Convergence properties of minimization algorithms for convex constraints using a structured trust region

by A.R. Conn[1], Nick Gould[2],
A. Sartenaer[3] and Ph.L. Toint[3]

Report 92/11(revision 2) September 11, 1995

[1] IBM T.J. Watson Research Center, P.O.Box 218, Yorktown Heights, NY 10598, USA
Email : arconn@watson.ibm.com

[2] Rutherford Appleton Laboratory, Chilton, Oxfordshire, England
Email : nimg@directory.rl.ac.uk
Current reports available by anonymous ftp from the directory
"pub/reports" on camelot.cc.rl.ac.uk (internet 130.246.8.61)

[3] Department of Mathematics, Facultés Universitaires ND de la Paix, 61, rue de Bruxelles,
B-5000 Namur, Belgium, EC
Email : as@math.fundp.ac.be and pht@math.fundp.ac.be
Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be (internet 138.48.4.14)

Keywords : Trust region methods, structured problems, large-scale optimization, partial separability, convex constraints.

Convergence properties of minimization algorithms for convex constraints using a structured trust region

A.R. Conn Nick Gould A. Sartenaer Ph.L. Toint

September 11, 1995

Abstract

In this paper, we present a class of trust region algorithms for minimization problems within convex feasible regions, in which the structure of the problem is explicitly used in the definition of the trust region. This development is intended to reflect the possibility that some parts of the problem may be more accurately modelled than others, a common occurrence in large-scale nonlinear applications. After describing the structured trust region mechanism, we prove global convergence for all algorithms in our class.

1 Introduction

Trust region algorithms have enjoyed a long and successful history as tools for the solution of nonlinear, nonconvex, optimization problems. They have been studied and applied to unconstrained problems (see [7], [17], [25], [28], [29], [30], [31], [34], [35], [38]) and to problems involving various classes of constraints, including simple bounds ([6], [10], [11], [27], [32]), convex constraints ([2], [3], [14], [41]), and nonconvex ones ([5], [8], [16], [36], [44]). This long lasting interest is probably justified by the attractive combination of a solid convergence theory, a noted algorithmic robustness, the existence of numerically efficient implementations and an intuitively appealing motivation. The main idea behind trust region algorithms is that, if a nonlinear function (objective and/or constraints) is expensive to compute or difficult to handle explicitly, it should be replaced by a suitable *model*. This model is deemed to be trustworthy within a certain *trust region* around the current point. The trust region is defined by its shape and its *radius*. The minimization involving the difficult nonlinear function(s) is then replaced by a sequence of minimizations of the simpler model(s) within appropriate trust regions. The trust region radii are adjusted to reflect the agreement between the model and true functions as the process proceeds.

It is remarkable that, up to now, all algorithms that we are aware of use a *single* trust region radius to measure the degree of trustworthiness of the models employed, even if several

[0]This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No F49620-91-C-0079. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

different functions are involved. This choice is somewhat surprising if one admits that some of the modelled functions could be substantially "better behaved" than others in the same problem, as this implies that the region in which their models can be trusted might also be substantially larger. In this context, the unstructured trust region choice might be viewed as a conservative strategy ensuring that *all* models may be trusted in what amounts to a "safe minimal" region. While this strategy might be reasonable for small problems, where each involved function depends on all the problem's variables, it is clearly questionable for large-scale applications, where each of the problem's function typically depends only on a small number of variables. For instance, one might consider the minimization of an unconstrained objective function consisting of the sum of many quadratic and a few highly nonlinear terms, the latter involving a small subset of the variables. If a classical unstructured trust region algorithm, with a quadratic model, is used, the quadratic terms are perfectly modelled, but the steps that one can make at each iteration are (unnecessarily) limited by the highly nonlinear behaviour of a small subset of the variables.

It is the purpose of this paper to present and analyze a class of algorithms that use the problem's *structure* in the *definition* of the trust region, allowing large steps in directions in which the model has proved to be adequate while restricting the movement in directions where the model seems unreliable. To be more precise, we will consider the problem of minimizing a *partially separable* objective function subject to convex constraints; we will then use the decomposition of the objective function into element functions as the basis for our structured trust region definition. The choice of the partially separable structure, a concept introduced in [21], is motivated by the very general geometric nature of this structure and by the increasing recognition of its practical use (see [4], [9], [12], [13], [18], [19], [20], [22], [26], [39], [42], [43], amongst others). More significantly, partial separability provides a decomposition of the considered nonlinear function into a linear combination of smaller *element functions*, each of which may then be modelled separately (see [40]). It is then quite natural to assign one trust region radius per element function and to decide on its increase or decrease separately. Because different element functions typically involve different sets of variables, each *element trust region* only restricts the components of the step corresponding to its *elemental variables*.

An obvious approach is to use the norm-scaling matrices allowed in the theory for unstructured trust region methods ([10], for instance) to account for differences in model adequacy among elements when constructing the trust region. This would be satisfactory if the existing theory did not require that the scaling matrices be of uniformly bounded condition number. Unfortunately, it is easy to conceive of instances where this is a severe handicap. For example, it would prevent the trust region radius of a well-modelled (perhaps linear or quadratic) element from increasing to infinity while at the same time ensuring that that of a badly behaved nonlinear element function remains of modest size. Moreover, this strategy may well cause numerical difficulties when attempting to solve the trust region problem. In fact, as we will shortly see, additional algorithmic safeguards are important when simultaneously handling trust regions of vastly different sizes. Thus, we do not consider such an approach further in this paper.

Section 2 of the paper presents the problem in more detail and the new class of algorithms using the principle of structured trust regions. Global convergence for all algorithms in the class is proved in Section 3. We briefly discuss the identification of active constraints in Section 4. We examine in Section 5 some extensions of the results of the previous sections. We finally give some comments and perspectives in Section 6.

2 Structured trust region for partially separable problems

2.1 A structured model of the objective and the corresponding structured trust region

2.1.1 The problem

The problem we consider is that of minimizing a smooth objective function subject to convex constraints. That is, we wish to solve the problem

$$\underset{x \in X}{\text{minimize}} \ \ f(x), \tag{2.1}$$

where X is a closed convex subset of $\mathbf{R}^n$. We denote the Euclidean inner product on $\mathbf{R}^n$ by $\langle \cdot, \cdot \rangle$, and the associated ℓ_2-norm by $\|\cdot\|$. Given Y a closed convex subset of $\mathbf{R}^n$, we define the operator $P_Y(\cdot)$ to be the orthogonal projection onto Y. We now list our additional assumptions on (2.1).

AS.1 X has a non-empty interior.

AS.2 f is bounded below on X.

AS.3 f is partially separable, which means that

$$f(x) = \sum_{i=1}^{p} f_i(x) \tag{2.2}$$

and that, for each $i \in \{1, \ldots, p\}$, there exists a subspace $\mathcal{N}_i \neq \{0\}$ such that, for all $w \in \mathcal{N}_i$ and all $x \in X$,

$$f_i(x + w) = f_i(x). \tag{2.3}$$

AS.4 For each $i \in \{1, \ldots, p\}$, f_i is continuously differentiable in an open set containing X and its gradient is uniformly bounded on X.

Note that we admit the case where X is unbounded or even identical to $\mathbf{R}^n$ itself, in which case we obtain an unconstrained problem. In relation to the partial separability of the objective function, we also consider the *range subspace* (see [23]) associated with each element function f_i, which is defined as

$$\mathcal{R}_i \stackrel{\text{def}}{=} \mathcal{N}_i^{\perp}. \tag{2.4}$$

We are mostly interested in the case where the dimension of each $\mathcal{R}_i$ is small compared to n. A commonly occurring case is when each element function f_i only depends on a small subset of the problem's variables: $\mathcal{R}_i$ is then the subspace spanned by the vectors of the canonical basis corresponding to the variables that occur in f_i (the elemental variables). The range of the projection operator $P_{\mathcal{R}_i}(\cdot)$ is therefore of low dimensionality. The reader is referred to [12] for a more detailed introduction to partially separable functions.

We note that f is invariant for any translation in the subspace $(\sum_{i=1}^{p} \mathcal{R}_i)^{\perp}$. We may therefore restrict our attention to the case where

$$\sum_{i=1}^{p} \mathcal{R}_i = \mathbf{R}^n \tag{2.5}$$

without loss of generality.

2.1.2 The element models

The algorithm we have in mind is iterative and generates feasible iterates (in the sense that all iterates belong to X). At iteration k, we will associate a model $m_{i,k}$ with each element function f_i. This model, defined on $\mathcal{R}_i$ in a neighbourhood of the projection of the k-th iterate x_k on this subspace, is meant to approximate f_i for all x in the *element trust region*

$$B_{i,k} \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\| \leq \Delta_{i,k}\}, \tag{2.6}$$

where $\Delta_{i,k} > 0$ is the i-th trust region radius at iteration k and the norm $\|\cdot\|$ is chosen to be the usual Euclidean norm in order to simplify the exposition. In what follows, we will slightly abuse notation by writing $m_{i,k}(x)$ for an $x \in \mathbf{R}^n$, instead of the more complete $m_{i,k}(P_{\mathcal{R}_i}(x))$. We will furthermore assume that each model $m_{i,k}$ $(i \in \{1, \ldots, p\}, k = 0, 1, 2, \ldots)$ is differentiable and has Lipschitz continuous first derivatives on an open set containing $B_{i,k}$, and that

$$m_{i,k}(x_k) = f_i(x_k) \quad (i \in \{1, \ldots, p\}, k = 0, 1, 2, \ldots). \tag{2.7}$$

Moreover, we assume that $g_{i,k} \stackrel{\text{def}}{=} \nabla m_{i,k}(x_k) \in \mathcal{R}_i$ approximates $\nabla f_i(x_k) \in \mathcal{R}_i$ in the sense that, for all $i \in \{1, \ldots, p\}$ and all k,

$$\|e_{i,k}\| \leq \kappa_1 \Delta_{min,k}, \tag{2.8}$$

where $e_{i,k} \stackrel{\text{def}}{=} g_{i,k} - \nabla f_i(x_k)$, where $\kappa_1 > 0$ is a constant and where $\Delta_{min,k}$ is defined by

$$\Delta_{min,k} \stackrel{\text{def}}{=} \min_{i \in \{1,\ldots,p\}} \Delta_{i,k}. \tag{2.9}$$

Condition (2.8) is quite weak, as it merely requires that the first order information be reasonably accurate whenever some trust region radius is small (i. e. the corresponding model fits badly). Indeed, one expects the coherency of this first order behaviour to be of crucial importance in such cases. Further arguments supporting a choice similar to (2.8) for problems with convex constraints are presented in [14].

Amongst the most commonly used element models, linear or quadratic approximations are pre-eminent. One can, for instance, consider the quadratic model given by the first three terms of the element function Taylor series around the current iterate. Another popular choice is a quadratic model where the second derivative matrix is recurred using quasi-Newton formulae.

2.1.3 The overall model and trust region

With all the element models at hand, we are now in position to define the overall model at iteration k, denoted m_k, whose purpose is to approximate the overall objective function f in a neighbourhood of the current iterate x_k. From (2.2), it is natural to use the overall model

$$m_k(x) \stackrel{\text{def}}{=} \sum_{i=1}^{p} m_{i,k}(x) \tag{2.10}$$

for all x in the overall trust region defined by

$$B_k = \bigcap_{i\in\{1,\dots,p\}} B_{i,k}. \tag{2.11}$$

Indeed B_k is the intersection of all element trust regions, that is the region in which all element models may be trusted, irrespective of the additional limitation possibly imposed by the feasible set X.

Of course, the actual shape of the trust region B_k is determined by the choice of the Euclidean norm: it corresponds to the intersection of cylinders whose axis are aligned with the subspaces $\mathcal{N}_i$ and whose radii reflect the quality of the element models: large in subspaces where the element models predict the element function correctly and smaller in subspaces where the prediction is poorer. In practice, one might wish to choose other norms, such as the ℓ_∞-norm. In this case, and assuming that the subspaces $\mathcal{R}_i$ are spanned by subsets of the canonical basis vectors, the shape of the trust region is that of a box, the length of whose sides again reflects the quality of the element models. The extension of the theory to more general norms is considered in Section 5.4.

2.1.4 Curvature

We now follow [14] and [41] and define the *generalized Rayleigh quotient* of f at x along $s \neq 0$ by

$$\omega(f,x,s) \stackrel{\text{def}}{=} \frac{2}{\|s\|^2}[f(x+s) - f(x) - \langle \nabla f(x), s\rangle]. \tag{2.12}$$

Obviously, this definition is valid only if s is such that $x+s$ belongs to the domain of definition of f. Note that, by convention,

$$\omega(f,x,s) = 0 \ \text{ whenever } \ s = 0. \tag{2.13}$$

If we assume that f is twice continuously differentiable, the mean-value theorem (see [24]) implies that

$$\omega(f,x,s) = 2\int_0^1 \int_0^1 t\frac{\langle s, \nabla^2 f(x+tvs)s\rangle}{\|s\|^2}\, dv\, dt. \tag{2.14}$$

Furthermore, if f is quadratic, then one easily verifies that $\omega(f,x,s)$ is independent of x and is equal to the Rayleigh quotient of the matrix $\nabla^2 f$ in the direction s. We note that, because of AS.4, $\omega(f_i,x,s)$ is bounded by some constant $L_i \geq 0$ (see [24]). Hence we obtain that

$$|\omega(f_i,x,s)| \leq \max\left\{1, \max_{i\in\{1,\dots,p\}} L_i\right\} \stackrel{\text{def}}{=} L \tag{2.15}$$

for all $x, x+s \in X$ and all $i \in \{1, \ldots, p\}$. The quantity that we need in our algorithm statement and analysis is a monotonically increasing upper bound on the magnitude of the generalized Rayleigh quotient $\omega(m_{i,k}, x_k, s_{i,k})$ defined by

$$\beta_k \stackrel{\text{def}}{=} 1 + \max_{\substack{q \in \{0,\ldots,k\} \\ i \in \{1,\ldots,p\}}} |\omega(m_{i,q}, x_q, s_{i,q})| \geq 1, \tag{2.16}$$

where $s_{i,k} \stackrel{\text{def}}{=} P_{\mathcal{R}_i}(s_k)$ for s_k the actual trial step computed by the algorithm, as defined below. The quantity $\omega(m_{i,k}, x_k, s_{i,k})$ measures the curvature of the model $m_{i,k}$ in the direction of the trial step s_k. If quadratic models $m_{i,k}$ are considered, an upper bound on β_k is given by the largest singular value of all Hessian matrices, plus one. We will assume that our choice of models is such that this curvature does not increase too fast, which could lead to premature convergence of the algorithm to a non-critical point (see [41]). More precisely, we make the following assumption, as in [14], [10], [35] and [41].

AS.5

$$\sum_{i=0}^{\infty} \frac{1}{\beta_k} = +\infty. \tag{2.17}$$

This condition is weaker than the common assumption that the model's second derivative matrices are uniformly bounded [32], which holds, for instance, for the classical Newton's method, where quadratic models using analytical second derivatives are used on a compact domain. It is also weaker than the condition

$$|\omega(m_{i,k}, x_k, s_{i,k})| \leq c_0 k \tag{2.18}$$

for some constant $c_0 > 0$, which holds in the case where quadratic element models are used and updated using either the BFGS or the safeguarded Symmetric Rank One quasi-Newton formulae.

2.1.5 Criticality

Before we can describe our algorithm in detail, we also need a *criticality criterion* for our problem. A critical point of our problem is a feasible point x where the negative gradient of the objective function $-\nabla f(x)$ belongs to the *normal cone* of X at $x \in X$, which is defined by

$$\mathcal{N}(x) \stackrel{\text{def}}{=} \{y \in \mathbf{R}^n \mid \langle y, u - x \rangle \leq 0, \forall u \in X\}. \tag{2.19}$$

The associated *tangent cone* of X at $x \in X$ is the polar of $\mathcal{N}(x)$, that is

$$\mathcal{T}(x) \stackrel{\text{def}}{=} \mathcal{N}(x)^0 = \text{closure}\{\lambda(u - x) \mid \lambda \geq 0 \text{ and } u \in X\}. \tag{2.20}$$

Thus every measure of criticality has to depend on the (differentiable) objective f and on the geometry of the feasible set at the current point. We will use the symbol $\alpha(x, f, X)$ to denote such a criticality measure.

AS.6 The criticality measure $\alpha(x, h, X)$ is non-negative for all $x \in X$ and all functions h differentiable in an open neighbourhood of x. Moreover $\alpha(x, h, X) = 0$ if and only if x is critical for the problem

$$\text{minimize}_{x \in X} \ h(x). \tag{2.21}$$

But, within the algorithm, only approximate gradient vectors might be available, namely the vectors g_k and $g_{i,k}$, the gradients of the models. It is therefore natural to use

$$\alpha_k \stackrel{\text{def}}{=} \alpha(x_k, m_k, X), \tag{2.22}$$

the criticality measure for the problem

$$\text{minimize}_{x \in X}\ m_k(x), \tag{2.23}$$

as an "approximate" criticality measure for (2.1). Note that $\alpha_k > 0$ implies that $g_k \neq 0$.

In unconstrained optimization, one typically chooses

$$\alpha_k = \|g_k\|, \tag{2.24}$$

the obvious criticality measure (see [31] or [34]). When bound constraints are present, the choice

$$\alpha_k = \|P_X(x_k - g_k) - x_k\| \tag{2.25}$$

is made in [10]. For the infinite dimensional case, the definition

$$\alpha_k = \|P_X(x_k - g_k) - x_k\|^2 \tag{2.26}$$

is used in [41]. For the case where convex constraints are considered,

$$\alpha_k = \frac{\|P_X(x_k - t_k^C g_k) - x_k\|}{t_k^C}, \tag{2.27}$$

is chosen in [32], where $t_k^C > 0$ is the line coordinate of the so-called "generalized Cauchy point" to be discussed below. In a similar context,

$$\alpha_k = |\min_{\substack{x_k + d \in X \\ \|d\| \leq 1}} \langle g_k, d \rangle| \tag{2.28}$$

is used in [14].

2.2 Ensuring sufficient model decrease

2.2.1 An overview of the classical sufficient decrease condition

A key to trust region algorithms is to choose a step s_k at iteration k that is guaranteed to provide a sufficient decrease on the overall objective function model m_k. In other words, a step such that

$$\delta m_k \stackrel{\text{def}}{=} m_k(x_k) - m_k(x_k + s_k) \tag{2.29}$$

is sufficiently positive, given the value of a suitable criticality measure α_k satisfying AS.6. This concept of "sufficient decrease" is usually made more formal by introducing the notion of the *(generalized) Cauchy point*. This remarkable point, denoted x_k^C, is typically computed by trust region algorithms as a point on (or close to) the projected gradient path $P_X(x_k - tg_k)$ $(t \geq 0)$ that is also within the trust region and sufficiently reduces the overall model in the sense that

$$m_k(x_k) - m_k(x_k^C) \geq \bar{\kappa}_2 \frac{\alpha_k^2}{\beta_k}, \tag{2.30}$$

where $\bar{\kappa}_2 > 0$ is a constant and α_k a criticality measure satisfying AS.6. However, such a point may not exist when the trust region radius Δ_k is small compared with α_k^2/β_k. In this case, the generalized Cauchy point is chosen as (or close to) the intersection of the projected gradient path with the boundary of the trust region, yielding an inequality of the form

$$m(x_k) - m(x_k^C) \geq \bar{\kappa}_2 \alpha_k \Delta_k. \tag{2.31}$$

A point on the projected gradient path satisfying (2.30) may also fail to exist because the projected gradient path itself ends on the boundary of X, well inside the trust region. In that case, this end point (or another feasible point close to it) is typically chosen as generalized Cauchy point, and it is then typically shown that

$$m(x_k) - m(x_k^C) \geq \bar{\kappa}_2 \alpha_k. \tag{2.32}$$

One then ensures the "sufficient decrease" by requiring that the chosen step s_k produces at least a fixed fraction of the overall model reduction achieved by the generalized Cauchy point, which is to say that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_k, 1\right\}, \tag{2.33}$$

where $\kappa_2 \in (0, \bar{\kappa}_2]$.

Many variants on the above scheme exist in the literature for the unstructured trust region case. All of these variants ensure that a suitable step is found after a finite number of trials. The best known is for unconstrained problems when the ℓ_2-norm is used to define the trust region shape. In that case, the projected gradient path is simply given by all negative multiples of the gradient g_k and the Cauchy point is simply the point that minimizes the model m_k in the intersection of the steepest descent direction and the trust region (see, for instance, [34] and [37]). When other norms are used, for example the ℓ_∞-norm, one can then choose either to minimize the model in the intersection of this steepest descent direction and the trust region, as before (see [10]), or to "bend" the projected gradient path onto the boundary of the trust region and to choose the generalized Cauchy point as a point which satisfies classical Goldstein-type linesearch conditions along that path while staying within the trust region (see [33] and [41]). Both these latter strategies are used in the LANCELOT software [13]. When additional convex constraints are present, the projected gradient path is additionally "bent" to follow the boundary of the feasible domain. Thus the philosophy is the same, in that (2.33) is guaranteed in the above cases. Indeed satisfaction of this condition has been derived for each of the choices (2.24)–(2.28) for α_k in the papers where they were respectively introduced.

2.2.2 Sufficient decrease for structured model and trust region

We will use a similar approach in our structured model and trust region framework to determine what is a sufficient decrease of the overall model m_k within the region B_k, whose shape is chosen to reflect the structure of the problem. Special care is needed because this region might be very "asymmetric" in the sense that it may allow very large steps in some directions and only very short ones in others. As a consequence, we have to adapt the notion of trust region "radius" to our context and adequately reformulate condition (2.33).

From a practical point of view, one might use a two-stage approach. In this, one first aims to find a step producing a sufficient model decrease in a smaller, but more symmetric, region. Following this, one then allows the step to increase within the trust region while maintaining control over the model decrease.

To be specific, let

$$B_{min,k} \stackrel{\text{def}}{=} B_k \cap \{x \in \mathbf{R}^n \mid \|x - x_k\| \leq \Delta_{min,k}\}, \tag{2.34}$$

be the trust region whose radius is determined by the possibly most nonlinear part of the model. Applying the results discussed in the previous section after condition (2.33), one may deduce that it is possible to find, in a finite number of trials, a step $s_{min,k}$ such that $x_k + s_{min,k} \in B_{min,k} \cap X$ and

$$m_k(x_k) - m_k(x_k + s_{min,k}) \geq \bar{\kappa}_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_{min,k}, 1\right\}, \tag{2.35}$$

for some suitably chosen criticality measure α_k satisfying AS.6 and some constant $\bar{\kappa}_2 > 0$.

However, the restriction that the length of $s_{min,k}$ is bounded by $\Delta_{min,k}$ makes the whole exercise of shaping B_k to reflect the problem's structure entirely irrelevant. One might therefore be prepared to accept a larger step provided it remains feasible, within the trust region B_k, and produces a further significant model decrease. More specifically, we allow our algorithm to choose any step s_k such that $x_k + s_k \in B_k \cap X$ and which guarantees that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \max[\Delta_{min,k}, \|s_k\|], 1\right\} \tag{2.36}$$

for some $\kappa_2 \in (0, \bar{\kappa}_2]$.

Note that, since (2.36) holds for $s_k = s_{min,k}$, this condition can therefore be achieved in practice after a finite number of trials. Observe also that (2.36) is fundamentally different from an angle test of the form

$$|\langle g_k, s_k\rangle| \geq \zeta \|g_k\| \|s_k\| \quad (\zeta \in (0,1)) \tag{2.37}$$

as (2.36) does not prevent s_k from being orthogonal to the steepest descent direction, so long as a sufficient model reduction is obtained. This is useful because such a step may occur when moving away from a saddle point of the objective function. Finally note that, as expected, (2.36) reduces to (2.33) in the case where only one trust region is considered.

2.3 A class of structured trust region algorithms

We now describe the class of algorithms that we consider for solving (2.1). Besides κ_1 used in (2.8) and κ_2 used in (2.36), it depends on the constants

$$0 < \gamma_1 \le \gamma_2 < 1 \le \gamma_3, \tag{2.38}$$

$$0 < \eta_1 \le \eta_2 < \eta_3 < 1 \tag{2.39}$$

and

$$0 < \mu_1 < \mu_2 < 1. \tag{2.40}$$

In addition to the above conditions, we also require a compatibility condition between the η_i's and the μ_i's. Specifically, we request that

$$\eta_2 - \eta_1 \ge \mu_1 + \mu_2. \tag{2.41}$$

Typical values for these constants are $\kappa_1 = 0.1$, $\kappa_2 = 0.01$, $\gamma_1 = 0.1$, $\gamma_2 = 0.5$, $\gamma_3 = 2$, $\eta_1 = 0.01$, $\eta_2 = 0.25$, $\eta_3 = 0.75$, $\mu_1 = 0.05$ and $\mu_2 = 0.1$.

Algorithm

step 0: initialization.
The starting point $x_0 \in X$ is given, together with the element function values $\{f_i(x_0)\}_{i=1}^p$ and the initial trust region radii $\{\Delta_{i,0}\}_{i=1}^p$. Set $k = 0$.

step 1: model choice.
For $i \in \{1, \ldots, p\}$, choose the model $m_{i,k}$ of the element function f_i in the trust region $B_{i,k}$ centered at x_k (as defined in (2.6)), satisfying (2.7) and (2.8).

step 2: determination of the step.
Choose a step s_k such that the sufficient decrease condition (2.36) holds and

$$x_k + s_k \in B_k \cap X. \tag{2.42}$$

step 3: measure overall model fit.
If

$$\delta f_k \stackrel{\text{def}}{=} f(x_k) - f(x_k + s_k) \ge \eta_1 \delta m_k \tag{2.43}$$

then

$$x_{k+1} = x_k + s_k, \tag{2.44}$$

else

$$x_{k+1} = x_k. \tag{2.45}$$

step 4: update the element trust region radii.

Denote the achieved changes in the element functions and their models by

$$\delta f_{i,k} \stackrel{\rm def}{=} f_i(x_k) - f_i(x_k + s_k), \quad i \in \{1, \ldots, p\} \tag{2.46}$$

and

$$\delta m_{i,k} \stackrel{\rm def}{=} m_{i,k}(x_k) - m_{i,k}(x_k + s_k), \quad i \in \{1, \ldots, p\}, \tag{2.47}$$

respectively. Then define the set of *negligible* elements at iteration k as

$$N_k \stackrel{\rm def}{=} \{i \in \{1, \ldots, p\} \mid |\delta m_{i,k}| \le \frac{\mu_1}{p} \delta m_k\} \tag{2.48}$$

and the set of *meaningful* elements as its complement, that is

$$M_k = \{1, \ldots, p\} \setminus N_k. \tag{2.49}$$

Then, for each $i \in \{1, \ldots, p\}$, perform the following.

Case 1: $i \in M_k$**.**

- If

$$\delta f_{i,k} \ge \delta m_{i,k} - \frac{1 - \eta_3}{p} \delta m_k \tag{2.50}$$

and (2.43) both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}]. \tag{2.51}$$

- If (2.50) holds but (2.43) fails then choose

$$\Delta_{i,k+1} = \Delta_{i,k}. \tag{2.52}$$

- If (2.50) fails, but

$$\delta f_{i,k} \ge \delta m_{i,k} - \frac{1 - \eta_2}{p} \delta m_k \tag{2.53}$$

holds, then choose

$$\Delta_{i,k+1} \in [\gamma_2 \Delta_{i,k}, \Delta_{i,k}]. \tag{2.54}$$

- If (2.53) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}]. \tag{2.55}$$

Case 2: $i \in N_k$**.**

- If

$$|\delta f_{i,k}| \le \frac{\mu_2}{p} \delta m_k, \tag{2.56}$$

and (2.43) both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}]. \tag{2.57}$$

- If (2.56) holds but (2.43) fails, then choose

$$\Delta_{i,k+1} = \Delta_{i,k}. \tag{2.58}$$

- If (2.56) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}]. \tag{2.59}$$

Increment k by one and return to step 1.

End of Algorithm

As is traditional in trust region algorithms, we will call an iteration *successful* if the test (2.43) is satisfied, that is when the achieved objective reduction δf_k is large enough compared to the reduction δm_k predicted by the overall model. If (2.43) fails, the iteration is said to be *unsuccessful*. In what follows, we will denote by $\mathcal{S}$ the set of all successful iterations.

We now comment on various aspects of the algorithm.

1. The algorithm is constructed in such a way that a successful step is always possible, for sufficiently small trust region radii, if the current iterate x_k is not critical. This result is formally proved in Corollary 8.

2. The choice of the element models $m_{i,k}$ is left rather open in the above description. It clearly needs to be made precise for any practical implementation of the algorithm. One common choice would be to set

$$m_{i,k}(x_k + s) = f_i(x_k) + \langle g_{i,k}, s \rangle + \tfrac{1}{2}\langle s, H_{i,k} s \rangle, \tag{2.60}$$

   where $H_{i,k}$ is a symmetric approximation to $\nabla^2 f_i(x_k)$ whose nullspace contains the subspace $\mathcal{N}_i$. In particular, Newton's method corresponds to the choice $g_{i,k} = \nabla f_i(x_k)$ and $H_{i,k} = \nabla^2 f_i(x_k)$, which is guaranteed to satisfy this latter condition. Another possible choice is $m_{i,k}(x_k + s) = f_i(x_k + s)$, which may be attractive for the simpler element functions. In this case, the model's fit to the true function is always good for the i-th element, and the algorithm guarantees that the $\Delta_{i,k}$ form a non-decreasing sequence.

3. If the model change for an element is negligible, that is small compared to the overall predicted change, we do not need to restrict its element trust region size unless the true element change is relatively large compared with the same overall predicted change. We can therefore afford to ignore negligible items until they stop being relatively negligible, something which is inevitable when convergence occurs. Hence our distinction between "negligible" elements (in N_k) and "meaningful" ones (in M_k).

   Condition (2.41) can be viewed in this context as a guarantee that a new iterate will be accepted in (2.43) whenever the model reduction obtained for all meaningful elements is also acceptable (i.e. (2.53) holds for all $i \in M_k$), irrespective of the contribution of the negligible ones. This interpretation is clarified in Lemma 2.

4. The apparent intricacy of (2.50) and (2.53) is caused by two complications which arise in the context of multiple elements. The first is that, although (2.36) ensures that δm_k is always positive, we may not assume in general that the same is true for $\delta m_{i,k}$. The second is that possible cancellation between elements makes it necessary to consider the "accuracy of model fit" for an element to be relative to the *overall* model fit. Indeed, requiring small relative errors for models with very large values may result in large absolute errors. If δm_k is small, these large errors will then cause δm_k to be a poor prediction of δf_k and the iteration might be unsuccessful. This explains why the perhaps more intuitive tests

$$\delta f_{i,k} \geq \delta m_{i,k} - (1-\eta_j)|\delta m_{i,k}| \quad (j = 2,3) \tag{2.61}$$

cannot be used instead of (2.53) ($j = 2$) and (2.50) ($j = 3$).

Observe also that conditions (2.50) and (2.53) reduce to the familiar

$$\delta f_k \geq \eta_j \delta m_k \quad (j = 2,3), \tag{2.62}$$

when $p = 1$.

5. Note again the consistency between the trust region radii updates in step 4 and the case where $p = 1$. In this latter case, the set N_k is always empty and (2.50) then implies (2.43), because of (2.39). Equation (2.52) is thus never invoked.

6. No stopping criterion has been explicitly included in our algorithm description. This is adequate for the theoretical analysis that we consider in the present paper, where we are interested in the asymptotic behaviour of the method, but it should be completed for any practical use. The choice of a particular stopping criterion will depend on the type of models being used.

7. The mechanism that we specified for updating the trust region radii does not exclude the additional requirement that the radii be uniformly bounded, if that is judged suitable for the type of models used. In practice, keeping the radii bounded is essential to prevent numerical overflow.

8. One possible implementation of Step 2 first computes a feasible step s_k^C that minimizes $m_k(P_X(x_k - tg_k))$ within a trust region of radius $\Delta_{min,k}$. Note that s_k^C satisfies (2.35) and (2.36) by construction. This step may then subsequently be increased by progressing further along the arc $P_X(x_k - tg_k) - x_k$ so long as the overall model m_k continues to decrease and (2.36) holds. Additional decrease in m_k may then be obtained (for instance by applying conjugate-gradient steps) provided condition (2.36) is maintained.

Before starting our global convergence analysis, we first state, for future reference, some properties that result from the mechanism of the algorithm.

Lemma 1 *Assume that AS.3 holds. At each iteration k of the algorithm,* *1. M_k contains at least one element. Furthermore*

$$\left(1-\frac{p-1}{p}\mu_1\right)\delta m_k \le \sum_{i\in M_k}\delta m_{i,k} \le \left(1+\frac{p-1}{p}\mu_1\right)\delta m_k; \tag{2.63}$$

2.

$$\gamma_1\Delta_{i,k} \le \Delta_{i,k+1} \le \gamma_3\Delta_{i,k} \tag{2.64}$$

for all $i\in\{1,\ldots,p\}$.

Proof. The first result immediately follows from the definition of N_k and the inequality $\mu_1 < 1$. One then deduces that N_k contains at most $p-1$ elements. Hence,

$$\delta m_k = \sum_{i\in M_k}\delta m_{i,k} + \sum_{i\in N_k}\delta m_{i,k} \le \sum_{i\in M_k}\delta m_{i,k} + \mu_1\frac{|N_k|}{p}\delta m_k \tag{2.65}$$

from which the first part of (2.63) may be deduced. The second inequality in this result is obtained from

$$\sum_{i\in M_k}\delta m_{i,k} = \delta m_k - \sum_{i\in N_k}\delta m_{i,k} \le \delta m_k + \sum_{i\in N_k}|\delta m_{i,k}|, \tag{2.66}$$

the relation (2.48) and $|N_k|\le p-1$. The bound (2.64) results from (2.51), (2.54), (2.55), (2.57) and (2.59). □

We also investigate the coherency between the measure of fit for individual elements and that for the overall model.

Lemma 2 *Assume AS.3 holds and that, at iteration k of the algorithm, (2.53) holds for all $i\in M_k$ and that (2.56) holds for all $i\in N_k$. Then iteration k is successful, i.e. $k\in\mathcal{S}$.*

Proof. Because (2.53) holds for $i\in M_k$, one has that

$$\sum_{i\in M_k}\delta f_{i,k} \ge \sum_{i\in M_k}\delta m_{i,k} - (1-\eta_2)\frac{|M_k|}{p}\delta m_k \ge \left(\eta_2-\frac{p-1}{p}\mu_1\right)\delta m_k \tag{2.67}$$

for all such i, where we used the inequality $|M_k|\le p$ and Lemma 1 to deduce the second inequality. On the other hand, since (2.56) holds for $i\in N_k$, one obtains for these i that

$$\sum_{i\in N_k}|\delta f_{i,k}| \le \frac{p-1}{p}\mu_2\delta m_k, \tag{2.68}$$

where we used item 1 of Lemma 1 to bound $|N_k|$. Now,

$$\delta f_k = \sum_{i\in M_k}\delta f_{i,k} + \sum_{i\in N_k}\delta f_{i,k} \ge \sum_{i\in M_k}\delta f_{i,k} - \sum_{i\in N_k}|\delta f_{i,k}|. \tag{2.69}$$

Combining this last inequality with (2.67) and (2.68) gives that

$$\delta f_k \ge \left(\eta_2-\frac{p-1}{p}\mu_1-\frac{p-1}{p}\mu_2\right)\delta m_k \tag{2.70}$$

which then yields (2.43) because of (2.41). □

We observe from this proof that the weaker condition

$$\eta_2 - \eta_1 \geq \frac{p-1}{p}(\mu_1 + \mu_2) \tag{2.71}$$

could be imposed instead of (2.41). However (2.71), and hence the setting of the algorithm's constants, would then be problem dependent, which one might consider to be undesirable.

Of course, (2.53) holds whenever (2.50) holds because of (2.39). Lemma 2 therefore shows that (2.43) is coherent with the measure of the fit between the element models and element functions.

3 Global convergence

We now study the convergence properties of the class of algorithms that we introduced in the preceding section. Our analysis follows the pattern of similar proofs with an unstructured trust region (see [14] or [41]). The central idea in the proof is that the algorithm will continue to make progress as long as a critical point is not reached. We first start by bounding the error between the true element functions and their models. We next derive a lower bound on the size of the smallest trust region radius at a non-critical point. This lower bound ensures that the trust region constraint will not prevent further progress towards a critical point. Only with this bound can we then prove that limit points of the sequence of iterates produced by the algorithm are indeed critical for the models used. We close the section by deriving some simple consequences of these results on the criticality of the limit points for the true objective function.

We first start by bounding the error made between the model of any element function and the element function itself at $x_k + s_k$.

Lemma 3 *Assume that AS.4 holds and consider a sequence $\{x_k\}$ of iterates generated by the algorithm. Then there exists a positive constant $c_1 \geq 1$ such that*

$$|f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| \leq c_1 \beta_k \Delta_{i,k}^2 \tag{3.1}$$

for all $i \in \{1, \ldots, p\}$ and all k.

Proof. We first observe that, for each $i \in \{1, \ldots, p\}$ and for all k, the definition (2.12), (2.7) and the Cauchy-Schwarz inequality imply that

$$\begin{aligned} |f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| &= |f_i(x_k + s_{i,k}) - m_{i,k}(x_k + s_{i,k})| \\ &\leq |\langle \nabla f_i(x_k) - g_{i,k}, s_{i,k} \rangle| \\ &\quad + \tfrac{1}{2}\|s_{i,k}\|^2 |\omega(f_i, x_k, s_{i,k}) - \omega(m_{i,k}, x_k, s_{i,k})| \\ &\leq \|e_{i,k}\| \, \|s_{i,k}\| \\ &\quad + \tfrac{1}{2}\|s_{i,k}\|^2 (|\omega(f_i, x_k, s_{i,k})| + |\omega(m_{i,k}, x_k, s_{i,k})|). \end{aligned} \tag{3.2}$$

But $\|s_{i,k}\| \leq \Delta_{i,k}$ because of (2.6), and hence we obtain from (2.8), (2.15) and (2.16) that

$$|f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| \leq \kappa_1 \Delta_{min,k} \Delta_{i,k} + \tfrac{1}{2}(L + \beta_k)\Delta_{i,k}^2. \tag{3.3}$$

Using (2.9), this then yields (3.1) with

$$c_1 \stackrel{\text{def}}{=} \kappa_1 + \tfrac{1}{2}(L+1) \geq 1, \tag{3.4}$$

where the last inequality results from (2.15). □

We now derive an upper bound on the change predicted for an element at a non-critical point, as a function of the size of the step in the corresponding range subspace.

Lemma 4 *Assume that AS.1, AS.3 and AS.4 hold. Consider iteration k of the algorithm and assume that, for some $i \in \{1, \ldots, p\}$,*

$$\beta_k \Delta_{i,k} \leq 1. \tag{3.5}$$

Then one has that

$$|\delta m_{i,k}| \leq c_2 \|s_{i,k}\| \tag{3.6}$$

for some constant $c_2 > 0$ independent of i and k.

Proof. We first note that (2.9), (2.16) and (3.5) imply that

$$\Delta_{min,k} \leq 1. \tag{3.7}$$

Using (2.12) and (2.16), we also obtain that

$$|\delta m_{i,k}| \leq |\langle g_{i,k}, s_{i,k}\rangle| + \tfrac{1}{2}\beta_k \|s_{i,k}\|^2 \leq |\langle \nabla f_i(x_k), s_{i,k}\rangle| + |\langle e_{i,k}, s_{i,k}\rangle| + \tfrac{1}{2}\beta_k \|s_{i,k}\|^2. \tag{3.8}$$

Remembering now (2.8), (2.6), (3.5) and (3.7), we can deduce that

$$\begin{aligned} |\delta m_{i,k}| &\leq \max_{x\in\mathcal{R}_i\cap X} \left(\|\nabla f_i(x)\|\right) \|s_{i,k}\| + \kappa_1 \Delta_{min,k} \|s_{i,k}\| + \tfrac{1}{2}\beta_k \|s_{i,k}\|^2 \\ &\leq \left[\max_{x\in\mathcal{R}_i\cap X} \left(\|\nabla f_i(x)\|\right) + \kappa_1 + \tfrac{1}{2}\right] \|s_{i,k}\|. \end{aligned} \tag{3.9}$$

Inequality (3.9) then gives (3.6) with

$$c_2 \stackrel{\text{def}}{=} \max_{\substack{x\in\mathcal{R}_i\cap X \\ i\in\{1,\ldots,p\}}} \left(\|\nabla f_i(x)\|\right) + \kappa_1 + \tfrac{1}{2}. \tag{3.10}$$

□

We next prove the important fact that, so long as a critical point has not been determined, the trust region radii stay sufficiently bounded away from zero, therefore allowing further progress to be made.

Lemma 5 *Assume that AS.1–AS.4 hold. Consider a sequence $\{x_k\}$ of iterates generated by the algorithm and assume that there exists a constant $\epsilon > 0$ such that*

$$\alpha_k \geq \epsilon \tag{3.11}$$

for all k. Then there is a constant $c_3 > 0$ such that

$$\Delta_{min,k} \geq \frac{c_3}{\beta_k} \tag{3.12}$$

for all k.

Proof. Assume, without loss of generality, that

$$\epsilon < \min\{1, \beta_0 \Delta_{min,0}\}. \tag{3.13}$$

In order to derive a contradiction, assume that there exists a k such that

$$\beta_k \Delta_{min,k} < \gamma_1 \min\left\{\epsilon, \frac{\mu_1 c_4^2(1-\eta_3)}{c_1 c_2 p^2}, \frac{c_4(\mu_2-\mu_1)}{c_1 p}\right\} \stackrel{\text{def}}{=} c_3, \tag{3.14}$$

where $c_4 \stackrel{\text{def}}{=} \kappa_2 \gamma_1 \epsilon$. Now define r to be the smallest iteration number such that (3.14) holds. (Note that $r \geq 1$ because of (3.13) and the inequality $\gamma_1 < 1$.) Also fix i such that $\Delta_{min,r} = \Delta_{i,r}$. The monotonic nature of the sequence $\{\beta_k\}$ and the bound (2.64) then ensure that

$$\beta_{r-1}\Delta_{min,r-1} \leq \beta_{r-1}\Delta_{i,r-1} \leq \beta_r \frac{\Delta_{i,r}}{\gamma_1} \leq \frac{c_3}{\gamma_1} \leq \epsilon < 1, \tag{3.15}$$

where we used (3.14) and the inequality (3.13). We note that the definitions of i and r give that

$$\beta_r \Delta_{i,r} = \beta_r \Delta_{min,r} < \beta_{r-1}\Delta_{min,r-1}, \tag{3.16}$$

which in turn implies that $\Delta_{min,r-1} > \Delta_{i,r}$ because of the monotonic nature of the sequence $\{\beta_k\}$. Using this inequality with (2.36), (3.11), and (3.15), we obtain that

$$\begin{aligned} \delta m_{r-1} &\geq \kappa_2 \epsilon \min\left\{\frac{\epsilon}{\beta_{r-1}}, \max[\Delta_{min,r-1}, \|s_{r-1}\|], 1\right\} \\ &\geq \kappa_2 \epsilon \min\left\{\frac{\epsilon}{\beta_{r-1}}, \Delta_{min,r-1}, 1\right\} \\ &\geq \kappa_2 \epsilon \Delta_{min,r-1} \\ &\geq \kappa_2 \epsilon \Delta_{i,r}, \end{aligned} \tag{3.17}$$

which ensures, because of (2.64), that

$$\delta m_{r-1} \geq c_4 \Delta_{i,r-1}. \tag{3.18}$$

But (3.15) guarantees that $\beta_{r-1}\Delta_{i,r-1} \leq 1$. We may thus apply Lemma 4 and deduce that

$$|\delta m_{i,r-1}| \leq c_2 \|s_{i,r-1}\| \leq c_2 \Delta_{i,r-1} \leq \frac{c_2}{c_4} \delta m_{r-1}, \tag{3.19}$$

where we also used (2.6) and (3.18).

Assume first that $i \in M_{r-1}$, which guarantees that $\delta m_{i,r-1} \neq 0$. Then, using (2.48) and (3.18),

$$|\delta m_{i,r-1}| > \frac{\mu_1}{p}\delta m_{r-1} \geq \frac{\mu_1 c_4}{p}\Delta_{i,r-1}. \tag{3.20}$$

Because of (2.7), (3.1) and (3.20), we therefore obtain that

$$\left|\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1\right| = \frac{|f_i(x_{r-1}+s_{r-1}) - m_{i,r-1}(x_{r-1}+s_{r-1})|}{|\delta m_{i,r-1}|} \leq \frac{c_1 p}{\mu_1 c_4}\beta_{r-1}\Delta_{i,r-1}. \tag{3.21}$$

But (3.14) and (3.15) together give that

$$\beta_{r-1}\Delta_{i,r-1} \leq (1-\eta_3)\frac{\mu_1 c_4^2}{c_1 c_2 p^2}, \tag{3.22}$$

which, with (3.21), implies that

$$\left|\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1\right| \leq \frac{(1-\eta_3)c_4}{c_2 p}. \tag{3.23}$$

Consider first the case where $\delta m_{i,r-1} > 0$. We may then apply (3.19) and deduce that

$$\delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1} = \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)}{p}\frac{\delta m_{r-1}}{|\delta m_{i,r-1}|}\right) \leq \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)c_4}{c_2 p}\right). \tag{3.24}$$

Using (3.23), we now deduce that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \geq 1 - \frac{(1-\eta_3)c_4}{c_2 p}, \tag{3.25}$$

and therefore, because of (3.24), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)c_4}{c_2 p}\right) \geq \delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1}, \tag{3.26}$$

which implies that (2.50) holds for element i at iteration $r-1$. Now turn to the case where $\delta m_{i,r-1} < 0$. Because of (3.19), we deduce that

$$\delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1} = \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)}{p}\frac{\delta m_{r-1}}{|\delta m_{i,r-1}|}\right) \leq \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)c_4}{c_2 p}\right). \tag{3.27}$$

As above, we use (3.23) to obtain that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \leq 1 + \frac{(1-\eta_3)c_4}{c_2 p}, \tag{3.28}$$

and therefore, because of (3.27), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)c_4}{c_2 p}\right) \geq \delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1}, \tag{3.29}$$

which again implies that (2.50) holds for element i at iteration $r-1$.

Assume now that $i \in N_{r-1}$. Then, because of (2.7), (2.48) and (3.1), we have that

$$\begin{aligned} |\delta f_{i,r-1}| &\leq |\delta m_{i,r-1}| + |f_i(x_{r-1}+s_{r-1}) - m_{i,r-1}(x_{r-1}+s_{r-1})| \\ &\leq \tfrac{\mu_1}{p}\delta m_{r-1} + c_1\beta_{r-1}\Delta_{i,r-1}^2. \end{aligned} \tag{3.30}$$

Now, multiplying (3.18) by $\Delta_{i,r-1}$, we obtain that

$$\Delta_{i,r-1}^2 \leq \frac{\Delta_{i,r-1}}{c_4}\delta m_{r-1}. \tag{3.31}$$

Combining (3.30) and (3.31), we deduce that

$$|\delta f_{i,r-1}| \leq \left(\frac{\mu_1}{p} + \frac{c_1}{c_4}\beta_{r-1}\Delta_{i,r-1}\right)\delta m_{r-1}. \tag{3.32}$$

Observing now that (3.14) and (3.15) imply that

$$\beta_{r-1}\Delta_{i,r-1} \leq \frac{c_4(\mu_2-\mu_1)}{c_1 p}, \tag{3.33}$$

we obtain from (3.32) that

$$|\delta f_{i,r-1}| \leq \frac{\mu_2}{p} \delta m_{r-1}. \tag{3.34}$$

But this inequality implies that (2.56) holds for element i at iteration $r-1$. Thus either (2.50) or (2.56) holds for element i at iteration $r-1$ and the mechanism of the algorithm then implies that $\Delta_{i,r} \geq \Delta_{i,r-1}$. But we may deduce from this inequality that

$$\beta_{r-1}\Delta_{min,r-1} \leq \beta_{r-1}\Delta_{i,r-1} \leq \beta_r \Delta_{i,r}, \tag{3.35}$$

which contradicts the assumption that r is the smallest iteration number such that (3.14) holds. The inequality (3.14) therefore never holds and we obtain that (3.12) is satisfied for all k. □

We now turn to one of the main results in this section, which proves a weak form of global convergence. The technique is inspired by [35].

Theorem 6 *Assume that AS.1–AS.6 hold. Consider a sequence $\{x_k\}$ of iterates generated by the algorithm. Then*

$$\liminf_{k\to\infty} \alpha_k = 0. \tag{3.36}$$

Proof. Assume, for the purpose of obtaining a contradiction, that there exists an $\epsilon \in (0,1)$ such that (3.11) holds for all $k \geq 0$. Then

$$\begin{array}{rcl} \sum_{k\in\mathcal{S}} \delta f_k & \geq & \eta_1 \sum_{k\in\mathcal{S}} \delta m_k \\ & \geq & \eta_1 \kappa_2 \epsilon \sum_{k\in\mathcal{S}} \min\left\{\frac{\epsilon}{\beta_k}, \max[\Delta_{min,k}, \|s_k\|], 1\right\} \\ & \geq & \eta_1 \kappa_2 \epsilon \sum_{k\in\mathcal{S}} \min\left\{\frac{\epsilon}{\beta_k}, \Delta_{min,k}\right\} \\ & \geq & \eta_1 \kappa_2 \epsilon \min\{\epsilon, c_3\} \sum_{k\in\mathcal{S}} \frac{1}{\beta_k}, \end{array} \tag{3.37}$$

where we used successively (2.43), (2.36), (3.11) and Lemma 5. We note that (3.37) and AS.2 then imply that

$$\sum_{k\in\mathcal{S}} \frac{1}{\beta_k} < +\infty. \tag{3.38}$$

Now let r be an integer such that

$$\gamma_3 \gamma_2^{(r-1)/p} < 1 \tag{3.39}$$

and define

$$\mathcal{S}(k) \stackrel{\text{def}}{=} |\mathcal{S} \cap \{0, \ldots, k-1\}|, \tag{3.40}$$

the number of successful iterations up to iteration $k-1$ ($k \geq 1$). Then define

$$\mathcal{F}_1 \stackrel{\text{def}}{=} \{k \geq 1 \mid k \leq r\mathcal{S}(k)\} \text{ and } \mathcal{F}_2 \stackrel{\text{def}}{=} \{k \geq 1 \mid k > r\mathcal{S}(k)\}. \tag{3.41}$$

We now wish to show that both sums

$$\sum_{k\in\mathcal{F}_1} \frac{1}{\beta_k} \text{ and } \sum_{k\in\mathcal{F}_2} \frac{1}{\beta_k} \tag{3.42}$$

are finite. Consider the first. If it has only finitely many terms, its convergence is obvious. Otherwise, we may assume that $\mathcal{F}_1$ has an infinite number of elements, and we then construct two subsequences. The first consists of the indices of $\mathcal{F}_1$ in ascending order and the second, $\mathcal{F}_3$ say, of the set of indices in $\mathcal{S}$ (in ascending order) with each index repeated r times. Hence the j-th element of $\mathcal{F}_3$ is no greater than the j-th element of $\mathcal{F}_1$. This gives that

$$\sum_{k\in\mathcal{F}_1}\frac{1}{\beta_k}\leq\sum_{k\in\mathcal{F}_3}\frac{1}{\beta_k}=r\sum_{k\in\mathcal{S}}\frac{1}{\beta_k}\leq+\infty \tag{3.43}$$

because of the nondecreasing nature of the sequence $\{\beta_k\}$ and (3.38). Now turn to the second sum in (3.42). Lemma 2 and the mechanism of the algorithm imply that, at each unsuccessful iteration, at least one element trust region radius satisfies (2.55) or (2.59) and none of them is allowed to increase. Hence

$$\prod_{i=1}^{p}\Delta_{i,k}\leq\gamma_3^{p\mathcal{S}(k)}\gamma_2^{k-\mathcal{S}(k)}\prod_{i=1}^{p}\Delta_{i,0}, \tag{3.44}$$

which immediately implies that

$$\Delta_{min,k}\leq\gamma_3^{\mathcal{S}(k)}\gamma_2^{(k-\mathcal{S}(k))/p}\Delta_{max,0}, \tag{3.45}$$

where $\Delta_{max,0}\stackrel{\text{def}}{=}\max_{i\in\{1,\ldots,p\}}\Delta_{i,0}$. We deduce from this inequality that, for $k\in\mathcal{F}_2$,

$$\frac{c_3}{\beta_k}\leq\Delta_{min,k}\leq\gamma_3^{\mathcal{S}(k)}\gamma_2^{(k-\mathcal{S}(k))/p}\Delta_{max,0}\leq\gamma_3^{k/r}\gamma_2^{(k-k/r)/p}\Delta_{max,0}\leq\left[\gamma_3\gamma_2^{(r-1)/p}\right]^{k/r}\Delta_{max,0}, \tag{3.46}$$

where we have also used Lemma 5 and the definition of $\mathcal{F}_2$ in (3.41). Using (3.39), this gives that

$$\sum_{k\in\mathcal{F}_2}\frac{1}{\beta_k}\leq\frac{\Delta_{max,0}}{c_3}\sum_{k\in\mathcal{F}_2}\left[\gamma_3\gamma_2^{(r-1)/p}\right]^{k/r}<+\infty, \tag{3.47}$$

and the second sum is convergent. Therefore the sum

$$\sum_{k=1}^{\infty}\frac{1}{\beta_k}=\sum_{k\in\mathcal{F}_1}\frac{1}{\beta_k}+\sum_{k\in\mathcal{F}_2}\frac{1}{\beta_k} \tag{3.48}$$

is finite, which contradicts AS.5. Hence condition (3.11) is impossible and (3.36) follows. □

Notice that the relation between α_k, the criticality measure for problem (2.23), and $\alpha(x_k,f,X)$, the criticality measure for problem (2.1), has been left rather unspecified up to this point. It is indeed remarkable that we can prove Theorem 6 assuming so little on α. In order to derive convergence properties for the original problem from Theorem 6, we have to be slightly more specific and request that, if both function and model have the same first order information, then the criticality measures on the original problem and on the model problem agree.

AS.7 Let h_1 and h_2 be two continuously differentiable functions in the intersection of X with a neighbourhood of the feasible point x, such that $h_1(x)=h_2(x)$. Then, the difference $\alpha(x,h_1,X)-\alpha(x,h_2,X)$ tends to zero when $\nabla h_1(x)-\nabla h_2(x)$ tends to zero.

In other words, we require the criticality measure to be continuous (near zero) in the *gradient* of its second argument. Again, this is true for the choices (2.24)–(2.25) and (2.28).

With this additional assumption, we are now ready to examine the criticality of the limit points of the sequence of iterates generated by the algorithm for the original problem (2.1).

Corollary 7 *Assume that AS.1–AS.7 hold. Consider a sequence $\{x_k\}$ of iterates generated by the algorithm and assume that*

$$\lim_{k\to\infty} \|e_{i,k}\| = 0 \tag{3.49}$$

for all $i \in \{1, \ldots, p\}$. Then this sequence has at least one critical limit point x_*.

Proof. From AS.7 and (3.49), we obtain that

$$\lim_{k\to\infty} [\alpha(x_k, f, X) - \alpha_k] = 0, \tag{3.50}$$

which, with (3.36), guarantees

$$\liminf_{k\to\infty} \alpha(x_k, f, X) = 0. \tag{3.51}$$

The desired conclusion then follows by taking a subsequence of $\{x_k\}$ if necessary. □

Condition (3.49) is important, otherwise the situation might arise that an iterate is critical for the current overall model (because its gradient is inexact) while not being critical for the original problem. There are various ways in which (3.49) can be achieved in a practical algorithm, the simplest being to make the size of $e_{i,k}$ also depend on α_k itself, ensuring that the first goes to zero if the latter does.

Corollary 8 *Assume that AS.1–AS.7 hold. If $\mathcal{S}$, the set of successful iterations generated by the algorithm is finite, then all iterates x_k are equal to some x_* for k large enough, and x_* is critical.*

Proof. Assume indeed that $\mathcal{S}$ is finite. It is then clear from (2.45) that x_k is unchanged for k large enough, and therefore that $x_* = x_{j+1}$, where j is the largest index in $\mathcal{S}$. Note now that Lemma 2 implies that, if $k \notin \mathcal{S}$, then (2.53) or (2.56) must be violated for at least one element. Hence we obtain that $\Delta_{min,k}$ converges to zero. But (2.8) then implies that $e_{i,k}$ also converges to zero for all $i \in \{1, \ldots, p\}$ and g_k converges to $\nabla f(x_k)$. Thus AS.7 and Corollary 7 then guarantee the criticality of x_*. □

As in existing theories for the unstructured trust region case, it is possible to replace the limit inferior in (3.36) by a true limit, therefore ensuring (if the gradients are asymptotically exact) that *all* limit points are critical. As in these theories, a slight strengthening of our assumptions is however necessary.

AS.8 We assume that

$$\lim_{k\to\infty} \beta_k \delta f_k = 0. \tag{3.52}$$

This assumption is similar to that used in [14] and [41], where it is motivated in detail. We only mention here that (3.52) holds for Newton's method on bounded domains, because β_k is bounded above in that case.

With this additional assumption, we are now able to replace the limit inferior by a true limit.

Theorem 9 *Assume that AS.1–AS.8 hold. Consider the sequence $\{x_k\}$ of iterates generated by the algorithm and assume that there are infinitely many successful iterations. Then*

$$\lim_{k\in\mathcal{S}} \alpha_k = 0, \tag{3.53}$$

where $\mathcal{S}$ is, as above, the set of successful iterations.

Proof. We again proceed by contradiction. Assume therefore that there exists an $\epsilon_1 \in (0,1)$ and a subsequence $\{q_j\}$ of successful iterates such that, for all q_j in this subsequence

$$\alpha_{q_j} \geq \epsilon_1. \tag{3.54}$$

Theorem 6 guarantees the existence of another subsequence $\{l_j\}$ such that

$$\alpha_k \geq \epsilon_2 \ \text{ for } \ q_j \leq k < l_j \ \text{ and } \ \alpha_{l_j} < \epsilon_2, \tag{3.55}$$

where we have chosen $\epsilon_2 \in (0, \epsilon_1)$. We may now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\text{def}}{=} \{k \mid k \in \mathcal{S} \ \text{ and } \ q_j \leq k < l_j\}, \tag{3.56}$$

where q_j and l_j belong, respectively, to the two subsequences defined above. Applying now (2.36) for $k \in \mathcal{K}$, we obtain from (2.43), (2.16) and $\epsilon_2 < 1$ that

$$\delta f_k \geq \eta_1 \kappa_2 \epsilon_2 \min\left\{\frac{\epsilon_2}{\beta_k}, \max[\Delta_{min,k}, \|s_k\|], 1\right\} = \eta_1 \kappa_2 \epsilon_2 \min\left\{\frac{\epsilon_2}{\beta_k}, \max[\Delta_{min,k}, \|s_k\|]\right\}. \tag{3.57}$$

But AS.8, along with (3.57), imply that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \beta_k \|s_k\| = 0 \ \text{ and } \ \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \beta_k \Delta_{min,k} = 0 \tag{3.58}$$

and, because of (2.16), that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \|s_k\| = 0 \ \text{ and } \ \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \Delta_{min,k} = 0. \tag{3.59}$$

Therefore, we can deduce from (3.57) and (3.58), that, for j sufficiently large,

$$\begin{aligned} \|x_{q_j} - x_{l_j}\| &\leq \textstyle\sum_{k=q_j}^{l_j-1} \|x_{k+1} - x_k\| \\ &= \textstyle\sum_{k=q_j}^{l_j-1}{}^{(\mathcal{K})} \|s_k\| \\ &\leq c_5 \textstyle\sum_{k=q_j}^{l_j-1}{}^{(\mathcal{K})} [f(x_k) - f(x_{k+1})] \\ &\leq c_5 [f(x_{q_j}) - f(x_{l_j})], \end{aligned} \tag{3.60}$$

where the sums with superscript $(\mathcal{K})$ are restricted to the indices in $\mathcal{K}$, and

$$c_5 \stackrel{\text{def}}{=} \frac{1}{\eta_1 \kappa_2 \epsilon_2}. \tag{3.61}$$

But AS.2 and the decreasing nature of the sequence $\{f(x_k)\}$ imply that the last right-hand side of (3.60) converges to zero as j tends to infinity. Hence the continuity of ∇f and AS.7 give that

$$|\alpha(x_{q_j}, f, X) - \alpha(x_{l_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.62}$$

for j sufficiently large. On the other hand, the second part of (3.59) and (2.8) imply that g_{q_j} is arbitrarily close to $\nabla f(x_{q_j})$ when j is large enough, and AS.7 hence guarantees that

$$|\alpha_{q_j} - \alpha(x_{q_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.63}$$

for j sufficiently large. We note also that, because of (2.8),

$$\|g_{l_j} - \nabla f(x_{l_j})\| \leq \sum_{i=1}^{p} \|e_{i,l_j}\| \leq \kappa_1 p \Delta_{min,l_j}. \tag{3.64}$$

But the mechanism of the algorithm guarantees that no $\Delta_{i,k}$ can increase between iterations $k_j + 1$ and l_j (assuming $k_j + 1 \neq l_j$), where k_j is the largest integer in $\mathcal{K}$ that is smaller than l_j. This yields that

$$\|g_{l_j} - \nabla f(x_{l_j})\| \leq \kappa_1 \gamma_3 p \Delta_{min,k_j}. \tag{3.65}$$

We now deduce from the second part of (3.59) that the left-hand side of (3.65) tends to zero when j tends to infinity, and therefore that, for j sufficiently large,

$$|\alpha_{l_j} - \alpha(x_{l_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.66}$$

because of AS.7. Combining (3.62), (3.63) and (3.66), we obtain, using (3.55), that

$$\alpha_{q_j} \leq \alpha_{l_j} + \tfrac{1}{2}(\epsilon_1 - \epsilon_2) \leq \tfrac{1}{2}(\epsilon_1 + \epsilon_2) < \epsilon_1, \tag{3.67}$$

which is impossible because of (3.54). Hence our initial assumption cannot hold and the theorem is proved. □

As above, we now consider the case where we impose that the element gradients are asymptotically exact.

Corollary 10 *Assume that AS.1–AS.8 hold. Consider the sequence $\{x_k\}$ of iterates generated by the algorithm and assume furthermore that (3.49) holds for all $i \in \{1, \ldots, p\}$. Then all limit points of this sequence are critical.*

Proof. If the set $\mathcal{S}$ is finite, the conclusion immediately follows from Corollary 8. If, on the other hand, $\mathcal{S}$ has an infinite number of elements, (3.49) implies that g_k is arbitrarily close to $\nabla f(x_k)$ and the combination of AS.7 and Theorem 9 ensures the criticality of any limit point of the sequence of successful iterates. □

Of course, (3.49) might be impossible to achieve in practice, and one might consider the case where we can only assert that

$$\limsup_{k\to\infty} \left[\max_{i\in\{1,\ldots,p\}} \|e_{i,k}\| \right] = \kappa_3, \tag{3.68}$$

for some small constant $\kappa_3 > 0$. This is the case, for instance, if gradients are approximated by finite differences.

Corollary 11 *Assume that AS.1–AS.6 and AS.8 hold. Consider the sequence $\{x_k\}$ of iterates generated by the algorithm. Assume furthermore that (3.68) holds and that, for some constant $L_\alpha > 0$, the criticality measure α satisfies*

$$|\alpha(x, h_1, X) - \alpha(x, h_2, X)| \leq L_\alpha \|\nabla h_1(x) - \nabla h_2(x)\| \tag{3.69}$$

for all $x \in X$ and all functions h_1 and h_2 continuously differentiable in a neighbourhood of x such that $h_1(x) = h_2(x)$. Then, for each limit point x_* of the sequence,

$$\alpha(x_*, f, X) \leq \kappa_3 p L_\alpha. \tag{3.70}$$

Proof. As in Corollary 10, the desired conclusion immediately follows from Corollary 8 if $\mathcal{S}$ is finite. Assume therefore that $\mathcal{S}$ has infinitely many elements. We then deduce that, for all $k \in \mathcal{S}$,

$$\begin{array}{rcl} \alpha(x_k, f, X) & \leq & \alpha_k + |\alpha(x_k, m_k, X) - \alpha(x_k, f, X)| \\ & \leq & \alpha_k + L_\alpha \|g_k - \nabla f(x_k)\| \\ & \leq & \alpha_k + L_\alpha p \max_{i \in \{1,\ldots,p\}} \|e_{i,k}\|. \end{array} \tag{3.71}$$

Taking the limit for k tending to infinity in $\mathcal{S}$ and using Theorem 9 and (3.68) then gives the desired conclusion. □

Finally observe that although (3.69) is stronger than AS.7, it is not a very strong condition. For instance, it is satisfied with $L_\alpha = 1$ for the choices (2.24), and also for (2.25) and (2.26) because of the non-expansive character of the projection operator P_X (see [41], for example). The same property also holds for the choice (2.28), as discussed in [14].

4 Finite identification of the correct active set

When applied to constrained problems, trust region algorithms typically use the notion of projected gradient or projected gradient path in order to identify a subset of inequality constraints that are satisfied as equalities. Ultimately, the aim thereby is to identify the constraints satisfied as equalities at the solution well before the solution is reached. The methods then reduce to an unconstrained calculation in the manifold defined by the currently "active" constraints. As a consequence, it is possible to guarantee fast asymptotic rates of convergence when using accurate models, as is the case when analytical second order information of the objective and constraint functions is available.

It is possible to show that structured trust regions do not upset the theory developed in the unstructured case: it can indeed be shown that the constraints active at a particular limit point of the sequence of iterates are identified after a finite number of iterations, provided the normals of the active constraints are linearly independent and strict complementarity holds, and provided the step s_{k+1} satisfies the inequality

$$\|s_{k+1}\|_{(k+1)} \geq \gamma_0 \|s_k\|_{(k)} \tag{4.1}$$

for each $k \notin \mathcal{S}$ and for some constant $\gamma_0 \in (0, \gamma_1]$. This latter condition is meant to avoid a situation where the successful iterates converge to a critical point while a subsequence of unsuccessful iterates converges to another point with a different active set. It does not constitute a severe restriction in the step selection procedure and is automatically verified if s_k is determined by a succession of steps of increasing norm such that they remain feasible, within the trust region B_k, and ensure (2.36). This is the case, for instance, if truncated conjugate gradients are used for computing the step in the solution of an unconstrained problem (see [37] or [38]).

The theory considers the active constraint identification problem from a quite general point of view. The main observation is that a number of the existing theories for active constraint identification are based on the definition of a special criticality measure that satisfies AS.6 while not satisfying AS.7 (see [2] or [3], for instance). Let us denote this measure at iteration k by $\bar{\alpha}_k$. The steps leading to constraint identification are then as follows.

1. The first step is to prove that a sufficient decrease condition of the type (2.33) also holds with $\bar{\alpha}_k$ instead of α_k.

2. One then proceeds to prove that
$$\liminf_{k\to\infty} \bar{\alpha}_k = 0 \tag{4.2}$$
much in the same way as for (3.36).

3. The measure $\bar{\alpha}_k$ is also constructed to ensure that it is asymptotically bounded away from zero for all points such that their active set is not identical to that of a (close) critical point. (This, in particular, prevents AS.7 from holding.)

4. Some contradiction is then deduced from these last two properties.

However, since this development is rather technical and lengthy, we do not include it in the present paper, but refer the interested reader to [15] for details of the results and additional assumptions. This reference also contains the theory concerning the convergence of the iterates to a single limit point, adapted from [14].

Our experience with the solution of practical problems however indicates that the identification of active constraints is seldom observed in practice before the very last iterations of the algorithm, which makes the results discussed in this section mainly of theoretical interest.

5 Extensions

We examine in this section some extensions and variants of the results presented above.

5.1 A hybrid technique

One of the possible drawbacks of the algorithm of Section 2.3 is that steps might be constrained to be unnecessarily small in directions corresponding to highly nonlinear element functions. Indeed, the negative effect of inaccurate models for these elements might be compensated by a successful step in directions corresponding to less nonlinear elements. This compromise between the different parts of the objective is, of course, inherent to the classical method using an unstructured trust region.

We might try to obtain the best of both classical and structured approaches by using a hydrid technique. In this technique, a *global* trust region radius Δ_k is recurred for the objective function considered as a single element (using the algorithm analyzed above, which is then equivalent to the classical one), along with the individual radii $\Delta_{i,k}$. We then define the individual "hybrid" radii by
$$\Delta_{i,k}^h \stackrel{\text{def}}{=} \max\{\Delta_k, \Delta_{i,k}\} \tag{5.1}$$

for each $i \in \{1, \ldots, p\}$ and redefine $B_{i,k}$ as

$$B_{i,k} \stackrel{\rm def}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\| \leq \Delta^h_{i,k}\}. \tag{5.2}$$

We can then apply our algorithm with these new quantities, to the effect that well-modelled elements have their associated trust regions possibly extended without having to contract those corresponding to badly-modelled ones, as long as the global agreement is satisfactory.

It is not difficult to verify that the theory presented above still holds for this hybrid modification. The key points are to observe that the revised definition of our trust region implies that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_k, 1\right\}, \tag{5.3}$$

which is the classical sufficient decrease condition (2.33), that the inequalities (2.64) are still valid with $\Delta_{i,k}$ replaced by $\Delta^h_{i,k}$, and also that an analogous result to Lemma 5 also holds for the global trust region radius, as is already well-known from the unstructured trust region case (see [14], for instance).

5.2 An alternative definition of success

An immediate consequence of inequality (2.63) in Lemma 1 is that it would be possible to replace the condition (2.43) for an iteration to be successful by

$$\delta f_k \geq \eta_1 \sum_{i \in M_k} \delta m_{i,k}(x_k), \tag{5.4}$$

without altering the developments presented above. Indeed, (2.63) shows the equivalence between (2.43) and (5.4). We have chosen to use (2.43) above, because it seems natural to consider the same collection of elements on both sides of the inequality.

5.3 Weaker sufficient decrease conditions

It is remarkable to note that Lemma 5 and Theorem 6 can be proved in a weaker context. Indeed, we could require the weaker sufficient decrease condition

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_{min,k}, 1\right\} \tag{5.5}$$

instead of (2.36), and still prove Lemma 5 and Theorem 6. However, we have not been able to prove Theorem 9, nor active constraint identification, with these assumptions, because (5.5) only involves the length of the step in a possibly small subspace of $\mathbf{R}^n$.

5.4 Using uniformly equivalent norms

Another possible generalization of the theory developed above allows the use of different norms for each element and for each iteration. Let us denote these norms by the symbol $\|\cdot\|_{(i,k)}$. The element trust region definition (2.6) then becomes

$$B_{i,k} \stackrel{\rm def}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\|_{(i,k)} \leq \Delta_{i,k}\}, \tag{5.6}$$

while the gradient approximation condition (2.8) may be written as

$$\|e_{i,k}\|_{[i,k]} \leq \kappa_1 \Delta_{min,k} \tag{5.7}$$

where the norm $\|\cdot\|_{[i,k]}$ is any norm that satisfies

$$|\langle x, y\rangle| \leq \|x\|_{(i,k)}\|y\|_{[i,k]} \tag{5.8}$$

for all $x, y \in \mathbf{R}^n$. In particular, one can choose the *dual norm* of $\|\cdot\|_{(i,k)}$ defined by

$$\|y\|_{[i,k]} \stackrel{\rm def}{=} \sup_{x\neq 0} \frac{|\langle x, y\rangle|}{\|x\|_{(i,k)}}. \tag{5.9}$$

With iteration k, we may also associate an overall norm $\|\cdot\|_{(k)}$ defined on the whole of $\mathbf{R}^n$, whose purpose is to reflect the relative weighting of the different elemental norms $\|\cdot\|_{(i,k)}$ in a global measure.

If we assume that all the considered norms are *uniformly equivalent*, that is if there exists a constant $\sigma \geq 1$ such that, for all x,

$$\frac{1}{\sigma}\|x\|_b \leq \|x\|_a \leq \sigma\|x\|_b, \tag{5.10}$$

where $\|\cdot\|_a$, $\|\cdot\|_b$ is any pair of the above defined norms, then the theory developed in all the preceding sections is still valid without any substantial modification. Again the details of the proofs in this more general setting are provided in [15]. Note that this extension covers the possible introduction of iteration dependent scaling in a practical implementation of our algorithm, which can be highly desirable for some difficult problems.

6 Conclusions

We have shown in this paper that the trust region concept, one of the most powerful tools for building efficient and robust algorithms for optimization, can be extended in a very natural way to reflect the structure of the underlying problem. The algorithm proposed above is indeed a direct generalization of the more usual case where only an unstructured uniform trust region is considered. Similar global convergence properties can be proved for the new algorithm, including the case where dynamic scaling is performed on the variables and the situation where the gradients are only known approximately.

It remains to see if this modification of a trust region algorithm will prove efficient in practice and justify the slight additional complexity of the method. Note that the results of preliminary numerical experiments (based on a modification of LANCELOT using the implementation described after the algorithm) have been encouraging. Tests on unconstrained problems from the CUTE collection [1] have shown that the new method, although very comparable to LANCELOT in many cases, sometimes produces substantial improvements. However, we anticipate the real power of the concept to appear when minimizing augmented Lagrangians or other penalty-like functions, because scaling is much more critical there than in many of the classical unconstrained test examples. The authors are planning to include the new technique described in this paper within the next release of LANCELOT.

One of the nice features of the partially separable functions considered in the present theory is that the objective is a *linear* combination of its elements. While group partially separability, as used in [12] or [13], has computational advantages in terms of economy of derivative calculation, this structure involves a *nonlinear* relationship between the elements and the overall function. This seems to make exploiting the link between local and global models much harder. While we would be interested in deriving structured trust region methods for group partially separable functions, the methods would undoubtedly be more complicated and less amenable to analysis. Thus, we are content, in the present paper, to consider the simpler, but nonetheless very general, partially separable structure.

Finally, there might be other ways to introduce structure in trust region methods than considering (group) partially separable objective functions. In particular, trust region methods for nonlinearly constrained problems seems attractive candidates for an alternative approach that would separate the trust region(s) on the objective from those on the constraints.

7 Acknowledgments

The authors are indebted to Johara Shahabuddin for twice pointing out an unsuitable definition of the sufficient decrease condition (2.36) in Section 2.2.2.

References

[1] I. Bongartz, A. R. Conn, N.I.M. Gould, and Ph. L. Toint. CUTE: Constrained and Unconstrained Testing Environment. *ACM Transactions on Mathematical Software*, 21(1):123–160, 1995.

[2] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25(5):1197–1211, 1988.

[3] J. V. Burke, J. J. Moré, and G. Toraldo. Convergence properties of trust region methods for linear and convex constraints. *Mathematical Programming, Series A*, 47(3):305–336, 1990.

[4] R. H. Byrd, E. Eskow, R. B. Schnabel, and S. L. Smith. Parallel global optimization: numerical methods, dynamic scheduling methods, and application to molecular configuration. Technical Report CU-CS-553-91, Department of Computer Science, University of Colorado at Boulder, Boulder, USA, 1991.

[5] R. H. Byrd, R. B. Schnabel, and G. A. Schultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24:1152–1170, 1987.

[6] P. H. Calamai and J. J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39:93–116, 1987.

[7] R. G. Carter. On the global convergence of trust region methods using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[8] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984*, pages 71–82, 1985.

[9] A. R. Conn, N. I. M. Gould, M. Lescrenier, and Ph. L. Toint. Performance of a multifrontal scheme for partially separable optimization. In *Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, volume 275, pages 79–96, Dordrecht, NL, 1994. Kluwer Academic Publishers.

[10] 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*, 25:433–460, 1988. See also same journal 26:764–767, 1989.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation*, 50:399–430, 1988.

[12] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. An introduction to the structure of large scale nonlinear optimization problems and the lancelot project. In R. Glowinski and A. Lichnewsky, editors, *Computing Methods in Applied Sciences and Engineering*, pages 42–51, Philadelphia, USA, 1990. SIAM.

[13] 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.

[14] A. R. Conn, Nick 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.

[15] A. R. Conn, N.I.M. Gould, A. Sartenaer, and Ph. L. Toint. Convergence properties of minimization algorithm for convex constraints using a structured trust region (revised). Technical Report 94/5, Department of Mathematics, FUNDP, Namur, Belgium, 1994.

[16] M. El-Alem. A global convergence theory for the Dennis-Celis-Tapia trust-region algorithm for constrained optimization. *SIAM Journal on Numerical Analysis*, 28(1):266–290, 1991.

[17] R. Fletcher. *Practical Methods of Optimization: Unconstrained Optimization.* J. Wiley and Sons, New-York, 1980.

[18] D. M. Gay. Is exploiting partial separability useful? Talk at the 6th Mexico United States Workshop, Oaxaca, Mexico, January 1992.

[19] D. Goldfarb and S. Wang. Partial-update Newton methods for unary, factorable and partially separable optimization. *SIAM Journal on Optimization*, 3(2):383–397, 1993.

[20] A. Griewank. The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients. *Mathematical Programming, Series A*, 50(2), 1991.

[21] A. Griewank and Ph. L. Toint. On the unconstrained optimization of partially separable functions. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312, London and New York, 1982. Academic Press.

[22] A. Griewank and Ph. L. Toint. Numerical experiments with partially separable optimization problems. In D. F. Griffiths, editor, *Numerical Analysis: Proceedings Dundee 1983*, pages 203–220, Berlin, 1984. Springer Verlag. Lecture Notes in Mathematics 1066.

[23] A. Griewank and Ph. L. Toint. On the existence of convex decomposition of partially separable functions. *Mathematical Programming*, 28:25–49, 1984.

[24] W. A. Gruver and E. Sachs. *Algorithmic Methods in Optimal Control.* Pitman, Boston, USA, 1980.

[25] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, UK, 1973.

[26] M. Lescrenier. Partially separable optimization and parallel computing. In R. R. Meyer and S. Zenios, editors, *Parallel Optimization on Novel Computer Architectures*, Switzerland, 1989. A. C. Baltzer Scientific Publishing Co.

[27] M. Lescrenier. Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM Journal on Numerical Analysis*, 28(2):476–495, 1991.

[28] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly Journal on Applied Mathematics*, 2:164–168, 1944.

[29] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.

[30] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Proceedings Dundee 1977*, Berlin, 1978. Springer Verlag. Lecture Notes in Mathematics.

[31] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 258–287, Berlin, 1983. Springer Verlag.

[32] J. J. Moré. Trust regions and projected gradients. In M. Iri and K. Yajima, editors, *System Modelling and Optimization*, volume 113, pages 1–13, Berlin, 1988. Springer Verlag. Lecture Notes in Control and Information Sciences.

[33] J. J. Moré and G. Toraldo. On the solution of large scale quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

[34] M. J. D. Powell. A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, New York, 1970. Academic Press.

[35] M. J. D. Powell. On the global convergence of trust region algorithms for unconstrained optimization. *Mathematical Programming*, 29:297–303, 1984.

[36] M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49(2), 1990.

[37] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[38] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88, London, 1981. Academic Press.

[39] Ph. L. Toint. Global convergence of the partitioned BFGS algorithm for convex partially separable optimization. *Mathematical Programming*, 36(3):290–307, 1986.

[40] Ph. L. Toint. On large scale nonlinear least squares calculations. *SIAM Journal on Scientific and Statistical Computing*, 8(3):416–435, 1987.

[41] Ph. L. Toint. Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, 8:231–252, 1988.

[42] Ph. L. Toint and D. Tuyttens. On large scale nonlinear network optimization. *Mathematical Programming, Series B*, 48(1):125–159, 1990.

[43] Ph. L. Toint and D. Tuyttens. LSNNO: a Fortran subroutine for solving large scale nonlinear network optimization problems. *ACM Transactions on Mathematical Software*, 18(3):308–328, 1992.

[44] A. Vardi. A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM Journal on Numerical Analysis*, 22(3):575–591, 1985.