* 'MPS-SIAM Series on Optimization'. SIAM, Philadelphia, USA, 2000.

A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal on Optimization*, **3**(1), 164–221, 1993.

A. R. Conn, L. N. Vicente, and C. Visweswariah. Two-step algorithms for nonlinear optimization with structured applications. *SIAM Journal on Optimization*, **9**(4), 924–947, 1999.

A. J. Cox and N. J. Higham. Backward error bounds for constrained least squares problems. Technical Report No. 321, Manchester, England, 1998.

J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1983. Reprinted as *Classics in Applied Mathematics 16*, SIAM, Philadelphia, USA, 1996.

E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, **91**(2), 201–213, 2002.

M. Domorádová and Z. Dostál. Projector preconditioning for bound-constrained quadratic optimization. *Linear Algebra and its Applications*, **(to appear)**, 2007.

M. Ehrgott. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Heidelberg, Berlin, New York, second edn, 2005.

M. Emilianenko. A nonlinear energy-based multilevel quantization scheme. Talk at the 12th Copper Mountain Conference on Multigrid Methods, 2005.

R. P. Fedorenko. On the speed of convergence of an iteration process. *Computational Mathematics and Mathematical Physics*, **4**(3), 227–235, 1964.

M. Fisher. Minimization algorithms for variational data assimilation. *in* 'Recent Developments in Numerical Methods for Atmospheric Modelling', pp. 364–385, Reading, UK, 1998. European Center for Medium-Range Weather Forecasts.

E. Gelman and J. Mandel. On multilevel iterative methods for optimization problems. *Mathematical Programming*, **48**(1), 1–17, 1990.

E. M. Gertz. *Combination Trust-Region Line-Search Methods for Unconstrained Optimization*. PhD thesis, Department of Mathematics, University of California, San Diego, California, USA, 1999.

G. H. Golub and C. F. Van Loan. *Matrix computations*. North Oxford Academic, Oxford, UK, 1983.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, second edn, 1989.

N. I. M. Gould and S. Leyffer. An introduction to algorithms for nonlinear optimization. *Frontiers in Numerical Analysis*, pp. 109–197, 2003.

N. I. M. Gould, D. Orban, and Ph. L. Toint. Results from a numerical evaluation of lancelot b. Technical Report 02/09, Department of Mathematics, FUNDP - University of Namur, Namur, Belgium, 2002.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr, a constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, **29**(4), 373–394, 2003.

N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Sensitivity of trust-region algorithms on their parameters. *4OR, Quarterly Journal of the Italian, French and Belgian OR Societies*, **3**(3), 227–241, 2005.

S. Gratton, M. Mouffe, A. Sartenaer, Ph. L. Toint, and D. Tomanos. Numerical experience with a recursive trust-region method for multilevel nonlinear optimization. *Mathematical Programming, Series A*, **(to appear)**, 2009.

S. Gratton, M. Mouffe, Ph. L. Toint, and M. Weber-Mendonça. A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization. *IMA Journal of Numerical Analysis*, **28**(4), 827–861, 2008*a*.

S. Gratton, A. Sartenaer, and Ph. L. Toint. Numerical experience with a recursive trust-region method for multilevel nonlinear optimization. Technical Report 06/01, Department of Mathematics, FUNDP - University of Namur, Namur, Belgium, 2006*a*.

S. Gratton, A. Sartenaer, and Ph. L. Toint. Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization. *Journal of Computational and Applied Mathematics*, **24**(6), 676–692, 2006*b*.

S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, **19**(1), 414–444, 2008*b*.

W. W. Hager and H. Zhang. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. on Optimization*, **14**(4), 1043–1056, 2004.

W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. on Optimization*, **17**(2), 526–557, 2006.

N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, USA, 1996.

M. Hintermüller and L. N. Vicente. Space mapping for optimal control of partial differential equations. *SIAM Journal on Optimization*, **15**, 1002–1025, 2005.

C. T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, USA, 1999.

R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities I. *Numerische Mathematik*, **69**, 167–184, 1994.

R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities II. *Numerische Mathematik*, **72**, 481–499, 1996.

R. Kornhuber. Adaptive monotone multigrid methods for some non-smooth optimization problems. *in* R. Glowinski, J. Périaux, Z. Shi and O. Widlund, eds, 'Domain Decomposition Methods in Sciences and Engineering', pp. 177–191. J. Wiley and Sons, Chichester, England, 1997.

M. Lewis and S. G. Nash. Practical aspects of multiscale optimization methods for VLSICAD. *in* J. Cong and J. R. Shinnerl, eds, 'Multiscale Optimization and VLSI/CAD', pp. 265–291, Dordrecht, The Netherlands, 2002. Kluwer Academic Publishers.

M. Lewis and S. G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing*, **26**(6), 1811–1837, 2005.

C. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, **9**(4), 1100–1127, 1999.

J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, **4**(3), 553–572, 1983.

J. J. Moré and J. D. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, **20**, 286–307, 1994.

J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, **7**(1), 17–41, 1981.

S. G. Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, **14**, 99–116, 2000.

Yu. Nesterov. *Introductory Lectures on Convex Optimization*. Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

J. Nocedal and Y. Yuan. Combining trust region and line search techniques. *in* Y. Yuan, ed., 'Advances in Nonlinear Programming', pp. 153–176, Dordrecht, The Netherlands, 1998. Kluwer Academic Publishers.

S. Oh, A. Milstein, Ch. Bouman, and K. Webb. A general framework for nonlinear multigrid inversion. *IEEE Transactions on Image Processing*, **14**(1), 125–140, 2005.

J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, London, 1970.

M. J. D. Powell and Ph. L. Toint. On the estimation of sparse Hessian matrices. *SIAM Journal on Numerical Analysis*, **16**(6), 1060–1074, 1979.

J. L. Rigal and J. Gaches. On the compatibility of a given solution with the data of a linear system. *J. ACM*, **14**(3), 543–548, 1967.

S. M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, **21**(3), 513–528, 1996.

A. Shapiro. Monte Carlo sampling methods. *in* A. Shapiro and A. Ruszczyński, eds, 'Stochastic Programming', Vol. 10 of *Handbooks in Operations Research and Management Science*, pp. 353–425. Elsevier, Amsterdam, The Netherlands, 2003.

T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, **20**(3), 626–637, 1983.

Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. *in* I. S. Duff, ed., 'Sparse Matrices and Their Uses', pp. 57–88, London, 1981. Academic Press.

Ph. L. Toint. VE08AD, a routine for partially separable optimization with bounded variables. *Harwell Subroutine Library*, **2**, 1983.

Ph. L. Toint. VE10AD, a routine for large scale nonlinear least squares. *Harwell Subroutine Library*, **2**, 1987.

U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Elsevier, Amsterdam, The Netherlands, 2001.

L. Xu and J. Burke. ASTRAL: An active set ℓ_∞-trust-region algorithm for box-constrained optimization. Technical Report preprint, Department of Mathematics, University of Washington, Seattle, USA, 2007.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. Technical Report NAM-11, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, USA, 1994.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 78: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, **23**(4), 550–560, 1997.