On the convergence of derivative-free methods for unconstrained optimization

by A.R. Conn[1], K. Scheinberg[2] and Ph.L. Toint[3]

Report 96/10 December 17, 1996

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

[2] Industrial Engineering and Operations Resaerch Department,
Columbia University, New York, NY 10027-6699, USA
Email: katya@ieor.columbia.edu

[3] Department of Mathematics, Facultés Universitaires ND de la Paix, 61, rue de Bruxelles,
B-5000 Namur, Belgium, EU
Email: pht@math.fundp.ac.be
Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be
WWW: http://www.fundp.ac.be/sciences/math/pht/publications.html

Invited presentation at the Powellfest, Cambridge, July 1996.

On the convergence of derivative-free methods for unconstrained optimization

A. R. Conn K. Scheinberg Ph. L. Toint

December 17, 1996

This paper is dedicated to Professor M. J. D. Powell for his 60th birthday.

Abstract

The purpose of this paper is to examine a broad class of derivative-free trust-region methods for unconstrained optimization inspired by the proposals of Powell (1994b) and to derive a general framework in which reasonable global convergence results can be obtained. The developments make extensive use of an interpolation error bound derived by Sauer and Xu (1995) in the context of multivariate polynomial interpolation.

1 Introduction

The problem that we consider in this paper is that of minimizing a nonlinear smooth objective function of several variables when the derivatives of the objective function are unavailable. More formally, we consider the problem

$$\min_{x \in \mathbf{R}^n} f(x), \tag{1.1}$$

where we assume that f is smooth and that $\nabla f(x)$ (and, a fortiori, $\nabla^2 f(x)$) cannot be computed for any x.

The main motivation for examining algorithmic solutions to this problem is the high demand from practitioners for such tools. In the cases presented to the authors, $f(x)$ is typically *very expensive* to compute, and its derivatives are not available either because $f(x)$ results from some physical, chemical or econometrical measure, or, more commonly, because it is the result of a possibly very large computer simulation, for which the source code is effectively unavailable. The occurrence of problem of this nature appears to be surprisingly frequent in the industrial world.

When users are faced with such problems, there are a few strategies that can be considered. The first, and maybe simplest, is to apply existing "direct search" optimization methods, like the well-known and widely used simplex reflection algorithm of Nelder and Mead (1965) or its modern variants, or the Parallel Direct Search algorithm of Dennis and Torczon (1991) and Torczon (1991). This first approach has the merit of requiring little additional effort from the user, but may require substantial computing resources: the inherent smoothness of the objective function is not very well exploited, and, as a result, the number of function evaluations is sometimes very large, a major drawback when these evaluations are expensive.

The second and more sophisticated approach is to turn to automatic differentiation tools (see Griewank and Corliss (1991) or Griewank (1994), for instance). However, such tools are unfortunately not applicable in the two typical cases mentioned above since they require the function that is to be differentiated be the result of a callable program that cannot be treated as a black box.

A third possibility is to resort to finite difference approximation of the derivatives (gradients and possibly Hessian matrices). A good introduction to these techniques is provided by the book of Dennis and Schnabel (1983), for instance. In general, given the cost of evaluating the objective function, evaluating its Hessian by finite difference is much too expensive; one can use quasi-Newton Hessian approximation techniques instead. In conjunction with the use of finite differences for computing gradients, this type of method has proved to be useful and sometimes surprisingly efficient.

We will however focus this paper on a fourth possible approach, which is based on the idea of modeling the objective function directly, instead of modeling its derivatives. This idea seems particularly attractive in that one can replace an expensive function evaluation by a much cheaper surrogate model and, especially for very complex problems, make considerable progress in obtaining improved solutions at moderate cost. Such modeling has been promoted by several contributions of Powell. The following interesting argument is for instance proposed in Powell (1974):

> However I believe that eventually the better methods will not use derivative approximations. (...) One reason for my belief comes from the relatively easy problem of solving a single nonlinear equation. Here I think it is fair to compare the secant method with the Newton-Raphson method. The secant method require one function evaluation per iteration and has a convergence rate of 1.618 per function value (...) while the Newton-Raphson method requires one function and one derivative value per iteration and has quadratic convergence. Therefore, if an extra function value is used to estimate the derivative in the Newton-Raphson iteration, the mean rate of convergence per function value is about 1.414, which is not as good as the secant method.

We will call *derivative free optimization* the techniques which do not attempt to directly compute approximations to the unavailable derivative information, but rather that improvement will be derived from a model of the objective function itself without this derivative information. Our purpose in this paper is to explore the theoretical aspects of these methods: we will define a broad class of derivative free trust-region methods and consider their convergence theory.

The next section will be devoted to Powell's contributions to derivative free optimization, because of their crucial nature and the influence they have had on the rest of our presentation. Section 3 will be concerned with the description of the algorithmic framework we wish to study, while convergence of this framework will be examined in Section 4. Some conclusions and perspectives will finally be outlined in Section 5.

2 Powell's contributions to derivative free optimization

Since the 1960's, Powell's views on derivative free optimization have had a substantial influence on the development of this area, both because of their varied and fundamental nature.

2.1 Conjugate directions

The first major contribution of Powell in this context appeared in Powell (1964), where he described a method for solving the nonlinear unconstrained minimization problem based on the use of conjugate directions. The main idea is that the minimum of a positive-definite quadratic form can be found by performing at most n successive line searches along mutually conjugate directions, where n is the number of variables. The same procedure may of course be applied to non-quadratic functions, adding a new composite direction at the end of each cycle of n line searches. Of course, finite termination is no longer expected in this case.

This algorithm has enjoyed a lot of interest amongst both numerical analysts and practitioners. The properties of the method were analyzed by Brent (1973), Callier and Toint (1977), Toint and Callier (1977) and Toint (1978). On the more practical side, Powell made his method available in the Harwell Subroutine Library under the name of VA04. The original routine is no longer distributed, and has been replaced by VA24, also written by Powell (see Harwell Subroutine Library, 1995). Brent also proposed a variant of the algorithm and gave an ALGOL W code named PRAXIS. This code was subsequently translated to Fortran by R. Taylor, S. Pinski and J. Chandler, and has since been widely used in practice[1].

Methods of conjugate directions have proved to be reasonably reliable, but suffer from two main disadvantages: the need to maintain good linear independence of the search directions and the relative difficulty of determining near-conjugate directions when the Hessian of the function is ill-conditioned.

Recognizing these difficulties, Powell (1974) suggested using orthogonal transformations of sets of conjugate directions, a technique leading to the idea of B-conjugate methods, as described in Powell (1975). The main idea of this proposal was to approximate the matrix of second derivatives itself, by modifying an initial estimate to ensure that it satisfies properties which would be satisfied if the objective function were quadratic. He also suggested the use of variational criteria, such as those used to derive quasi-Newton updates, in order that good information from an approximation can be inherited by its successors at subsequent iterations.

We wish to emphasize here that this approach is already aimed at building approximate (quadratic) models for the objective function, an idea which is crucial in our version of derivative free optimization methods.

[1] PRAXIS is distributed in the public domain by J. Chandler, Computer Science Department, Oklahoma State University, Stillwater, Oklahoma 70078, USA (jpc@a.cs.okstate.edu). There is also an interface with the CUTE testing environment of Bongartz *et al.* (1995).

2.2 Methods for nonlinear equations

Powell also had significant influence on the related question of the numerical solution of sets of nonlinear equations and the nonlinear least-squares problems. The method described in Powell (1965) again uses uses the idea of searching along a set of linearly independent directions, but estimates the derivatives of the residuals along these directions by finite differences, therefore avoiding the use of analytical expressions for derivatives. A search direction is then computed using the Gauss-Newton equation, and is used to replace one search direction of the current set. The method exploits the fact that successive Gauss-Newton matrices only differ by a low rank term.

Five years later, Powell (1970b) describes a method where the Broyden quasi-Newton formula is used to approximate the Jacobian matrices associated with such problems, promoting at the same time the idea of a modeling the nonlinear function of interest within a trust-region, an idea that we will pursue in the rest of this paper. The implementation of this algorithm for the Harwell Subroutine Library under the name of NS01 is detailed in Powell (1970a), and was for a long time a favourite amongst practitioners and numerical analysts alike.

2.3 Methods based on multivariate interpolation

The third important contribution by Powell is the main source of our inspiration for the rest of this paper. Powell (1994a) proposed a method for constrained optimization, where the objective function and constraints are approximated by linear multivariate interpolation. At the same time, the associated Fortran code, named COBYLA, was (and still is) distributed by Powell to interested parties[2]. A little later, Powell (1994b) described an algorithm for unconstrained optimization using a multivariate quadratic interpolation model of the objective function, an approach revisited by Powell (1996).

The main idea of Powell's proposal was to use the available objective function values $f(x_i)$ (or in the constrained case, an l_∞ penalty function). For the purposes of this paper we will confine our attention to the unconstrained results where an interpolation model for f is used. This model is assumed to be valid in a neighbourhood of the current iterate: this neighbourhood is described as a trust region (an hypersphere centered at x_i and of radius Δ). This model is then minimized within the trust region, hopefully yielding a point with a low function value. As the algorithm proceeds and more objective function values become available, the points defining the interpolation model (the interpolation set) are updated in a way that preserves their geometrical properties and the trust-region radius is also adapted. Of course, this is only a very crude description of Powell's idea, but we will return to a more formal framework in the next section.

A variant of Powell's quadratic interpolation scheme is discussed in Conn and Toint (1996), where encouraging numerical results are presented. Similar computational results were also mentioned in Powell (1996).

[2]An interface with the CUTE environment of Bongartz *et al.* (1995) is also provided for COBYLA.

3 A class of trust region algorithms for derivative free optimization

We now turn to a more precise description of the problem and of the algorithmic framework we propose to analyze. We consider problem (1.1) and assume that $f(x)$ can be evaluated (at a supposedly high cost) for every $x \in \mathbf{R}^n$. As in Powell (1994b), Conn and Toint (1996) and Powell (1996), we propose to use an iterative algorithm of the trust-region type, in which the objective function is modeled, at iteration k, by a quadratic of the form

$$m_k(x_k + s) = f(x_k) + \langle g_k, s\rangle + \frac{1}{2}\langle s, H_k s\rangle. \tag{3.2}$$

In this definition, we have used the notation $\langle \cdot, \cdot \rangle$ to denote the usual Euclidean inner product on $\mathbf{R}^n$, and we will subsequently use the symbol $\|\cdot\|$ for the associated Euclidean norm. We also used x_k to denote the k-th iterate, while g_k is a vector of $\mathbf{R}^n$ and H_k a symmetric $n \times n$ matrix to be determined. The vector s represents a step from the iterate x_k, which, according to the trust-region methodology, will be restricted in norm to ensure that $x_k + s$ belongs to

$$\mathcal{B}_k \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid \|x - x_k\| \le \Delta_k\}, \tag{3.3}$$

where Δ_k is a positive parameter. The ball $\mathcal{B}_k$ is called the *trust region*, and Δ_k is the *trust-region radius*. This radius will be adjusted by the algorithm from iteration to iteration.

In order to keep our analysis simple and hopefully intuitive, we choose a simple set of assumptions.

$\boxed{\textbf{A.1}}$ The objective function f is twice continuously differentiable and its gradient and Hessian is uniformly bounded over $\mathbf{R}^n$, which means that there exists constants $\kappa_{\rm fg} > 0$ and $\kappa_{\rm fh} > 0$ such that

$$\|\nabla f(x)\| \le \kappa_{\rm fg}$$

and

$$\|\nabla^2 f(x)\| \le \kappa_{\rm fh}$$

for all $x \in \mathbf{R}^n$.

$\boxed{\textbf{A.2}}$ The objective function is bounded below on $\mathbf{R}^n$.

$\boxed{\textbf{A.3}}$ The Hessians of all models generated are uniformly bounded, that is there exists a constant $\kappa_{\rm mh} > 0$ such that

$$\|H_k\| \le \kappa_{\rm mh}$$

for all $x \in \mathcal{B}_k$.

In what follows, we will also use the constant $\kappa_{\rm h} = \max[\kappa_{\rm fg}, \kappa_{\rm fh}, \kappa_{\rm mh}]$.

3.1 Properties of multivariate interpolation models

We now turn to the determination of the model (3.2). As stated above, we will derive the values of g_k and H_k from a set of *interpolation conditions* of the form

$$m_k(y) = f(y) \text{ for each } y \in Y, \tag{3.4}$$

for some *interpolation set* Y. If a "fully quadratic" model of the form (3.2) is used, Y needs to contain $p = (n+1)(n+2)/2$ points to determine g_k and H_k uniquely. However, even p interpolation conditions of the form (3.4) are in general not sufficient for this determination if the number of variables, n, is larger than one. For instance, six interpolation points on a circle in the plane do not specify a two-dimensional quadratic uniquely, because any quadratic which is a multiple of the equation of the circle can be added to the interpolant without affecting (3.4). One therefore sees that some *geometric conditions* on Y must be added to the conditions (3.4) to ensure existence and uniqueness of the quadratic interpolant. In the case of our example, we must have that the interpolation points do not lie on any quadratic surface in $\mathbf{R}^n$ or that the chosen basis includes terms of higher degree (see De Boor and Ron (1992) or Sauer and Xu (1995) for details). More formally, we need the condition referred to as poisedness, which relates directly to the interpolation points and the approximation space.

In order to continue the discussion of the geometry of the interpolation set, we need to explore multivariate interpolation techniques a little further, which we do by considering the more general problem of finding interpolating polynomials of degree d. To be able to approximate in the desired interpolation space (the quadratic polynomials, say) one typically chooses a suitable basis (for example the monomials) to initiate the process. The techniques then differ in that they combine these basis functions in different ways, although the final result is clearly unique in the interpolation space if the geometry of the interpolation set is suitable. At variance with Powell's proposals, which are based on the use of the Lagrange interpolation polynomials as a tool to build the interpolant, we will, in this paper, emphasize Newton fundamental polynomials. Although there are some significant computational advantages to using Newton polynomials, in the context of our discussion the main reason for this choice is that we will exploit an error formula associated with this latter class of polynomials.

It is interesting to note at this point that the choice of the monomials as an initial basis for the interpolation space has the advantage that any known sparsity structure in the objective function's Hessian may directly be exploited. For instance, if the (i,j)-th entry of this Hessian is known to be zero, then the monomial $x_i x_j$ may simply be removed from the initial basis, and the interpolating model will automatically exhibit the desired sparsity pattern in its Hessian. Other choices for the initial basis are nevertheless possible: for instance, Sauer (1995) considers using Bernstein polynomials for improved numerical stability.

In the framework of multivariate interpolation using Newton's fundamental polynomials, the point y in the interpolation set Y are organized into $d+1$ *blocks* $Y^{[\ell]}$, $(\ell = 0,\ldots,d)$, the ℓ-th block containing $|Y^{[\ell]}| = \binom{\ell+n-1}{\ell}$ points. To each point $y_i^{[\ell]} \in Y^{[\ell]}$ corresponds a single *Newton fundamental polynomial* of degree ℓ satisfying conditions

$$N_i^{[\ell]}(y_j^{[m]}) = \delta_{ij}\delta_{\ell m} \ \text{ for all } \ y_j^{[m]} \in Y^{[m]} \ \text{ with } \ m \le \ell. \tag{3.5}$$

For instance, if we consider cubic interpolation on a regular grid in the plane, we require ten interpolation points using four blocks

$$Y^{[0]} = \{(0,0)\}, \quad Y^{[1]} = \{(1,0),(0,1)\},$$

$$Y^{[2]} = \{(2,0),(1,1),(0,2)\} \text{ and } Y^{[3]} = \{(3,0),(2,1),(1,2),(0,3)\},$$

as shown in Figure 1.

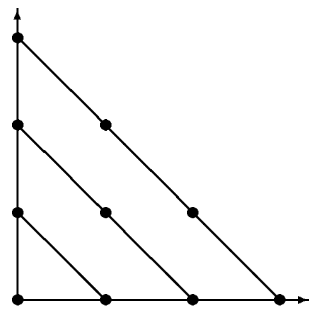

Figure 1: Interpolation set and the four blocks (connected with thick lines) for cubic interpolation on a regular grid in the plane.

The interpolating polynomial $m(x)$ is then given as

$$m(x) = \sum_{y_i^{[\ell]} \in Y} \lambda_i^{[\ell]}(Y, f) N_i^{[\ell]}(x),$$

where the coefficients $\lambda_i^{[\ell]}(Y, f)$ are generalized finite differences applied on f. We refer the readers to Sauer and Xu (1995) for more details on these and multivariate interpolation in general.

We will now return to the concept of poisedness. Thus we need to consider in more detail the procedure of constructing the basis of fundamental Newton polynomials as described in Sauer and Xu (1995). Namely we consider the procedure below for any given Y.

Procedure CNP for constructing fundamental Newton polynomials

Initialize the $N_i^{[\ell]}$ $(i = 1, \ldots, |Y^{[\ell]}|, \ell = 0, \ldots, d)$ to the chosen polynomial basis (the monomials).
Set $Y_{temp} = \emptyset$.
For $\ell = 0, \ldots, d$,
  for $i = 1, \ldots, |Y^{[\ell]}|$
    choose some $y_i^{[\ell]} \in Y \setminus Y_{temp}$ such that $|N_i^{[\ell]}(y_i^{[\ell]})| \neq 0$,
    if no such $y_i^{[\ell]}$ exists in $Y \setminus Y_{temp}$, reset $Y = Y_{temp}$ and stop
      (the basis of Newton polynomials is incomplete),
    $Y_{temp} \leftarrow Y_{temp} \cup \{y_i^{[\ell]}\}$
    normalize the current polynomial by

$$N_i^{[\ell]}(x) \leftarrow N_i^{[\ell]}(x)/|N_i^{[\ell]}(y_i^{[\ell]})|, \tag{3.6}$$

    update all Newton polynomials in block ℓ and above by

$$N_j^{[\ell]}(x) \leftarrow N_j^{[\ell]}(x) - N_j^{[\ell]}(y_i^{[\ell]}) N_i^{[\ell]}(x) \quad (j \neq i, j = 1, \ldots, |Y^{[\ell]}|), \tag{3.7}$$

$$N_j^{[k]}(x) \leftarrow N_j^{[k]}(x) - N_j^{[k]}(y_i^{[\ell]}) N_i^{[\ell]}(x) \quad (j = 1, \ldots, |Y^{[k]}|, k = \ell + 1, \ldots, d).$$

end

End (the basis of Newton polynomials is complete).

Clearly, poisedness relates to non-zero pivots in (3.6). Notice that after applying procedure CNP, Y is always poised since we only include the points that create non-zero pivots. This is true even if the procedure stops with an incomplete basis of Newton polynomials, which then results in an interpolating polynomial which is not of full degree d (meaning that it does not include contributions of all the monomials of degree d). In practice we need sufficiently large pivots, which is equivalent to "well-poisedness". Thus checking if $|N_i^{[\ell]}(y_i^{[\ell]})| \neq 0$ is replaced by $|N_i^{[\ell]}(y_i^{[\ell]})| \geq \theta$, for some $\theta > 0$. We call θ the pivot threshold. We will show that if throughout the algorithm the interpolation problem can be made sufficiently well-poised we are able to assure the existence of a bound on the distance between the interpolating polynomial and interpolated function at a point x ($x \notin Y$). Otherwise we provide a mechanism that guarantees we can find a suitable interpolation for which the bound holds. This bound depends upon an important property proved by Sauer and Xu and uses the concept of a *path* between the zero-th block and x, which uses a sequence of points of Y of the form

$$\pi(x) = (y_0^{[0]}, y_1^{[1]}, \ldots, y_d^{[d]}, y_{d+1}^{[d+1]} = x)$$

where

$$y_i^{[i]} \in Y^{[i]} \quad (i = 0, \ldots, d).$$

A path therefore contains, besides x itself, exactly one interpolation point in each block. Let us denote by $\Pi(x) = \{\pi(x)\}$, the set of all possible paths from $Y^{[0]}$ to x. Using this notion, Sauer and Xu (1995) derive in their Theorem 3.11 a bound on $|f(x) - m(x)|$, where $m(x)$ is the polynomial interpolating the function $f(x)$ at the points in Y. This bound was further simplified by Sauer (1996), giving that

$$|f(x) - m(x)| \leq \frac{n^{d+1} \|f^{(d)}\|_\infty}{(d+1)!} \sum_{\pi(x) \in \Pi(x)} \left[\prod_{i=0}^{d} \|y_{i+1}^{[i+1]} - y_i^{[i]}\|_\infty \, |N_i^{[i]}(y_{i+1}^{[i+1]})| \right], \tag{3.8}$$

for all x, where $f^{(d)}$ is the d-th derivative of f. Interestingly, the quantities $N_i^{[i]}(y_{i+1}^{[i+1]})$ are all computed in the course of the evaluation of the generalized finite differences $\lambda_i^{[\ell]}(Y, f)$. We see that the error between $m(x)$ and $f(x)$ is smaller if we can make the values $N_i^{[i]}(y_{i+1}^{[i+1]}) \|y_{i+1}^{[i+1]} - y_i^{[i]}\|_\infty$ small. This fact could possibly be exploited in a specific algorithm for computing the interpolation point y_i in a given domain, but we will here focus on the use of the bound (3.8), and thus of these values, for the purpose of proving convergence of an algorithm based on multivariate polynomial interpolation. We will further verify that, if all the interpolation points and the point x are chosen in a given hypersphere of radius δ, it is then possible to provide an upper bound on the maximal error, which will be of interest in the sequel of this paper. The following theorem establishes a relationship between well-poisedness and such a bound.

Theorem 1 *Suppose we are given $\theta > 0$, $Y \subset \mathcal{Q}_k(\delta)$, where $\mathcal{Q}_k(\delta) \stackrel{\text{def}}{=} \{y \mid \|y - x_k\| \leq \delta\}$ and the basis of Newton polynomials (possibly incomplete) which is built for Y using procedure CNP with θ as a pivot threshold. Then there exists a number $\kappa_n(\theta) > 0$ such that $|N_j^{[\ell]}(x)| \leq \kappa_n(\theta)$, for any $x \in \mathcal{Q}_k(\delta)$, for $l = 0, \ldots, d$ and $j = 1, \ldots, |Y^{[\ell]}|$.*

Proof. For simplicity we number the Newton polynomials (and the corresponding points) in this proof consecutively without regard to blocks. We will use an induction argument. We use procedure CNP with the N_j, $j = 1, \ldots, |Y|$ initialized by some polynomial basis (e.g. monomials). Thus on the bounded set $\mathcal{Q}_k(\delta)$, $|N_j(x)|$ are all bounded, say by M.
Induction step: Suppose we have successfully executed procedure CNP through i pivots and there exists $M > 0$ such that $|N_j(x)| < M$, $j = 1, \ldots, |Y|$ and for $x \in \mathcal{Q}_k(\delta)$. Consider the next step, first we normalize the $i+1$-st polynomial N_{i+1}, since we follow the pivot threshold rule, then $|N_{i+1}(x_{i+1})| > \theta$. Thus after the normalizing $|N_{i+1}(x)| < M/\theta$. After updating the other polynomials of the same block as in (3.7) we have $|N_j(x)| < M + M^2/\theta$. Therefore the Newton polynomials are still bounded on $\mathcal{Q}_k(\delta)$.
The induction is complete. The bound depends on the initial M, the threshold θ and the number of Newton polynomials. Since the last is finite, the bound is finite and we set $\kappa_n(\theta)$ to this bound.
□

Unfortunately $\kappa_n(\theta)$'s dependence upon θ means that it may become arbitrarily large as δ tends to zero. In practice δ is bounded away from zero but nevertheless to guarantee convergence we need to provide other means of maintaining the bound. This is the purpose of the next theorem.

Theorem 2 *Assume that an arbitrary $x_k \in \mathbf{R}^n$ and a $\delta > 0$ are given, together with the interpolation degree d. Then it is possible to construct an interpolation set Y yielding a complete basis of Newton polynomials such that all $y \in Y$ satisfy*

$$y \in \mathcal{Q}_k(\delta) \stackrel{\text{def}}{=} \{y \mid \|y - x_k\| \leq \delta\}, \tag{3.9}$$

and also that

$$|N_j^{[\ell]}(x)| \leq 2^{|Y^{[d]}|}. \tag{3.10}$$

for all $\ell = 0, \ldots, d$, all $j = 1, \ldots, |y^{[\ell]}|$ and all $x \in \mathcal{Q}_k(\delta)$.

Proof. We consider the procedure CNP but instead of choosing the interpolation points $y_i^{[\ell]}$ from a given set Y, we determine them successively by

$$y_i^{[\ell]} = \arg \max_{x \in \mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)|, \tag{3.11}$$

Note that, because of the independence of the polynomials in the basis, none of the fundamental polynomials can become identically zero, and thus that the value of the maximum in (3.11), and thus of the denominator in (3.6), is always nonzero. Applying this algorithm, we see that the normalization step ensures that

$$\max_{x \in \mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)| \leq 1, \tag{3.12}$$

except that this value may be modified when polynomials of the same block are updated by (3.7) later in the computation. Consider one such update: using (3.7) and (3.12), we have that

$$\max_{x\in\mathcal{Q}_k(\delta)} |\hat{N}_j^{[\ell]}(x)| \le \max_{x\in\mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)| + |N_j^{[\ell]}(y_i^{[\ell]})| \max_{x\in\mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)| \le 2 \max_{x\in\mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)|,$$

where $\hat{N}_j^{[\ell]}$ denote the j-th polynomial of the ℓ-th level after the update (3.7). Thus, at the end of the process and taking into account that (3.12) holds for $i = j$ when $N_j^{[\ell]}(x)$ is first normalized,

$$\max_{x\in\mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)| \le 2^{|Y^{[\ell]}|-j} \le 2^{|Y^{[\ell]}|} \le 2^{|Y^{[d]}|}, \tag{3.13}$$

as desired. □

The number of interpolation points in the last block, $|Y^{[d]}|$, only depends on the problem dimension n and the interpolation degree d. Notice that to construct an interpolation model of the form (3.2) we need $x_k \in Y$. This can be guaranteed by choosing $y_1^{[0]} = x_k$, since $N_1^{[0]}$ is a nonzero constant.

We now return to the case of quadratic models of the form (3.2) and define what we mean by an adequate geometry of the interpolation set. In the spirit of Theorem 2, we assume that we are at iteration k of the algorithm, where x_k is known (but arbitrary). We also assume that we are given a set $\mathcal{Q}_k(\delta)$ (as defined in (3.9)) for some $\delta > 0$. We then say that Y is *adequate in* $\mathcal{Q}_k(\delta)$ whenever the cardinality of Y is at least $n+1$ (which means the model is at least fully linear) and

$$y \in \mathcal{Q}_k(\delta) \ \text{ for all } \ y \in Y, \tag{3.14}$$

$$|N_i^{[\ell]}(y_j^{[\ell+1]})| \le \kappa_{\rm n} \quad (i = 1, \dots, |Y^{[\ell]}|, j = 1, \dots, |Y^{[\ell+1]}|, \ell = 0, \dots, d-1), \tag{3.15}$$

and

$$|N_i^{[d]}(x)| \le \kappa_{\rm n} \quad (i = 1, \dots, |Y^{[d]}|, x \in \mathcal{Q}_k(\delta)), \tag{3.16}$$

where $\kappa_{\rm n}$ is any positive constant such that $\kappa_{\rm n} > 2^{|Y^{[d]}|}$. (This choice of $\kappa_{\rm n}$ is merely intended to make (3.15) and (3.16) possible in view of Theorem 2.)

The value of $\kappa_{\rm n}$ is very large, except for small values of n and d, and only serves a theoretical purpose. In practice the threshold pivoting strategy may be a better way to verify the adequacy of the geometry because as long as the new point satisfies the threshold provision we can include it. The use of the procedure suggested in the proof of Theorem 2 may thus be viewed as a last resort if threshold pivoting rules do not prevent the $|N_j^{[\ell]}(y_j^{[\ell+1]})|$ from becoming very large. Interestingly, it is possible to prove that, if this procedure is applied for example to computing a linear model, then pivots cannot be arbitrarily small.

Lemma 3 *For given x_k and δ there exist $\theta > 0$ and Y adequate in $\mathcal{Q}_k(\delta)$, which can constructed by CNP with θ as the pivoting threshold.*

Proof. We will construct Y exactly as in Theorem 2. We only need to show that the pivot value in that case stays above a certain positive number. We initialize the Newton polynomial basis with the monomials. Then the first pivot equals 1. It is easy to see from the construction process of the basis and from the fact that we start with monomials that at the i-th pivot, for $0 < i \leq n$, the i-th polynomial is of the form $a_0 + a_1 x_1 + \ldots + a_{i-1} x_{i-1} + x_i$. Thus the maximum absolute value of this polynomial in $\mathcal{Q}_k(\delta)$ is at least δ. Therefore by choosing $0 < \theta < \delta$ we guarantee to build at least a full linear model in $\mathcal{Q}_k(\delta)$. Then from Theorem 2 we know that Y is adequate in $\mathcal{Q}_k(\delta)$. □

Note that one could equally use higher order polynomials and obtain a similar results as long as an appropriate relationship between θ and δ is chosen. For example if the highest monomial term is x_i^2 then one needs $0 < \theta < \min[\delta, \delta^2]/4$.

Unfortunately, this result is of little theoretical use, because it does not imply that, if an interpolation set Y such that $y \in \mathcal{Q}_k(\delta)$ for all $y \in Y$ is determined by the CNP procedure with threshold θ, then it is adequate in $\mathcal{Q}_k(\delta)$. This property would allow us to decide if a model is adequate just by considering the associated pivot values, an algorithmically desirable feature.

If Y is adequate in some ball centered at x_k, we may then combine (3.8) and Theorem 2 to obtain the following useful result.

Theorem 4 *Assume (A.1) and (A.3) hold, that we are using a model of the form (3.2) and that Y is adequate in $\mathcal{Q}_k(\delta)$ defined in (3.9). Then*

$$|f(x) - m_k(x)| \leq \kappa_{\rm em} \max[\delta^2, \delta^3], \tag{3.17}$$

and

$$\|\nabla f(x_k) - g_k\| \leq \kappa_{\rm eg} \max[\delta, \delta^2] \tag{3.18}$$

for all $x \in \mathcal{Q}_k(\delta)$ and some constants $\kappa_{\rm em}, \kappa_{\rm eg} > 0$ independent of k.

Proof. We consider the bound (3.8) for $d = 1$ and $d = 2$. Because of (A.1) and the definition of $\kappa_{\rm h}$, we obtain that the quantity $\|f^{(d)}\|$ is bounded above by $\kappa_{\rm h}$. Furthermore, since Y is adequate in $\mathcal{Q}_k(\delta)$, we may deduce that

$$\|y_i - x_k\| \leq \delta \quad (i = 1, \ldots, p),$$

and therefore that

$$\|y_i - y_j\| \leq 2\delta \quad (i, j = 1, \ldots, p, i \neq j).$$

Furthermore, again using the adequacy of Y in $\mathcal{Q}_k(\delta)$, we deduce that (3.15) and (3.16) hold. Combining these bounds with (3.8) for $d = 1$ and $d = 2$ then yields that

$$|f(x) - m_k(x)| \leq 2n^6 \kappa_{\rm h} \kappa_{\rm n} \max[\delta^2, \delta^3]$$

which obviously gives (3.17) with $\kappa_{\rm em} \stackrel{\rm def}{=} 2n^6 \kappa_{\rm h} \kappa_{\rm n}$.

If $g_k = \nabla f(x_k)$ then (3.18) trivially follows. Otherwise, since both the model m_k and the objective function are twice continuously differentiable by (A.1) and (A.3), Taylor's theorem gives, for every h satisfying the bound $\|h\| \leq \delta$, that

$$f(x_k + h) = f(x_k) + \langle \nabla f(x_k), h \rangle + \frac{1}{2} \langle h, \nabla^2 f(\xi_k) h \rangle$$

and

$$m_k(x_k + h) = f(x_k) + \langle g_k, h \rangle + \frac{1}{2}\langle h, H_k h \rangle,$$

where ξ_k belongs to the segment $[x_k, x_k + h]$. Taking the difference of these two equations, we deduce that

$$\langle \nabla f(x_k) - g_k, h \rangle = f(x_k + h) - m_k(x_k + h) - \frac{1}{2}\langle h, [\nabla^2 f(\xi_k) - H_k] h \rangle, \tag{3.19}$$

and therefore, using (3.17), (A.1), (A.3) and the Cauchy-Schwarz inequality, that

$$\begin{aligned} |\langle \nabla f(x_k) - g_k, h \rangle| &\leq \kappa_{\rm em} \max[\delta^2, \delta^3] + \tfrac{1}{2}|\langle h, \nabla^2 f(\xi_k) h \rangle| + \tfrac{1}{2}|\langle h, H_k h \rangle| \\ &\leq \kappa_{\rm em} \max[\delta^2, \delta^3] + \kappa_{\rm h} \delta^2. \end{aligned}$$

Choosing now

$$h = \delta \frac{\nabla f(x_k) - g_k}{\|\nabla f(x_k) - g_k\|},$$

we obtain that

$$\|\nabla f(x_k) - g_k\| \leq \kappa_{\rm em} \max[\delta, \delta^2] + \kappa_{\rm h} \delta \leq (\kappa_{\rm em} + \kappa_{\rm h}) \max[\delta, \delta^2].$$

This is (3.18) with $\kappa_{\rm eg} = \kappa_{\rm em} + \kappa_{\rm h}$. $\Box$

Before we state our algorithmic framework, we need to introduce how we consider updating the interpolation set. We will only use two operations on this set: the *insertion* of a given point x in Y and the *improvement* of the model's geometry in a given ball $\mathcal{Q}_k(\delta)$ for some $\delta > 0$.

In the first case (the *insertion* of x), we may simply add x to Y if $|Y| < p$, or we need to remove a point of Y, if $|Y|$ is already maximal ($|Y| = p$). Ideally, this point should be chosen to make the geometry of Y as good as possible. There are various ways to attempt to achieve this goal. For instance, one might choose to remove $y_i^{[\ell]}$ such that $|N_i^{[\ell]}(x)|$ is maximal, therefore trying to make the pivots as large as possible, but other techniques are possible and will not be discussed here. It is important to note that, since x is given irrespective of the geometry of the points already in Y, there is no guarantee that Y will remain adequate in $\mathcal{B}_k$ if it has that property before the insertion.

The second operation (the model's geometry *improvement* in $\mathcal{Q}_k(\delta)$) promotes making Y adequate in $\mathcal{Q}_k(\delta)$. There is no need to specify exactly how this is done for the purpose of our theory, except that we must guarantee that this property is obtained after a *finite number* of such improvements and that x_k must remain in Y. For instance, a reasonable strategy consists in first eliminating a point $y_i^{[\ell]} \in Y$ which is not in $\mathcal{Q}_k(\delta)$ (if such a point exists), and replacing it in Y by a point z which maximizes $N_i^{[\ell]}$ within $\mathcal{Q}_k(\delta)$. If no such exchange is possible, one may then consider replacing interpolation points in $Y \setminus \{x_k\}$ by new ones in order to ensure that (3.15) and (3.16) hold after a finite number of replacements. The proof of Theorem 2 guarantees that this is indeed possible whereas Theorem 1 attempts to do the same more efficiently. We refer the reader to [Powell, 1994b], Conn and Toint (1996) or Powell (1996) for more details on practical procedures for improving the model's geometry.

3.2 The algorithmic framework

We now describe our class of algorithms formally as follows. We assume that the constants

$$0 < \eta_0 \le \eta_1 < 1, \quad 0 < \gamma_0 \le \gamma_1 < 1 \le \gamma_2, \quad \epsilon_g > 0 \text{ and } 1 \le \mu$$

are given. We also use the notation

$$\mathcal{A}_k = \{y \in \mathbf{R}^n \mid \|y - x_k\| \le \mu \|g_k\|\}.$$

Derivative free trust-region algorithm

Step 0: Initializations.

Let x_s and $f(x_s)$ be given. Choose an initial interpolation set Y containing x_s and at least another point. Then determine $x_0 \in Y$ such that $f(x_0) = \min_{y_i \in Y} f(y_i)$. Choose an initial trust region radius $\Delta_0 > 0$. Set $k = 0$.

Step 1: Build the model.

Using the interpolation set Y, build a model $m_k(x_k + s)$, possibly restricting Y to a subset containing x_k, such that conditions (3.4) hold for the resulting Y. If $\|g_k\| \le \epsilon_g$ and Y is inadequate in $\mathcal{A}_k$, then *improve* the geometry until Y is adequate in $\mathcal{Q}_k(\delta_k)$ for some $\delta_k \in (0, \mu \|g_k\|]$.

Step 2: Minimize the model within the trust region.

Compute the point $x_k + s_k$ such that

$$m_k(x_k + s_k) = \min_{x \in \mathcal{B}_k} m_k(x). \tag{3.20}$$

Compute $f(x_k + s_k)$ and the ratio

$$\rho_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \tag{3.21}$$

Step 3: Update the interpolation set.

- If $\rho_k \ge \eta_1$, *insert* $x_k + s_k$ in Y, dropping one of the existing interpolation points if $|Y| = p$.
- If $\rho_k < \eta_1$ and Y is inadequate in $\mathcal{B}_k$, *improve* the geometry in $\mathcal{B}_k$.

Step 4: Update the trust-region radius.

- If $\rho_k \ge \eta_1$, then set
$$\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k]. \tag{3.22}$$
- If $\rho_k < \eta_1$ and Y is adequate in $\mathcal{B}_k$, then set
$$\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k] \tag{3.23}$$
- Otherwise, set $\Delta_{k+1} = \Delta_k$.

Step 5: Update the current iterate.

Determine $\hat{x}_k$ such that

$$f(\hat{x}_k) = \min_{\substack{y_i \in Y \\ y_i \neq x_k}} f(y_i). \tag{3.24}$$

Then, if

$$\hat{\rho}_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(\hat{x}_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq \eta_0, \tag{3.25}$$

set $x_{k+1} = \hat{x}_k$. Otherwise, set $x_{k+1} = x_k$. Increment k by one and go to Step 1.

End of algorithm

If the algorithm is able to progress, that is if (3.25) holds, we say that the iteration is *successful*. If (3.25) fails, then the iteration is *unsuccessful*. We denote the index set of all successful iterations by $\mathcal{S}$. In other words,

$$\mathcal{S} = \{k \mid \hat{\rho}_k \geq \eta_0\}. \tag{3.26}$$

We also define $\mathcal{R}$ to be the index set of all iterations where the trust region radius is reduced, that is

$$\mathcal{R} = \{k \mid \Delta_{k+1} < \Delta_k\}. \tag{3.27}$$

This description calls for several comments.

1. Our algorithmic specification is admittedly simplistic. It is enough to consider the methods proposed by Powell (1994b), Conn and Toint (1996) or Powell (1996) to be convinced that practical algorithms involve a number of additional features that enhance efficiency. We recall here that our purpose is merely to explore a framework in which convergence theory can be discussed, and the above will serve that goal.

2. Any further improvement in the model, compared to what the algorithm explicitly includes, is also possible. For instance, one might wish to include $x_k + s_k$ in Y, even if $\rho_k < \eta_1$, provided it does not deteriorate the quality of the model. Indeed, we have computed $f(x_k + s_k)$ and any such evaluation should be exploited if at all possible. One could also decide to perform a geometry improvement step if ρ_k is very small, indicating a bad fit of the model to the objective function. Or one could require that the model remains at least linear for all k. Any further decrease in function values obtained within these steps is then taken into account by Step 5.

3. The loop in Step 1 may be viewed as a model's geometry improvement inner iteration, whose purpose is to ensure that g_k, the information of first-order for the model, is not too far from the true first-order information for the objective function. This feature is close in spirit to a proposal of Carter (1991). It remains to see whether this inner iteration can be well-defined and finite.

   Other ways to ensure that the model's gradient is correctly approximated when it become small can of course be considered.

4. The trust-region management is merely outlined in our description, restricting a usually more involved procedure to its essentials.

5. We haven't mentioned a stopping test in the algorithm, which is acceptable for studying convergence properties. However, practical methods would require a stopping criterion. One possible test is to stop the calculation if either the trust-region radius falls below a certain threshold, or the model's gradient becomes sufficiently small and the geometry of the interpolation set is adequate.

6. We have assumed that the models are at most quadratic but this need not be so in practical implementations. For instance, Conn and Toint (1996) suggest the use of models of degree exceeding two and the theory can be readily extended to account for this. Models that are less than fully quadratic, as is allowed by our framework, are also beneficial when function evaluations are expensive since we are able to use them as soon as they are available.

7. The set $\mathcal{A}_k$ plays a role similar to that of the trust region $\mathcal{B}_k$, but is intended to monitor the first-order part of the model. The use of these two regions is reminiscent of the two regions proposed by Powell (1996). Note that the additional computational effort required to obtain an adequate geometry may be practically adjusted by a suitable choice of ϵ_g.

We conclude this section by stating four useful properties of our algorithm.

Lemma 5
(i) We have that, for all k, if $\rho_k \geq \eta_1$, then $\hat{\rho}_k \geq \eta_0$ and thus iteration k is successful.
(ii) If $k \in \mathcal{R}$, then Y is adequate in $\mathcal{B}_k$.
(iii) The conditions of Step 1 may be satisfied in a finite number of improvements of the geometry, unless $\nabla f(x_k) = 0$.
(iv) There can only be a finite number of iterations such that $\rho_k < \eta_1$ before the trust-region radius is reduced in (3.23).

Proof. If $\rho_k \geq \eta_1$ the mechanism of Step 3 implies that $x_k + s_k$ is added to the interpolation set Y. As a consequence, we have that

$$f(\hat{x}_k) \leq f(x_k + s_k)$$

because of the definition of $\hat{x}_k$ in (3.24). Thus

$$\hat{\rho}_k = \frac{f(x_k) - f(\hat{x}_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} = \rho_k \geq \eta_1 \geq \eta_0,$$

and conclusion (i) follows. Property (ii) immediately results from the definition of $\mathcal{R}$ and the mechanism of Step 4. Property (iv) follows from our definition of the model's improvement in $\mathcal{B}_k = \mathcal{Q}_k(\Delta_k)$. We must be more careful for (iii) because of the interaction between g_k and δ_k. Consider the following procedure. First choose a scalar $\nu \in (0, 1)$, set $g_k^{(0)} = g_k$ and determine Y such that it is adequate in $\mathcal{Q}_k(\nu\epsilon_g)$ (in a finite number of improvements of the geometry). If the resulting $g_k^{(1)}$ satisfies $\mu\|g_k^{(1)}\| \geq \nu\epsilon_g$, the procedure stops with $\delta_k = \nu\epsilon_g$. Otherwise, that is if $\mu\|g_k^{(1)}\| < \nu\epsilon_g$, determine Y such that it is adequate in $\mathcal{Q}_k(\nu^2\epsilon_g)$. Then again either the procedure stops or the radius of $\mathcal{Q}_k$ is again multiplied by ν, and so on. The only possibility for this procedure to be infinite (and to require an infinite number of improvements of the geometry) is if $\mu\|g_k^{(i)}\| < \nu^i\epsilon_g$ for all $i \geq 0$. But Y is adequate in each of the $\mathcal{Q}_k(\nu^i\epsilon_g)$ and therefore Theorem 4 implies that

$$\|\nabla f(x_k) - g_k^{(i)}\| \leq \kappa_{\rm eg}\nu^i\epsilon_g$$

for each $i \geq 0$, and thus, using the triangle inequality, that, for all $i \geq 0$,

$$\|\nabla f(x_k)\| \leq \|g_k^{(i)}\| + \|\nabla f(x_k) - g_k^{(i)}\| \leq \left(\frac{1}{\mu} + \kappa_{\rm eg}\right)\nu^i\epsilon_g.$$

Since $\nu \in (0,1)$, this implies that $\nabla f(x_k) = 0$, as desired. □

The first of these results indicates that the iteration has to be successful if the model fits the objective function well enough. The second says that iterations when the trust region is reduced correspond to adequate geometry. The third property indicates that if an infinite iteration occurs within Step 1, this must be because the current iterate is critical, which would then solve our problem. Note that in this case the number of successful iterations is finite. The fourth guarantees the overall coherence of the method in that it reduces the trust-region radius eventually, if improvements in the model's geometry are not enough to ensure progress.

4 A brief convergence analysis

In many ways, our algorithm shares the common features of all trust-region methods: the use of a model of the objective function, the role of the ratio ρ_k of achieved versus predicted reduction, most of the trust-region updating technique. However, it differs from this standard framework in three important aspects. The first is that the model and objective function gradients need not coincide. Indeed, the objective function gradient is unavailable and that of the model is derived from the interpolation conditions (3.4) which are not designed to yield a good approximation of the gradient at the current iterate (at variance with a technique that would use finite difference approximations). We thus have to verify the coherence of the model's gradient with that (unknown) of the objective function. The second important difference with respect to a more traditional trust-region algorithm is that the model is subject to a condition on the geometry of the interpolation points: for the model to be "valid" this condition must be satisfied. This implies that the model cannot be judged locally inappropriate (and, consequently, the trust-region radius reduced) before the geometry of the interpolation set is itself satisfactory. Hence the additional condition in the second part of Step 4. A third difference is also worth indicating: because new interpolation points may be computed in the geometry improvement phase of Steps 1 and 3, it is possible that the objective's value at one of these points is lower than that of the current iterate. Since function values are assumed to be expensive, this lower value is "exploited" by shifting the current iterate to the point with lowest value (in Step 5), which means that the algorithm may generate a new iterate, even if the model is bad (in the sense that ρ_k is small). These three differences are unfortunately enough to make the standard convergence theory inapplicable to our framework, and we have to reconsider it in detail. This is the object of this section.

Lemma 6 *At every iteration k, one has that*

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm mdc} \|g_k\| \min\left[\frac{\|g_k\|}{\kappa_{\rm h}}, \Delta_k\right] \tag{4.28}$$

for some constant $\kappa_{\rm mdc} \in (0,1)$ independent of k.

Proof. We do not prove this result in detail, as it is classical in the trust-region literature. It results from the fact that the value of the model at $x_k + s_k$ must be lower than that at the *Cauchy point*, which is defined as the minimizer of the model on the intersection of its steepest descent direction and the trust region $\mathcal{B}_k$. A proof that the bound (4.28) holds at the Cauchy point may be found in Moré (1983). □

We may conclude from Theorem 4 that the error between the objective and the model decreases at least quadratically with the trust-region radius. The smaller this radius becomes, the better the model approximates the objective, which should intuitively guarantee that minimizing the model within the trust region will also decrease the objective function, as desired. We next show that this intuition is vindicated, in that an iteration must be successful if the current iterate is not critical and the trust-region radius is small enough.

Lemma 7 *Assume that (A.1)–(A.3) hold. Assume furthermore that there exists a constant $\kappa_{\rm g} > 0$ such that $\|g_k\| \geq \kappa_{\rm g}$ for all k. Then there is a constant $\kappa_{\rm d} > 0$ such that*

$$\Delta_k > \kappa_{\rm d} \tag{4.29}$$

for all k.

Proof. Assume that k is the first k such that

$$\Delta_{k+1} \leq \gamma_0 \min\left[1, \frac{\kappa_{\rm mdc}\kappa_{\rm g}(1-\eta_1)}{\max[\kappa_{\rm h}, \kappa_{\rm em}]}\right]. \tag{4.30}$$

We first note that the mechanism of the algorithm then implies that $k \in \mathcal{R}$, and also, from (3.23), that

$$\Delta_k \leq \min\left[1, \frac{\kappa_{\rm mdc}\kappa_{\rm g}(1-\eta_1)}{\max[\kappa_{\rm h}, \kappa_{\rm em}]}\right]. \tag{4.31}$$

Observe now that the condition $\eta_1 \in (0,1)$ and the inequality $\kappa_{\rm mdc} < 1$ imply that

$$\kappa_{\rm mdc}(1-\eta_1) \leq 1. \tag{4.32}$$

Thus condition (4.31) and our assumption on g_k imply that

$$\Delta_k \leq \frac{\|g_k\|}{\kappa_{\rm h}}.$$

As a consequence, Lemma 6 immediately gives that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm mdc} \|g_k\| \min\left[\frac{\|g_k\|}{\kappa_{\rm h}}, \Delta_k\right] = \kappa_{\rm mdc}\|g_k\|\Delta_k. \tag{4.33}$$

On the other hand, we may apply Theorem 4 with $\mathcal{B}_k = \mathcal{Q}_k(\Delta_k)$, since $k \in \mathcal{R}$, and thus, by Lemma 5, Y is adequate in $\mathcal{B}_k$. We therefore deduce from (4.33), (3.17), (4.32) and (4.31) that

$$|\rho_k - 1| = \left|\frac{f(x_k+s_k) - m_k(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}\right| \le \frac{\kappa_{\rm em}}{\kappa_{\rm mdc}\|g_k\|}\max[\Delta_k^2, \Delta_k] \le \frac{\kappa_{\rm em}}{\kappa_{\rm mdc}\|g_k\|}\Delta_k \le 1 - \eta_1$$

where we have used (4.31) to deduce the penultimate inequality. Therefore, $\rho_k \ge \eta_1$. The mechanism of Step 4 then ensures that (3.22) is applied, which contradicts the fact that $k \in \mathcal{R}$. Hence there is no k such that (4.30) holds, which gives the desired conclusion with

$$\kappa_{\rm d} = \gamma_0 \min\left[1, \frac{\kappa_{\rm mdc}\kappa_{\rm g}(1-\eta_1)}{\max[\kappa_{\rm h}, \kappa_{\rm em}]}\right].$$

□

We therefore see that the radius cannot become too small as long as a critical point is not approached. This property is crucial in that it ensures that progress of the algorithm is always possible (except at critical points).

The above results are sufficient to analyze the criticality of the unique limit point of the sequence of iterates when there are only finitely many successful iterations.

Theorem 8 *Assume that (A.1)–(A.3) hold. Assume furthermore that there are only finitely many successful iterations. Then $x_k = x_*$ for k sufficiently large and $\nabla f(x_*) = 0$.*

Proof. If an infinite loop occurs within Step 1, the result follows from Lemma 5 (iii). Otherwise, the mechanism of the algorithm ensures that $x_* = x_{k_0+1} = x_{k_0+j}$ for all $j > 0$, where k_0 is the index of the last successful iteration. Moreover, since all iterations are unsuccessful for sufficiently large k, we obtain from Lemma 5, (i) that $\rho_k < \eta_1$, and (3.23) along with Lemma 5, (iv) thus implies that the sequence $\{\Delta_k\}$ converges to zero and, hence, that $\mathcal{R}$ contains at least an infinite subsequence $\{k_j\}$. Moreover, Lemma 5, (ii) implies that Y is adequate in $\mathcal{B}_{k_j}$ for all j. Assume now that $\nabla f(x_*) \neq 0$. Using the second part of Theorem 4 have that, for $k > k_0$,

$$\|\nabla f(x_*) - g_{k_j}\| \le \kappa_{\rm eg}\max[\Delta_k, \Delta_k^2],$$

which, since $\{\Delta_k\}$ converges to zero, then implies that $\|g_{k_j}\| \ge \kappa_{\rm g} = \frac{1}{2}\|\nabla f(x_*)\| > 0$ for $k \ge k_0$ sufficiently large. But Lemma 7 states that this situation is impossible, and hence $\nabla f(x_*) = 0$, as desired. □

Having proved the desired convergence property for the case where $\mathcal{S}$ is finite, we now restrict our attention to the case where there are infinitely many successful iterations. In this context, we start by proving that at least one accumulation point of the sequence of iterates (when it is infinite) must be critical. The intuition behind this result is that, if Δ_k is small enough, then the model should approximate the function well, due to Theorem 4, but at the same time, Δ_k cannot become so small as to prevent progress, as shown by Lemma 7: hence convergence should occur.

Theorem 9 *Assume that (A.1)–(A.3) hold. Then one has that*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{4.34}$$

Proof. Assume, for the purpose of deriving a contradiction, that, for all k,

$$\|g_k\| \geq \kappa_{\rm g} \tag{4.35}$$

for some $\kappa_{\rm g} > 0$ and consider a successful iteration of index k. The fact that $k \in \mathcal{S}$ and conditions (4.28), (4.35) and (4.29) and (A.1) then give that

$$f(x_k) - f(x_{k+1}) \geq \eta_0[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\rm mdc}\kappa_{\rm g}\eta_0 \min\left[\frac{\kappa_{\rm g}}{\kappa_{\rm h}}, \kappa_{\rm d}\right].$$

Summing now over all successful iterations from 0 to k, we obtain that

$$f(x_0) - f(x_{k+1}) \geq \sum_{\substack{j=0 \\ j\in\mathcal{S}}}^{k}[f(x_j) - f(x_{j+1})] \geq \sigma_k\kappa_{\rm mdc}\kappa_{\rm g}\eta_0 \min\left[\frac{\kappa_{\rm g}}{\kappa_{\rm h}}, \kappa_{\rm d}\right],$$

where σ_k is the number of successful iterations up to iteration k. But since there are infinitely many such iterations, we have that

$$\lim_{k\to\infty} \sigma_k = +\infty,$$

and the difference between $f(x_0)$ and $f(x_{k+1})$ is unbounded, which clearly contradicts the fact that, according to (A.2), the objective function is bounded below. Our assumption (4.35) must therefore be false, which yields (4.34). □

Lemma 10 *Assume that (A.1)-(A.3) hold and that $\{k_i\}$ is a subsequence such that*

$$\lim_{i\to\infty} \|g_{k_i}\| = 0. \tag{4.36}$$

Then one has that

$$\lim_{i\to\infty} \|\nabla f(x_{k_i})\| = 0. \tag{4.37}$$

Proof. By (4.36), $\|g_{k_i}\| \leq \epsilon_g$ for i sufficiently large, Lemma 5, (iii) and the mechanism of Step 1 ensures that Y is adequate in $\mathcal{Q}_{k_i}(\delta_{k_i})$ for i sufficiently large, where

$$\delta_{k_i} \leq \mu\|g_{k_i}\|. \tag{4.38}$$

Theorem 4 then allows us to deduce that, for i sufficiently large,

$$\|\nabla f(x_{k_i}) - g_{k_i}\| \leq \kappa_{\rm eg}\max[\delta_{k_i}, \delta_{k_i}^2] \leq \kappa_{\rm eg}\max[\mu\|g_{k_i}\|, \mu^2\|g_{k_i}\|^2] \leq \kappa_{\rm eg}\mu\|g_{k_i}\|,$$

where the middle inequality results from (4.38). As a consequence, we have that, for i sufficiently large,

$$\|\nabla f(x_{k_i})\| \leq \|g_{k_i}\| + \|\nabla f(x_{k_i}) - g_{k_i}\| \leq (1 + \kappa_{\rm eg}\mu)\|g_{k_i}\|.$$

The limit (4.36) and this last bound then give (4.37) □

Using this lemma, we obtain the following global convergence result.

Theorem 11 *Assume that (A.1)-(A.3) hold. Then there is at least a subsequence of successful iterates $\{x_k\}$ whose limit is a critical point, that is*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Proof. There is no loss of generality to restrict our attention to successful iterations. This result then immediately follows Theorems 9 and Lemma 10. □

Note that the critical limit of the subsequence may be at infinity.

This result corresponds to the usual first step in the convergence analysis of a trust-region algorithm. We now prove that *all* limit points of the sequence of iterates are critical.

Theorem 12 *Assume that (A.1)–(A.3) hold. Then every limit point x_* of the sequence $\{x_k\}$ is critical, that is $\nabla f(x_*) = 0$.*

Proof. Assume, for the purpose of establishing a contradiction, that there is a subsequence of successful iterates, indexed by $\{t_i\} \subseteq \mathcal{S}$, such that

$$\|\nabla f(x_{t_i})\| \geq \epsilon_0 > 0 \tag{4.39}$$

for some $\epsilon_0 > 0$ and for all i. Then, because of Lemma 10, we obtain that

$$\|g_{t_i}\| \geq 2\epsilon > 0 \tag{4.40}$$

for some $\epsilon > 0$ and for all i, sufficiently large. Without loss of generality, we may assume that

$$(2 + \kappa_{\rm eg}\mu)\epsilon \leq \frac{1}{2}\epsilon_0. \tag{4.41}$$

Theorem 9 then ensures the existence for each t_i, of a first successful iteration $\ell(t_i) > t_i$ such that $\|g_{\ell(t_i)}\| < \epsilon$. Denoting $\ell_i \stackrel{\rm def}{=} \ell(t_i)$, we thus obtain that there exist another subsequence of $\mathcal{S}$ indexed by $\{\ell_i\}$ such that

$$\|g_k\| \geq \epsilon \;\text{ for }\; t_i \leq k < \ell_i \;\text{ and }\; \|g_{\ell_i}\| < \epsilon, \tag{4.42}$$

for sufficiently large k. We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\rm def}{=} \{k \in \mathcal{S} \mid t_i \leq k < \ell_i\},$$

where t_i and ℓ_i belong to the two subsequences defined above. Using (4.28), the fact that $\mathcal{K} \subseteq \mathcal{S}$ and (4.42), we deduce that, for $k \in \mathcal{K}$,

$$f(x_k) - f(x_{k+1}) \geq \eta_0[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\rm mdc}\epsilon\eta_0 \min\left[\frac{\epsilon}{\kappa_{\rm h}}, \Delta_k\right]. \tag{4.43}$$

But the sequence $\{f(x_k)\}$ is monotonically decreasing and bounded below because of (A.2). Hence it is convergent and the left-hand side of (4.43) must tend to zero when k tends to infinity. This gives that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \Delta_k = 0.$$

As a consequence, the second term dominates in the minimum of (4.43) and we obtain that, for $k \in \mathcal{K}$ sufficiently large,

$$\Delta_k \leq \frac{1}{\kappa_{\rm mdc}\epsilon\eta_0}[f(x_k) - f(x_{k+1})].$$

We then deduce from this bound that, for i sufficiently large,

$$\|x_{t_i} - x_{\ell_i}\| \le \sum_{\substack{j=t_i \\ j\in\mathcal{K}}}^{\ell_i-1} \|x_j - x_{j+1}\| \le \sum_{\substack{j=t_i \\ j\in\mathcal{K}}}^{\ell_i-1} \Delta_j \le \frac{1}{\kappa_{\rm mdc}\epsilon\eta_0}[f(x_{t_i}) - f(x_{\ell_i})].$$

Using (A.2) and the monotonicity of the sequence $\{f(x_k)\}$ again, we see that the right-hand side of this inequality must converge to zero, and we therefore obtain that

$$\lim_{i\to\infty} \|x_{t_i} - x_{\ell_i}\| = 0. \tag{4.44}$$

Now

$$\|\nabla f(x_{t_i})\| \le \|\nabla f(x_{t_i}) - \nabla f(x_{\ell_i})\| + \|\nabla f(x_{\ell_i}) - g_{\ell_i}\| + \|g_{\ell_i}\|.$$

The first term of the right-hand side tends to zero because of the continuity of the gradient of f (A.1), and is thus bounded by ϵ for i sufficiently large. The third term is bounded by ϵ by (4.42). Using a reasoning similar to that of Lemma 10, we also deduce that the second term is bounded by $\kappa_{\rm eg}\mu\epsilon$ for i sufficiently large. As a consequence, we obtain from these bounds and (4.41) that

$$\|\nabla f(x_{t_i})\| \le (2 + \kappa_{\rm eg}\mu)\epsilon \le \frac{1}{2}\epsilon_0$$

for i large enough, which contradicts (4.39). Hence our initial assumption must be false and the theorem follows. □

5 Conclusions and perspectives

We have presented a broad class of derivative-free trust-region methods for unconstrained optimization, for which global convergence to first-order critical points can be proved. The definition of this class is inspired by the proposals of Powell (1994b), Conn and Toint (1996) and Powell (1996). The derivation of the convergence result is based a bound on the interpolation error provided by Sauer and Xu (1995).

It is clear to the authors that much additional work remains to specify which members of the class provide the best numerical performance. The class itself could also be extended, and we have provided suggestions for potentially useful generalizations. Finally, a challenging question is to examine if other bounds than that of Sauer and Xu could be used to derive similar convergence results.

6 Acknowledgements

The authors are pleased to acknowledge the helpful and friendly comments made by T. Sauer and C. De Boor on the use of Newton's polynomials. The second author was supported in part by an IBM Cooperative Fellowship. Thanks are also due to M. Roma for his kind support.

References

[Bongartz *et al.*, 1995] 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.

[Brent, 1973] R. P. Brent. *Algorithms for Minimization Without Derivatives.* Prentice-Hall, Engelwood Cliffs, USA, 1973.

[Callier and Toint, 1977] F. M. Callier and Ph. L. Toint. Recent results on the accelerating property of an algorithm for function minimization without calculating derivatives. In A. Prekopa, editor, *Survey of Mathematical Programming*, pages 369–376. Publishing House of the Hungarian Academy of Sciences, 1977.

[Carter, 1991] 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.

[Conn and Toint, 1996] A. R. Conn and Ph. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Gianessi, editors, *Nonlinear Optimization and Applications*, pages 27–47, New York, 1996. Plenum Publishing.

[De Boor and Ron, 1992] C. De Boor and A. Ron. Computational aspects of polynomial interpolation in several variables. *Mathematics of Computation*, 58(198):705–727, 1992.

[Dennis and Schnabel, 1983] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, Englewood Cliffs, USA, 1983.

[Dennis and Torczon, 1991] J. E. Dennis and V. Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.

[Griewank and Corliss, 1991] A. Griewank and G. Corliss. *Automatic Differentiation of Algorithms.* SIAM, Philadelphia, USA, 1991.

[Griewank, 1994] A. Griewank. Computational differentiation and optimization. In J. R. Birge and K. G. Murty, editors, *Mathematical Programming: State of the Art 1994*, pages 102–131, Ann Arbor, USA, 1994. The University of Michigan.

[Harwell Subroutine Library, 1995] Harwell Subroutine Library. *A catalogue of subroutines (release 12).* Advanced Computing Department, Harwell Laboratory, Harwell, UK, 1995.

[Moré, 1983] 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.

[Nelder and Mead, 1965] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[Powell, 1964] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 17:155–162, 1964.

[Powell, 1965] M. J. D. Powell. A method for minimizing a sum of squares of nonlinear functions without calculating derivatives. *Computer Journal*, 7:303–307, 1965.

[Powell, 1970a] M. J. D. Powell. A Fortran subroutine for solving systems of nonlinear algebraic equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 115–161, London, 1970. Gordon and Breach.

[Powell, 1970b] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114, London, 1970. Gordon and Breach.

[Powell, 1974] M. J. D. Powell. Unconstrained minimization algorithms without computation of derivatives. *Bollettino della Unione Matematica Italiana*, 9:60–69, 1974.

[Powell, 1975] M. J. D. Powell. A view of unconstrained minimization algorithms that do not require derivatives. *ACM Transactions on Mathematical Software*, 1(2):97–107, 1975.

[Powell, 1994a] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, volume 275, pages 51–67, Dordrecht, NL, 1994. Kluwer Academic Publishers.

[Powell, 1994b] M. J. D. Powell. A direct search optimization method that models the objective by quadratic interpolation. Presentation at the 5th Stockholm Optimization Days, 1994.

[Powell, 1996] M. J. D. Powell. Trust region methods that employ quadratic interpolation to the objective function. Presentation at the 5th SIAM Conference on Optimization, 1996.

[Sauer and Xu, 1995] Th. Sauer and Yuan Xu. On multivariate Lagrange interpolation. *Mathematics of Computation*, 64:1147–1170, 1995.

[Sauer, 1995] Th. Sauer. Computational aspects of multivariate polynomial interpolation. *Advances in Computational Mathematics*, 3:219–238, 1995.

[Sauer, 1996] Th. Sauer. Notes on polynomial interpolation. Private communication, February 1996.

[Toint and Callier, 1977] Ph. L. Toint and F. M. Callier. On the accelerating property of an algorithm for function minimization without calculating derivatives. *Journal of Optimization Theory and Applications*, 23(4):531–547, 1977. See also same journal, vol. 26(3), pp. 465–467, 1978.

[Toint, 1978] Ph. L. Toint. *Unconstrained optimization: the analysis of conjugate directions method without derivatives and a new sparse quasi-Newton update.* PhD thesis, Department of Mathematics, FUNDP, Namur, Belgium, 1978.

[Torczon, 1991] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991.