Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization

by K. Scheinberg[1] and Ph. L. Toint[2]

Report 09/06 2 February 2009

[1] IBM T.J. Watson Center
1101 Kitchawan Road, Yorktown Heights,
New York 10598, USA
(katyas@us.ibm.com)

[2] Department of Mathematics,
FUNDP-University of Namur,
61, rue de Bruxelles, B-5000 Namur, Belgium.
(philippe.toint@fundp.ac.be)

Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization

K. Scheinberg and Ph. L. Toint

2 February 2009

Abstract

Several efficient methods for derivative-free optimization (DFO) are based on the construction and maintenance of an interpolation model for the objective function. Most of these algorithms use special "geometry-improving" iterations, where the geometry (poisedness) of the underlying interpolation set is made better at the cost of one or more function evaluations. We show that such geometry improvements cannot be completely eliminated if one wishes to ensure global convergence, but also provide an algorithm where such steps only occur in the final stage of the algorithm where criticality of a putative stationary point is verified. Global convergence for this method is proved by making use of a self-correction mechanism inherent to the combination of trust regions and interpolation models. This mechanism also throws some light on the surprisingly good numerical results reported by Fasano, Nocedal and Morales (2009) for a method where no care is ever taken to guarantee poisedness of the interpolation set.

Keywords: derivative-free optimization, geometry of the interpolation set, unconstrained minimization.

1 Introduction

The past years have seen the emergence of model-based algorithms for optimization in the frequent case where the derivatives of the objective function are unavailable. Pioneered by Winfield (1969, 1973) and Powell (1994*a*, 1994*b*, 1998, 2003, 2004), they have been developped, for constrained and unconstrained problems, by a number of authors (see Conn and Toint, 1996, Conn, Scheinberg and Toint, 1997*a*, 1997*b*, Marazzi and Nocedal, 2002, Colson, 2004, Colson and Toint, 2001, 2002, Vanden Berghen and Bersini, 2005, Oeuvray, 2005, Driessen, 2006, Conn, Scheinberg and Vicente, 2008*a*, 2008*b*, 2008*c*, Oeuvray and Bierlaire, 2008, Wild, 2008, Wild, Regis and Shoemaker, 2008). Numerically efficient (Moré and Wild, 2007), they are widely used in practice (see, for instance, Conn, Scheinberg and Toint, 1998, Mugunthan, Shoemaker and Regis, 2005, Driessen, Brekelmans, Hamers and den Hertog, 2006 or Oeuvray and Bierlaire, 2007) and are of special interest when the cost of evaluating the objective function is high, as is for instance the case when this value is obtained by an expensive simulation process. Their details are discussed in Conn, Gould and Toint (2000) and, more specifically, in the very recent book by Conn, Scheinberg and Vicente (2008*d*).

These algorithms are based on the well-known trust-region methodology (see Conn et al., 2000 for an extensive coverage). They typically proceed by constructing a local polynomial interpolation-based model of the objective function. The interpolation is carried out by using previously computed function values, which are available for a subset of the past iterates and possibly at specially constructed points. For the interpolation process to be well-defined, the geometry of this set of points (the interpolation set) has to be satisfactory in the sense that, broadly speaking, all directions of the space have to be sufficiently well-covered. This property of the interpolation set is called poisedness. The algorithms then minimize the constructed interpolation model in a region (the trust region) where this model is believed to represent the objective function sufficiently well. A new function value is computed at this model minimizer and the predicted reduction in the model is compared to the achieved reduction in the real objective. If the ratio of these decreases is positive, the new point is accepted as the next iterate and the trust region is expanded, while the new point is rejected and the trust region (possibly) contracted in the latter case. Most algorithms for derivative-free optimization (DFO) crucially differ

from more standard trust-region schemes in that the decision to contract the trust region depends on the quality of the interpolation model. Indeed, if the interpolation set is not "sufficiently poised" (in a sense that we discuss below), then it may turn out that the failure of the current iteration is due to the poor characteristic of the resulting model rather than a too large trust region. The most common approach has therefore been to improve the poisedness of the interpolation set first, before considering contracting the trust region. This improvement is carried out at special "geometry improving" steps, which involve computing additional function values at well-chosen points. These special iterations are therefore expensive, and one is naturally led to wonder if they are truly necessary for the algorithm to be globally convergent (in the sense that convergence is guaranteed to a stationary point irrespective of the starting guess). In particular, it has been observed by Fasano et al. (2009) that an algorithm which simply ignores the geometry considerations may in fact perform quite well in practice.

The purpose of the present paper is to explore this question further. We will first show that it is impossible to ignore geometry considerations altogether if one wishes to maintain global convergence, but we will also provide an algorithm which resorts to the geometry-improving steps as little as possible, while still maintaining a mechanism for taking geometry into account. Interestingly, the design and convergence proof of this new algorithm crucially depends on a self-correction mechanism resulting from the combination of the trust-region mechanism with the polynomial interpolation setting. The main features of the new algorithm, in fact, are very similar to those in practical implementations, such as DFO (Conn et al., 1998) and NEWUOA (Powell, 2006, 2008) in that the new trust region trial point may be included to improve geometry of the interpolation set, if not the objective function value. Hence, the main objective of this paper is to advance the understanding of the role of geometry in model based DFO methods, rather than to suggest a new practical optimization scheme.

Our exposition is organized as follows. After recalling the DFO trust-region algorithm and some of the necessary concepts in Section 2, we discuss, in Section 3, two examples which show that completely ignoring the geometry of the interpolation set may result in convergence to a non-stationary point. The new algorithm is then presented in Section 4, while its self-correcting property and global convergence to first-order stationary points are analyzed in Section 5. Some discussion is finally provided in Section 6.

2 Interpolation models and trust-region methods

We consider the unconstrained minimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.1}$$

where the first derivatives of the objective function $f(x)$ are assumed to exist and be Lipschitz continuous. However, explicit evaluation of these derivatives is assumed to be impossible, either because they are unavailable (a situation frequently occuring when $f(x)$ is defined via a possibly complex simulation process) or because they are too costly.

In this paper, we consider model-based trust-region algorithms for computing local solutions of (2.1): these methods (which we formally describe below) iteratively use a local interpolation model of the objective to define a descent step, and adaptively adjust the region in which this model is deemed to be suitable. In order to understand their mechanism, we start by introducing the necessary definitions and properties in multivariate interpolation.

2.1 Polynomial interpolation and Lagrange Polynomials

Let us consider $\mathcal{P}_n^d$, the space of polynomials of degree $\leq d$ in $\mathbb{R}^n$ and let $p_1 = p+1$ be the dimension of this space. One knows that for $d=1$, $p_1 = n+1$ and that for $d=2$, $p_1 = \frac{1}{2}(n+1)(n+2)$. A basis $\Phi = \{\phi_0(x), \phi_1(x), \ldots, \phi_p(x)\}$ of $\mathcal{P}_n^d$ is a set of p_1 polynomials of degree $\leq d$ that span $\mathcal{P}_n^d$. For any such basis Φ, any polynomial $m(x) \in \mathcal{P}_n^d$ can be written as

$$m(x) = \sum_{j=0}^{p} \alpha_j \phi_j(x),$$

where the α_j's are real coefficients. We say that the polynomial $m(x)$ interpolates the function $f(x)$ at a given point y if $m(y) = f(y)$.

Assume now we are given a set $\mathcal{Y} = \{y_0, y_1, \ldots, y_p\} \subset \mathbb{R}^n$ of interpolation points, and let $m(x)$ denote a polynomial of degree d in $\mathbb{R}^n$ that interpolates a given function $f(x)$ at the points in $\mathcal{Y}$. The coefficients $\alpha_0, \ldots, \alpha_p$ can then be determined by solving the linear system

$$M(\Phi, \mathcal{Y})\alpha_\Phi = f(\mathcal{Y}),$$

where

$$M(\Phi, \mathcal{Y}) = \begin{bmatrix} \phi_0(y_0) & \phi_1(y_0) & \cdots & \phi_p(y_0) \\ \phi_0(y_1) & \phi_1(y_1) & \cdots & \phi_p(y_1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(y_p) & \phi_1(y_p) & \cdots & \phi_p(y_p) \end{bmatrix},$$

$$\alpha_\Phi = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} \quad \text{and} \quad f(\mathcal{Y}) = \begin{bmatrix} f(y_0) \\ f(y_1) \\ \vdots \\ f(y_p) \end{bmatrix}.$$

For the above system to have a unique solution, the matrix $M(\phi, \mathcal{Y})$ has to be nonsingular.

Definition 2.1 *The set $\mathcal{Y} = \{y_0, y_1, \ldots, y_p\}$ is poised for polynomial interpolation in $\mathbb{R}^n$ if the corresponding matrix $M(\Phi, \mathcal{Y})$ is nonsingular for some basis Φ in $\mathcal{P}_n^d$.*

The most commonly used measure of well-poisedness in the multivariate polynomial interpolation literature is based on Lagrange polynomials (Powell, 1994*b*).

Definition 2.2 *Given a set of interpolation points $\mathcal{Y} = \{y_0, y_1, \ldots, y_p\}$, a basis of $p_1 = p+1$ polynomials $\ell_j(x)$, $j = 0, \ldots, p$, in $\mathcal{P}_n^d$, is called a basis of Lagrange polynomials if*

$$\ell_j(y_i) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

If $\mathcal{Y}$ is poised, Lagrange polynomials exist, are unique and have a number of useful properties. We are in particular interested in the crucial fact that, if $m(x)$ interpolates $f(x)$ at the points of $\mathcal{Y}$, then, for all x,

$$m(x) = \sum_{j=0}^{p} f(y_i)\ell_j(x). \tag{2.2}$$

It can also be shown that

$$\sum_{j=0}^{p} \ell_j(x) = 1 \quad \text{for al} \quad x \in \mathbb{R}^n. \tag{2.3}$$

For more details and other properties of Lagrange polynomials see Section 3.2 in Conn et al. (2008*d*).

For our purposes we will need to consider and upper bound on their absolute value in a region $\mathcal{B}$ as a classical measure of poisedness of $\mathcal{Y}$ in $\mathcal{B}$. In particular, it is shown in Ciarlet and Raviart (1972) that for any x in the convex hull of $\mathcal{Y}$

$$\|\mathcal{D}^r f(x) - \mathcal{D}^r m(x)\| \leq \frac{\kappa_{\rm der}}{(d+1)!} \sum_{j=0}^{p} \|y_j - x\|^{d+1} \|\mathcal{D}^r \ell_j(x)\|,$$

where $\mathcal{D}^r$ denotes the r-th derivative of a function and $\kappa_{\rm der}$ is an upper bound on $\mathcal{D}^{d+1} f(x)$. We will make use of the following concept (borrowed from Conn et al., 2008*a*) of Λ–poisedness of an interpolation set.

Definition 2.3 *Let $\Lambda > 0$ and a set $\mathcal{B} \in I\!R^n$ be given. A poised set $\mathcal{Y} = \{y_0, y_1, \ldots, y_p\}$ is said to be Λ–poised in $\mathcal{B}$ if and only if, for the basis of Lagrange polynomials associated with $\mathcal{Y}$, one has that*

$$\Lambda \geq \max_{j=0,\ldots,p} \max_{x\in\mathcal{B}} |\ell_j(x)|.$$

An alternative way to define Lagrange polynomials is as follows. Given the set $\mathcal{Y}$ and any polynomial basis (of appropriate degree) Φ and a point x consider the sets $\mathcal{Y}_j(x) = \mathcal{Y} \setminus \{y_j\} \cup \{x\}$, $j = 0, \ldots, p$. Then

$$\ell_j(x) = \frac{\det(M(\Phi, \mathcal{Y}_j(x)))}{\det(M(\Phi, \mathcal{Y}))}. \tag{2.4}$$

Remarkably (and as noticed by Powell, 1998), the polynomial $\ell(x)$ does not depend on the choice of Φ as long as the polynomial space $\mathcal{P}_n^d$ is fixed. To help further understand the meaning of (2.4), consider a set $\Phi(\mathcal{Y}) = \{\phi(y_j), j = 0, \ldots, p\}$ in $I\!R^{p_1}$. Let vol$[\Phi(\mathcal{Y})]$ be the volume of the simplex whose vertices are the vectors of $\Phi(\mathcal{Y})$, given by

$$\text{vol}[\Phi(\mathcal{Y})] = \frac{|\det(M(\Phi, \mathcal{Y}))|}{p_1!}.$$

(Such a simplex is the p_1-dimensional convex hull of $\Phi(\mathcal{Y})$.) Then

$$|\ell_j(x)| = \frac{\text{vol}[\Phi(\mathcal{Y}_j(x))]}{\text{vol}[\Phi(\mathcal{Y})]}. \tag{2.5}$$

In other words, the absolute value of the j-th Lagrange polynomial at a given point x is the change in the volume of (the p_1-dimensional convex hull of) $\Phi(\mathcal{Y})$ when y_j is replaced by x in $\mathcal{Y}$. The following result can be derived using this definition.

Lemma 2.4 *Given a closed bounded domain $\mathcal{B}$, any initial interpolation set $\mathcal{Y} \in \mathcal{B}$ and a constant $\Lambda > 1$, consider the following procedure: find $j \in \{0, \ldots, p\}$ and a point $x \in \mathcal{B}$ such that $|\ell_j(x)| \geq \Lambda$ (if such a point exists), and replace y_j by x to obtain a new set $\mathcal{Y}$. Then this procedure terminates after a finite number of iterations with a model which is Λ–poised in $\mathcal{B}$.*

Proof. Fix any basis Φ and consider the volume vol$[\Phi(\mathcal{Y})]$. Since the initial set $\mathcal{Y} \in \mathcal{B}$ and remains in $\mathcal{B}$ after each point exchange, this volume is always uniformly bounded from above. Each time a point is replaced, the volume is increased by at least $\Lambda > 1$. Hence the factor by which this volume can be increased by a single point exchange has to eventually (after a finite number of such point exchanges) become smaller than Λ, and the procedure must stop in that a new point x with $|\ell_j(x)| \geq \Lambda$ can no longer be found in $\mathcal{B}$. □

The following two bounds will also be useful in our analysis below.

Lemma 2.5 *Given a ball $\mathcal{B}(x, \Delta) \stackrel{\text{def}}{=} \{v \in I\!R^n \mid \|v - x\| \leq \Delta\}$, a poised interpolation set $\mathcal{Y} \in \mathcal{B}(x, \Delta)$ and its associated basis of Lagrange polynomials $\{\ell_j(y)\}_{j=0}^p$, there exists constants $\kappa_{\text{ef}} > 0$ and $\kappa_{\text{eg}} > 0$ such that, for any interpolating polynomial $m(x)$ of degree one or higher of the form (2.2) and any given point $y \in \mathcal{B}(x, \Delta)$,*

$$\|f(y) - m(y)\| \leq \kappa_{\text{ef}} \sum_{j=0}^{p} \|y_i - y\|^2 |\ell_j(y)|$$

and

$$\|\nabla_x f(y) - \nabla_x m(y)\| \leq \kappa_{\text{eg}}\, \Lambda\, \Delta,$$

where $\Lambda = \max_{j=0,\ldots,p} \max_{x\in\mathcal{B}(x,\Delta)} |\ell_j(x)|$.

See Theorem 3.16, p. 59, in Conn et al., 2008*c*.

2.2 A trust-region framework

As announced, we may now use the interpolation polynomial $m(x)$ to define a local model of the objective function $f(x)$ of (2.1). This is accomplished in the framework of a trust-region algorithms. Such algorithms are iterative and build, around an iterate x_k, a model $m_k(x_k + s)$ of the objective function which is assumed to represent this latter function sufficiently well in a "trust region" $\mathcal{B}(x_k, \Delta_k)$, where Δ_k is known as the radius of the trust region. The model is then minimized (possibly approximately) in $\mathcal{B}(x_k, \Delta_k)$ to define a trial step x_k^+, and the value $f(x_k^+)$ is then computed. If this value achieves (a fraction of) the reduction from $f(x_k)$ which is anticipated on the basis of the model reduction $m_k(x_k) - m_k(x_k^+)$, then the trial point is accepted as the new iterate, the model is updated and the trust-region radius is possibly increased: this is a "successful iteration". If, on the contrary, the reduction in the objective function is too small compared to the predicted one, then the trial point is rejected and the trust-region radius is decreased: this is an unsuccessful iteration. (See Conn et al., 2000 for an extensive coverage of trust-region algorithms.) More formally, we start by considering the "simple" algorithm defined as Algorithm 1 and also considered by Fasano et al. (2009).

Algorithm 1: A simple DFO algorithm for unconstrained optimization

Step 0: Initialization. An initial trust-region radius Δ_0 is given. An initial poised interpolation set $\mathcal{Y}_0$ is known, that contains the starting point x_0. This interpolation set defines an (at most quadratic) interpolation model m_0 around x_0. Constants $\eta \in (0, 1)$ and $0 < \gamma_1 \leq \gamma_2 < 1$ are also also given. Set $k = 0$.

Step 1: Compute a trial point. Compute x_k^+ such that $\|x_k^+ - x_k\| \leq \Delta_k$ and $m_k(x_k^+)$ is "sufficiently small" compared to $m_k(x_k)$".

Step 2: Evaluate the objective function at the trial point. Compute $f(x_k^+)$ and

$$\rho_k \stackrel{\rm def}{=} \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}. \tag{2.6}$$

Step 3: Define the next iterate. Let $y_{k,\max} = \arg\max_{y \in \mathcal{Y}_k} \|y - x_k\|$.

Step3a: Successful iteration. If $\rho_k \geq \eta$, define $x_{k+1} = x_k^+$ and choose $\Delta_{k+1} \geq \Delta_k$. Set $\mathcal{Y}_{k+1} = \mathcal{Y}_k \setminus \{y_{k,\max}\} \cup \{x_k^+\}$

Step3b: Unsuccessful iteration. If $\rho_k < \eta$, then define $x_{k+1} = x_k$ and choose $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$. Set

$$\mathcal{Y}_{k+1} = \begin{cases} \mathcal{Y}_k & \text{if} \quad \|y_{k,\max} - x_k\| \leq \|x_k^+ - x_k\|, \\ \mathcal{Y}_k \setminus \{y_{k,\max}\} \cup \{x_k^+\} & \text{otherwise.} \end{cases}$$

Step 4: Update the (at most quadratic) model and Lagrange polynomials. If $\mathcal{Y}_{k+1} \neq \mathcal{Y}_k$, compute the interpolation model m_{k+1} around x_{k+1} using $\mathcal{Y}_{k+1}$. Increment k by one and go to Step 1.

This algorithm remains theoretical at this point, since we have not specified any practical stopping rule, nor have we said what we meant by "sufficiently small' compared to $m_k(x_k)$" (in Step 1). In our framework, one would typically compute x_k^+ by minimizing the model within $\mathcal{B}(x_k, \Delta_k)$, but our convregence analysis merely requires the weaker condition that

$$m_k(x_k) - m_k(x_k^+) \geq \kappa_{\rm C} \|g_k\| \min \left[\frac{\|g_k\|}{1 + \|H_k\|}, \Delta_k \right], \tag{2.7}$$

where we define $g_k \stackrel{\rm def}{=} \nabla_x m_k(x_k)$ and $H_k \stackrel{\rm def}{=} \nabla_{xx} m_k(x_k)$, and where $\kappa_{\rm C}$ is some constant in $(0,1)$. This condition is well-known in trust-region analysis under the name of "Cauchy condition", and indicates that the model reduction must be at least a fraction of that achievable along the steepest descent direction while remaining in the trust region. It is the cornerstone of convergence analysis for a very large number of trust-region-like methods.

3 Why considering geometry is necessary

Algorithm 1 is sufficient to examplify some of the potential difficulties arising with the use of interpolation models, and in particular, problems related to the (lack of) poisedness of the interpolation set. This is the object of this section, where we show, by two examples, that some geometry considerations are necessary in order to guarantee global convergence.

3.1 Example 1

The first example illustrates that ignoring geometry considerations completely may lead to degenerate models and, hence, to convergence to a non-stationary point.

Consider the following starting set of interpolation points:

$$\mathcal{Y}_0 = (y_{0,0}, y_{0,1}, y_{0,2}, y_{0,3}, y_{0,4}, y_{0,5}) = \left\{ \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 0 \end{pmatrix}, \begin{pmatrix} 10 \\ -1 \end{pmatrix}, \begin{pmatrix} 10 \\ 1 \end{pmatrix}, \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix} \right\}.$$

This set is Λ–poised, in a ball of radius 2 around $x = (10,0)^T$, with $\Lambda < 2.25$. Assume that we are given a function $f(x)$ for $x = (x_1, x_2)^T$ with the following function values on $\mathcal{Y}_0$:

$$\{121 + \alpha,\ 121,\ 100 + \alpha,\ 100 + \alpha,\ 100,\ 81\},$$

for some fixed $\alpha > 0$. Also assume that along the $x_2 = 0$ subspace the function $f(x)$ reduces to x_1^2 and has a minimum at $x_1 = 0$. For instance the simple function

$$f(x) = \begin{cases} x_1^2 + \alpha(x_2^2 + (10 - x_1)x_2) & \text{if } x_1 < 10; \\ x_1^2 + \alpha x_2^2 & \text{if } x_1 \geq 10, \end{cases}$$

has such properties. Note that this function has a discontinuous Hessian, however, $\nabla_x f(x)$ is Lipschitz continuous, so convergence to a first order stationary point is possible, as we show in Section 5. Also observe that it is possible to construct a function in C^2 with the same properties as $f(x)$.

Now let us consider a quadratic model based on $\mathcal{Y}_0$. It is easy to see that the model is

$$m(x) = x_1^2 + \alpha x_2^2.$$

Choose now a trust region of radius $\Delta = 2$ centered around $y_{0,4} = (10,0)^T$. If we minimize $m(x)$ in this trust region, then we obtain the trial point $x^+ = (8,0)^T$. Clearly, the predicted reduction in this case is $(81 - 64) = 17$ which is equal to the achieved reduction and the step is accepted. The trust region is moved (for simplicity we assume that the trust region radius is not increased) to be centered at $x^+ = (8,0)^T$ and the point in $\mathcal{Y}_0$ which is furthest away from the center is dropped. This point is $y_{0,1} = (11,1)^T$. Hence the new interpolation set is

$$\mathcal{Y}_1 = \left\{ \begin{pmatrix} 11 \\ 0 \end{pmatrix}, \begin{pmatrix} 10 \\ -1 \end{pmatrix}, \begin{pmatrix} 10 \\ 1 \end{pmatrix}, \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}.$$

and the function values on $\mathcal{Y}_1$ are:

$$\{121,\ 100 + \alpha,\ 100 + \alpha,\ 100,\ 81,\ 64\},$$

The set $\mathcal{Y}_1$ is not poised for quadratic interpolation, however it allows multiple interpolation models to interpolate $f(x)$ on it. The most natural one in this case is, again, $m(x) = x_1^2 + \alpha x_2^2$.

During the next step the model is optimized again and the new trial point $x^+ = (6,0)^T$ is obtained and accepted. The furstest point from $x^+ = (6,0)^T$, that is $y_{1,1} = (11,0)^T$, is dropped from the interpolation set and the new set becomes

$$\mathcal{Y}_2 = \left\{ \begin{pmatrix} 10 \\ -1 \end{pmatrix}, \begin{pmatrix} 10 \\ 1 \end{pmatrix}, \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 0 \end{pmatrix} \right\},$$

which is again non-poised, and again allows $m(x) = x_1^2 + \alpha x_2^2$ to interpolate $f(x)$ on $\mathcal{Y}_2$. Iterations are repeated in a similar manner to obtain the next interpolation set

$$\mathcal{Y}_3 = \left\{ \begin{pmatrix} 10 \\ 1 \end{pmatrix}, \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix} \right\},$$

and then, finally,

$$\mathcal{Y}_4 = \left\{ \begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\}.$$

At this point the interpolation set is completely alined with the direction $x_2 = 0$ and the model degenerates into $m(x) = x_1^2$. The algorithm then terminates at the point $x = (0,0)$, which is obtained at the next iteration and which is a non-stationary point for the original function $f(x)$. The original nonlinear problem as well as these iterates of our algorithm are shown in Figure 3.1 on the next page.

We see here that, if the gradient of the model converges to zero, it does not imply that so does the gradient of the true function, unless the poisedness of the interpolation set is maintained. Note that we have considered an extreme example where the interpolation set becomes non-poised. This is because we are interested in the convergence properties of the algorithm in infinite precision. However, if we consider the finite precision case and choose an appropriately small value of α in the above example, we can see that the Cauchy condition can still be satisfied by the points along (or near) the $x_2 = 0$ direction. Hence the iterates may be generated to produce badly poised models and the algorithm will terminate near a non-stationary point.

3.2 Example 2

Let us present another two-dimensional example, where due to selecting an exiting interpolation point solely based on its proximity to the current iterate results in a non-poised model and in convergence to a non-stationary point. Consider the function

$$f(x) = x_1^2 + 4(x_2 - \tfrac{1}{2})^2,$$

and the interpolation set

$$\mathcal{Y}_0 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}.$$

The interpolation values are $\{2,1,1\}$, and we now build a linear interpolation model which, in this case, is $m(x) = x_1 + 1$. Consider now a trust region of radius $\Delta_0 = \frac{1}{2}$ around the point $x = (0,0)^T$. The minimum of the model over the trust region is the point $x^+ = (-\frac{1}{2}, 0)^T$, at which the function value is $\frac{5}{4}$, and the model value is $m(x^+) = \frac{1}{2}$. The model reduction is $\frac{3}{2}$ and the function reduction is $\frac{3}{4}$. If $\eta < \frac{1}{2}$, the trial point is accepted as a new iterate and the trust region radius does not decrease. Let us assume it is doubled and hence is now equal to one. The new interpolation set (due to the removal of one of the furthest points $y_{0,1} = (1,0)^T$) may now be

$$\mathcal{Y}_1 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{1}{2} \\ 0 \end{pmatrix} \right\}.$$

The new model is then $m(x) = -\frac{1}{2}x_1 + 1$ and the minimum of the model in the trust region is $x^+ = (\frac{1}{2}, 0)$. It is easy to see that no function reduction is achieved by this step, and hence that the new iterate is not accepted. However it is closer to the current trust-region center $x = (-\frac{1}{2}, 0)^T$ than the interpolation

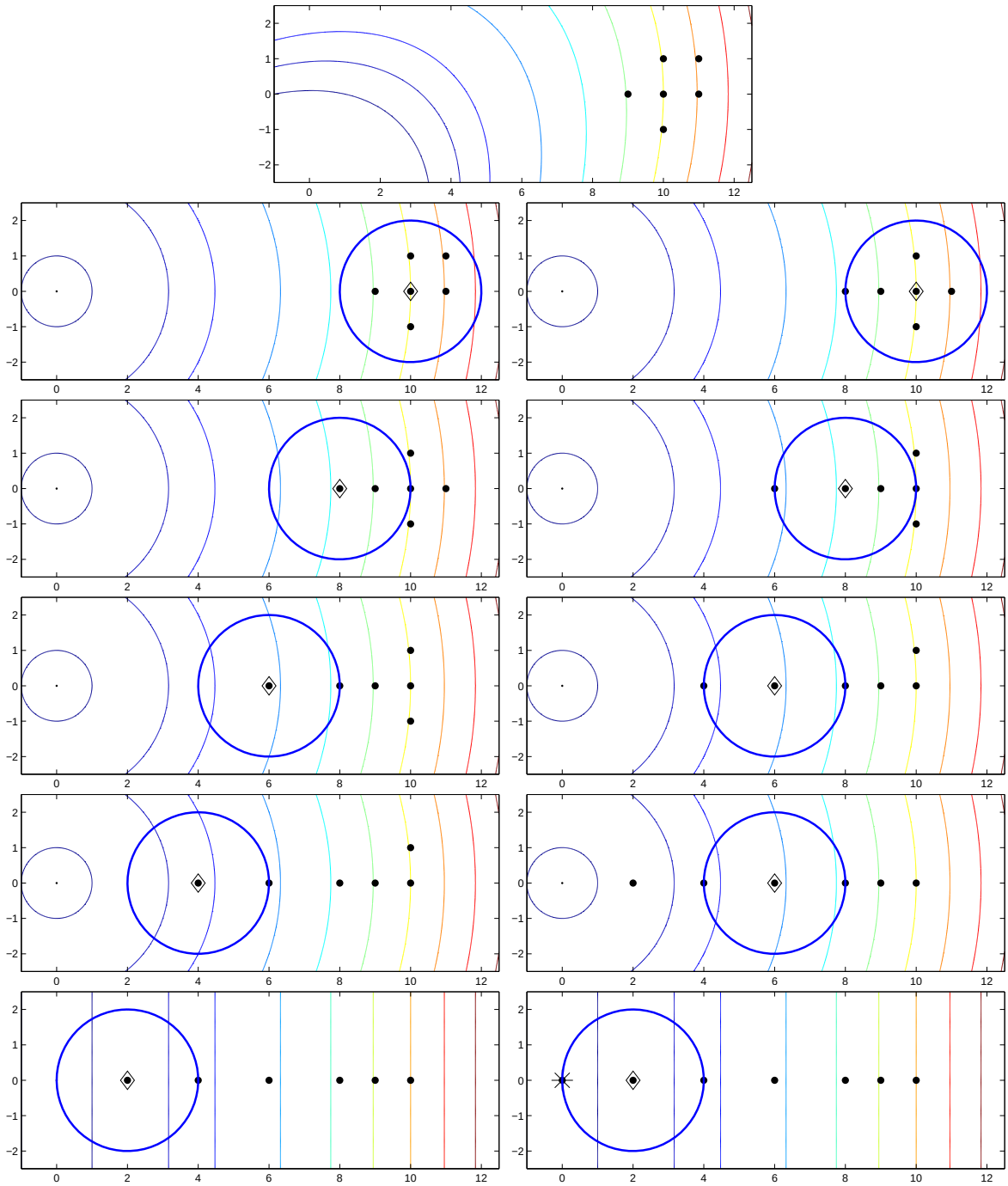

Figure 1: Top: level curves of the nonlinear objective function (for $\alpha = 1$) and the initial interpolation set; from left to right and top to bottom: the successive iterates of the algorithm on the associated models, where the current iterate is marked by a diamond and surrounded by its circular-shaped trust region. The final convergence point is indicated by a star.

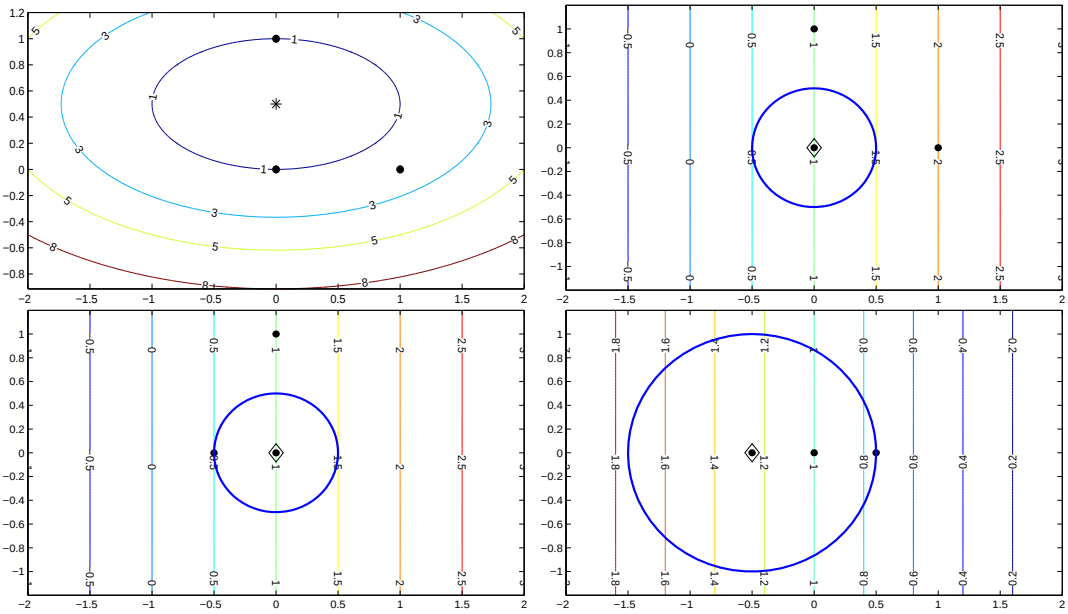

Figure 2: From left to right and top to bottom: the nonlinear objective function and the initial interpolation set, and the successive iterates of the algorithm on the associated, where the current iterate is marked by a diamond and surrounded by its circular-shaped trust region.

point $(0,1)^T$, and hence the point $(0,1)^T$ is replaced by the new point $x^+ = (0, \frac{1}{2})$. The interpolation set becomes

$$\mathcal{Y}_2 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\},$$

which is not poised for linear interpolation. In fact the new interpolation set does not span the x_2 dimension and hence the algorithm will not move away from the $x_2 = 0$ hyperplane. Since the minimum of $f(x)$ is at $\bar{x} = (0, \frac{1}{2})^T$ the algorithm will never converge to that point. These iterates are shown in Figure 3.2.

4 The new algorithm

Although we have just seen that we cannot afford to ignore geometry considerations altogether if we wish to maintain provable global convergence to first-order critical points, one may hope to reduce the frequency and cost of the necessary tests as much as possible. In what follows, we present an algorithm which, at variance with methods proposed in Conn et al. (1997*a*) or Conn et al. (2008*b*), exploits a self-correction property of this geometry which results from the combination of the trust-region mechanism with the geometry itself. This property is stated below in Lemma 5.2. The main idea of the new method is then to rely on new points generated by the algorithm to maintain poisedness of the interpolation set, with the understanding that some special care must be taken for the final criticality test and also that the geometry should be monitored, a task for which we use Lagrange polynomials. For the context to be well-defined, we formally state the new algorithm as Algorithm 2 on the next page.

In this algorithm, Step 1 is intended to provide a test for the criticality of the current iterate, which is verified (as we discuss in Theorem 5.8 below) whenever ϵ_i converges to zero, or, in a more practical setting, falls below some user-defined threshold. Steps 4b and 4c can be viewed as a variant of Step 3 in Algorithm 1, where the choice of the interpolation point to be replaced by x_k^+ is not only dependent of the distance to x_k, but also on the value of the Lagrange polynomials at x_k^+. Note also that, at variance with Algorithm 1, the trust-region radius is not reduced when an interpolation point is exchanged with the unsuccessful trial point. Finally observe that the exchange in Steps 4b and 4c is prevented when the

Algorithm 2: Another DFO algorithm for unconstrained optimization

Step 0: Initialization. An initial trust-region radius Δ_0 and an initial accuracy threshold ϵ_0 are given. An initial poised interpolation set $\mathcal{Y}_0$ is known, that contains the starting point x_0. This interpolation set defines an interpolation model m_0 around x_0 and associated Lagrange polynomials $\{\ell_{0,j}\}_{j=0}^p$. Constants $\eta \in (0,1)$, $0 < \gamma_1 \leq \gamma_2 < 1$, $\mu \in (0,1)$, $\theta > 0$, $\beta \geq 1$ and $\Lambda > 1$ are also also given. Choose $v_0 \neq x_0$ and set $k = 0$ and $i = 0$.

Step 1: Criticality test.

Step 1a: Define $\hat{m}_i = m_k$.

Step 1b: If $\|\nabla_x \hat{m}_i(x_k)\| < \epsilon_i$, set $\epsilon_{i+1} = \mu\|\nabla_x \hat{m}_i(x_k)\|$, compute a Λ–poised model $\hat{m}_{i+1}$ in $\mathcal{B}(x_k, \epsilon_{i+1})$, increment i by one and start Step 1b again.

Step 1c: Set $m_k = \hat{m}_i$, $\Delta_{k+1} = \theta\|\nabla_x m_k(x_k)\|$ and define $v_i = x_k$ if a new model has been computed.

Step 2: Compute a trial point. Compute x_k^+ such that (2.7) holds and $\|x_k^+ - x_k\| \leq \Delta_k$.

Step 3: Evaluate the objective function at the trial point. Compute $f(x_k^+)$ and ρ_k from (2.6).

Step 4: Define the next iterate.

Step 4a: Successful iteration. If $\rho_k \geq \eta$, define $x_{k+1} = x_k^+$, choose $\Delta_{k+1} \geq \Delta_k$ and define $\mathcal{Y}_{k+1} = \mathcal{Y}_k \setminus \{y_{k,r}\} \cup \{x_k^+\}$ for

$$y_{k,r} = \arg\max_{y_{k,j} \in \mathcal{Y}_k} \|y_{k,j} - x_k^+\|^2 |\ell_{k,j}(x_k^+)|. \tag{4.1}$$

Step 4b: Replace a far interpolation point. If $\rho_k < \eta$, either $x_k \neq v_i$ or $\Delta_k \leq \epsilon_i$, and the set

$$\mathcal{F}_k \stackrel{\rm def}{=} \{y_{k,j} \in \mathcal{Y}_k \ \text{ such that } \ \|y_{k,j} - x_k\| > \beta\Delta_k \quad \text{and} \quad \ell_{k,j}(x_k^+) \neq 0\}$$

is non-empty, then set $x_{k+1} = x_k$, $\Delta_{k+1} = \Delta_k$ and define $\mathcal{Y}_{k+1} = \mathcal{Y}_k \setminus \{y_{k,r}\} \cup \{x_k^+\}$ where r is an index of any point in $\mathcal{F}_k$, for instance, such that

$$y_{k,r} = \arg\max_{y_{k,j} \in \mathcal{F}_k} \|y_{k,j} - x_k^+\|^2 |\ell_{k,j}(x_k^+)|. \tag{4.2}$$

Step 4c: Replace a close interpolation point. If $\rho_k < \eta$, either $x_k \neq v_i$ or $\Delta_k \leq \epsilon_i$, the set $\mathcal{F}_k$ is empty, and the set

$$\mathcal{C}_k \stackrel{\rm def}{=} \{y_{k,j} \in \mathcal{Y}_k \setminus \{x_k\} \ \text{ such that } \ \|y_{k,j} - x_k\| \leq \beta\Delta_k \quad \text{and} \quad |\ell_{k,j}(x_k^+)| > \Lambda\}$$

is non-empty, then set $x_{k+1} = x_k$, $\Delta_{k+1} = \Delta_k$ and define $\mathcal{Y}_{k+1} = \mathcal{Y}_k \setminus \{y_{k,r}\} \cup \{x_k^+\}$ where r is an index of any point in $\mathcal{C}_k$, for instance, such that

$$y_{k,r} = \arg\max_{y_{k,j} \in \mathcal{C}_k} \|y_{k,j} - x_k^+\|^2 |\ell_{k,j}(x_k^+)|. \tag{4.3}$$

Step 4d: Reduce the trust-region radius. If $\rho_k < \eta$ and either $x_k = v_i$ and $\Delta_k > \epsilon_i$ or $\mathcal{F}_k \cup \mathcal{C}_k = \emptyset$, then set $x_{k+1} = x_k$, $\Delta_{k+1} \in [\gamma_1\Delta_k, \gamma_2\Delta_k]$ and define $\mathcal{Y}_{k+1} = \mathcal{Y}_k$.

Step 5: Update the model and Lagrange polynomials. If $\mathcal{Y}_{k+1} \neq \mathcal{Y}_k$, compute the interpolation model m_{k+1} around x_{k+1} using $\mathcal{Y}_{k+1}$ and the associated Lagrange polynomials $\{\ell_{k+1,j}\}_{j=0}^p$. Increment k by one and go to Step 1.

iterate has not moved away from a point v_i at which a well-poised model is known and the current trust region is larger that the model's radius of poisedness. The purpose of this restriction will become clear in Theorem 5.8 below.

5 Global convergence

We now show that Algorithm 2 produces a sequence of iterates $\{x_k\}$ such that the corresponding sequence of gradients of the true objective function $\{\nabla_x f(x_k)\}$ admits a subsequence converging to zero. We start by stating our assumptions.

A1: the objective function f is continuoulsy differentiable in an open set $\mathcal{V}$ containing all iterates generated by the algorithm, and its gradient $\nabla_x f$ is Lipschitz continuous in $\mathcal{V}$ with constant $\frac{1}{2}L$;

A2: there exists a constant $\kappa_{\rm low}$ such that $f(x) \geq \kappa_{\rm low}$ for every $x \in \mathcal{V}$;

A3: there exists a constant $\kappa_{\rm H} \geq L$ such that $1 + \|H_k\| \leq \kappa_{\rm H}$ for every $k \geq 0$.

Note that A1 merely assumes the existence of first derivatives, not that they can be computed. We will also use the following lemma.

Lemma 5.1 *Assume that, for some real numbers $\{\alpha_i\}_{i=1}^t$ with*

$$\sigma_{\rm abs} \stackrel{\rm def}{=} \sum_{i=0}^{t} |\alpha_i| > 2 \sum_{i=0}^{t} \alpha_i \stackrel{\rm def}{=} \sigma.$$

If one defines

$$i^* = \arg\max_{i=1\ldots,t} |\alpha_i| \quad \textit{and} \quad j^* = \arg\max_{\substack{j=1,\ldots,t \\ j\neq i^*}} |\alpha_j|.$$

then

$$|\alpha_{j^*}| \geq \frac{\sigma_{\rm abs} - 2\sigma}{2p}. \tag{5.1}$$

Proof. Given i^*, we consider three cases. Assume first $\alpha_{i^*} > \frac{1}{2}\sigma_{\rm abs}$. Then we have that

$$\sum_{\substack{j=0 \\ j\neq i^*}}^{t} \alpha_i = \sum_{j=0}^{t} \alpha_i - \alpha_{i^*} < \sigma - \tfrac{1}{2}\sigma_{\rm abs} < 0,$$

which implies that

$$\sum_{\substack{j=1 \\ j\neq i^*}}^{1} |\alpha_j| \geq \left| \sum_{\substack{j=1 \\ j\neq i^*}}^{t} \alpha_j \right| > \tfrac{1}{2}\sigma_{\rm abs} - \sigma,$$

and hence, for at least one $j \in \{0,\ldots,p\} \setminus \{i^*\}$, we have that $|\alpha_j| > (\frac{1}{2}\sigma_{\rm abs} - \sigma)/p$. Assume now that $|\alpha_{i^*}| \leq \frac{1}{2}\sigma_{\rm abs}$ then

$$\sum_{\substack{j=1 \\ j\neq i^*}}^{t} |\alpha_j| \geq \sigma_{\rm abs} - |\alpha_{i^*}| \geq \tfrac{1}{2}\sigma_{\rm abs} > \tfrac{1}{2}\sigma_{\rm abs} - \sigma,$$

and, for at least one $j \in \{0,\ldots,p\} \setminus \{i^*\}$, we must have that $|\alpha_j| > (\frac{1}{2}\sigma_{\rm abs} - \sigma)/p$. Assume finally that $\alpha_{i^*} \leq -\frac{1}{2}\sigma_{\rm abs}$. Then

$$\sum_{\substack{j=1 \\ j\neq i^*}}^{t} |\alpha_j| \geq \sum_{\substack{j=1 \\ j\neq i^*}}^{t} \alpha_j \geq \sigma - \alpha_{i^*} \geq \sigma + \tfrac{1}{2}\sigma_{\rm abs} > \tfrac{1}{2}\sigma_{\rm abs} - \sigma$$

and, again, must have that $|\alpha_j| > (\frac{1}{2}\sigma_{\rm abs} - \sigma)/p$ for at least one $j \in \{0,\ldots,p\} \setminus \{i^*\}$, which completes the proof. □

We now prove the crucial self-correction property of Algorithm 2.

Lemma 5.2 *Suppose that AS1 and AS3 hold, and that m_k is of degree one or higher. Then, for any constant $\Lambda > 1$, if iteration k is unsuccessful,*

$$\mathcal{F}_k = \emptyset \tag{5.2}$$

and

$$\Delta_k \leq \min\left[\frac{1}{\kappa_{\rm H}}, \frac{(1-\eta)\kappa_{\rm C}}{2\kappa_{\rm ef}(\beta+1)^2(p\Lambda+1)}\right] \|g_k\| \stackrel{\rm def}{=} \kappa_\Lambda \|g_k\|, \tag{5.3}$$

then

$$\mathcal{C}_k \neq \emptyset. \tag{5.4}$$

Proof. Assume iteration k is unsuccessful, which is to say that

$$\frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)} < \eta.$$

Now, because of the identity $f(x_k) = m_k(x_k)$, this in turn means that

$$f(x_k^+) > (1-\eta)\, m_k(x_k) + \eta\, m_k(x_k^+),$$

which then implies that

$$|f(x_k^+)) - m(x_k^+)| > (1-\eta)\, |m(x_k) - m(x_k^+)|. \tag{5.5}$$

We may now deduce from Theorem 2.5 (with $y = x_k^+$) that

$$|f(x_k^+) - m(x_k^+)| \leq \kappa_{\rm ef} \sum_{j=0}^{p} \|y_{k,j} - x_k^+\|^2 |\ell_{k,j}(x_k^+)|. \tag{5.6}$$

Observe now that (5.2) ensures that

$$\|y_{k,j} - x_k\| \leq \beta\Delta_k \quad \text{whenever} \quad \ell_{k,j} \neq 0. \tag{5.7}$$

This observation and the trust-region bound then imply that, for j such that $\ell_{k,j}(x_k^+) \neq 0$,

$$\|y_{k,j} - x_k^+\| \leq \|y_{k,j} - x_k\| + \|x_k^+ - x_k\| \leq (\beta+1)\Delta_k,$$

so that (5.5) and (5.6) then imply that

$$(1-\eta)\, |m(x_k) - m(x_k^+)| < |f(x_k^+) - m(x_k^+)| \leq \kappa_{\rm ef}(\beta+1)^2\Delta_k^2 \sum_{j=0}^{p} |\ell_{k,j}(x_k^+)|. \tag{5.8}$$

On the other hand, the Cauchy condition (2.7) and (5.3) together imply that

$$|m(x_k) - m(x_k^+)| \geq \kappa_{\rm C}||g_k||\Delta_k,$$

and hence (5.8) gives that

$$\sum_{j=0}^{p} |\ell_{k,j}(x_k^+)| \geq \frac{(1-\eta)\kappa_{\rm C}||g_k||}{\kappa_{\rm ef}(\beta+1)^2\Delta_k}. \tag{5.9}$$

As a consequence, we have, using (5.9) and (5.3) successively, that

$$\sum_{j=0}^{p} |\ell_{k,j}(x_k^+)| \geq 2(p\Lambda+1). \tag{5.10}$$

Moreover, (2.3) also implies that

$$\sum_{j=0}^{p} \ell_{k,j}(x_k^+) = 1.$$

We may then use this equality, (5.10) and Lemma 5.1 (with $\alpha_j = \ell_{k,j}(x_k^+)$ for $j = 0, \ldots, p$, and $\sigma_{\rm abs} = 2(p\Lambda + 1) > 2 = 2\sigma$) to deduce that, if $m = \arg\max_{j=0,\ldots,p} |\ell_{k,j}(x_k^+)|$, then

$$|\ell_{k,r}(x_k^+)| \geq \Lambda \quad \text{for} \quad r = \arg \max_{\substack{j=0,\ldots,p \\ j\neq m}} |\ell_{k,j}(x_k^+)|,$$

which then, together with (5.7), implies (5.4). □

This property essentially states that, provided the trust-region radius is small enough compared to the model's gradient and all the significant interpolation points are contained in the trust region, then every unsuccessful iteration must result in an improvement of the interpolation set geometry. The geometry is therefore *self-correcting at unsuccessful iterations* of this type. Moreover, the value of the geometry improvement is only dependent on Λ, while the maximum size of Δ_k compared with $\|g_k\|$ depends on the problem (via $\kappa_{\rm ef}$ and $\kappa_{\rm H}$), on the algorithms' parameters (via η, Λ and $\kappa_{\rm C}$) and on the size p of the interpolation set.

We now verify, as is usual in trust-region methods, that the step bound Δ_k cannot become arbitrarily small far away from a critical point.

Lemma 5.3 *Suppose that AS1 and AS3 hold and assume that, for some $k_0 \geq 0$ and all $k \geq k_0$, the model is of degree one or higher and*

$$\|g_k\| \geq \kappa_{\rm g} \tag{5.11}$$

for some $\kappa_{\rm g} > 0$. Then there exists a constant $\kappa_\Delta > 0$ such that, for all $k \geq k_0$,

$$\Delta_k \geq \kappa_\Delta. \tag{5.12}$$

Proof. Assume that, for some $k \geq 0$,

$$\Delta_k < \min(\kappa_\Lambda, \mu)\, \kappa_{\rm g}. \tag{5.13}$$

If, on one hand, iteration k is successful (i.e. $\rho_k \geq \eta$ and Step 4a is used), then we have that $\Delta_{k+1} \geq \Delta_k$. If, on the other hand, $\rho_k < \eta$, then we show that only two cases may occur. The first case is when $\mathcal{F}_k \neq \emptyset$. Observe now that, if $i > 0$, (5.11) and (5.13) ensure that

$$\Delta_k < \mu \|g_{k_i}\| = \epsilon_i, \tag{5.14}$$

where k_i is the index of the last iteration before k where a new Λ-poised model has been recomputed in Step 1. Step 4b is therefore executed and $\Delta_{k+1} = \Delta_k$. The second is when $\mathcal{F}_k = \emptyset$ in which case (5.13) and Lemma 5.2 guarantee that $\mathcal{C}_k \neq \emptyset$. Since (5.14) also hold in this case, Step 4c is executed and $\Delta_{k+1} = \Delta_k$. As a consequence, the trust-region radius may only be decreased if Δ_k is at least equal to $\min(\kappa_\Lambda, \mu)\kappa_{\rm g}$, and the mechanism of the algorithm then implies the desired result with $\kappa_\Delta = \min[\Delta_0, \gamma_1 \min(\kappa_\Lambda, \mu)\kappa_{\rm g}]$. □

This results allows us to continue the convergence analysis in the spirit of the standard trust-region theory (see Chapter 6 of Conn et al., 2000). We start by considering the case where the number of successful iterations is finite.

Lemma 5.4 *Suppose that AS1 and AS2 hold, that the model is of degree one or higher for all k sufficiently large and that there is a finite number of successful iterations. Then*

$$\liminf_{k\to\infty} ||g_k|| = 0. \tag{5.15}$$

Proof. Observe first that, since every iteration is eventually unsuccessful, $x_k = x_*$ for some x_* and the model is of degree one or higher and all k sufficiently large. Assume, for the purpose of deriving a contradiction, that (5.11) holds for some $\kappa_{\rm g} > 0$ and all k. Then, by Lemma 5.3, we have that $\Delta_k > \kappa_\Delta > 0$ on all iterations. Since the number of iterations of type 4a is finite, then eventually all iterations are of type 4b, 4c or 4d (no infinite loop within Step 1 is possible because $\|g_k\|$ is bounded away from zero and this step can be invoked only finitely many times). As a consequence the sequence $\{\Delta_k\}$ is non-increasing and bounded below, and therefore convergent. Let $\Delta_\infty \stackrel{\rm def}{=} \lim_{k\to\infty} \Delta_k \geq \kappa_\Delta$. Now iterations of type 4d cannot happen infinitely often because Δ_k is bounded below by Δ_∞ and $\gamma_2 < 1$. Thus $\Delta_k = \Delta_\infty$ for all k sufficiently large, and all iterations are eventually of type 4c since at most p such iterations can possibly be necessary to ensure that all interpolation points belong to $\mathcal{B}(x_*, \Delta_\infty)$. We must therefore have that, for all k large enough, the trial point x_k^+ replaces a previous interpolation point $y_{k,j}$ such that $|\ell_{k,j}(x_k^+)| \geq \Lambda$. But this is impossible in view of Lemma 2.4, which leads to the desired contradiction. □

We next turn to the case where there are infinitely many successful iterations.

Lemma 5.5 *Suppose that AS1–AS3 hold, that the model is of degree one or higher for all k sufficiently large and that the number of successful iteration is infinite. Then (5.15) holds.*

Proof. Assume, again that the lemma is not true, in that there exists some $\kappa_{\rm g} > 0$ such that (5.11) holds and the model is of degree one or higher for all k sufficiently large. Then Lemma 5.3 again implies that that (5.12) holds for all k, and in particular for all successful iterations with k large enough. But at every such iteration, we have, from (2.7), that

$$f(x_k) - f(x_{k+1}) \geq \eta(m_k(x_k) - m_k(x_k^+)) \geq \eta\kappa_{\rm C}\kappa_{\rm g} \min\left[\frac{\kappa_{\rm g}}{\kappa_{\rm H}}, \kappa_\Delta\right] \stackrel{\rm def}{=} \kappa_{\rm d} > 0,$$

Since there are infinitely many successful iterations, we deduce by summing up the corresponding inequalities that

$$\lim_{k\to\infty} f(x_k) = f(x_0) - \sum_{i=1}^{\infty} \kappa_{\rm d} = -\infty,$$

which contradicts AS2. Hence (5.11) cannot hold and the conclusion follows. □

We have shown that, eventually, the gradient of the model has to become smaller than ϵ_0. When this happens, the algorithm essentially restarts with a well-poised model in a sufficiently smaller ball. We then apply the same algorithm, but with the value ϵ_0 replaced by the smaller ϵ_1. Applying the same argument as above we can show that eventually $||g_k||$ will become smaller than ϵ_1 and the process repeats. To prove that this process leads to global convergence, we need the following additional two technical results.

Lemma 5.6 *Suppose that AS1 and AS3 hold. Then*

$$|f(x_k^+) - m_k(x_k^+)| \leq \|\nabla_x f(x_k) - g_k\| \, \Delta_k + \kappa_{\rm H}\Delta_k^2. \tag{5.16}$$

Proof. (See Theorem 8.4.2, page 285 in Conn et al., 2000.) Applying the mean-value theorem on the objective function, we deduce that

$$f(x_k^+) = f(x_k) + \langle \nabla_x f(x_k), x_k^+ - x_k\rangle + \int_0^1 \langle \nabla_x f(x_k + t(x_k^+ - x_k)) - \nabla_x f(x_k), x_k^+ - x_k\rangle \, dt,$$

and we also know that

$$m_k(x_k^+) = m_k(x_k) + \langle g_k, x_k^+ - x_k\rangle + \tfrac{1}{2}\langle x_k^+ - x_k, H_k(x_k^+ - x_k)\rangle.$$

Substracting these equalities, taking absolute values and using the Cauchy-Schwartz inequality together with the identity $f(x_k) = m_k(x_k)$, AS1, AS3 and the trust-region bound then give that

$$\begin{aligned} |f(x_k^+) - m_k(x_k^+)| &\leq |\langle \nabla_x f(x_k) - g_k, x_k^+ - x_k\rangle| + \tfrac{1}{2}|\langle x_k^+ - x_k, H_k(x_k^+ - x_k)\rangle| \\ &\quad + \|\nabla_x f(x_k + t(x_k^+ - x_k)) - \nabla_x f(x_k)\| \, \|x_k^+ - x_k)\| \\ &\leq \|\nabla_x f(x_k) - \nabla_x m_k(x_k)\| \, \Delta_k + \tfrac{1}{2}\kappa_{\rm H}\Delta_k^2 + \tfrac{1}{2}L\Delta_k^2, \end{aligned}$$

and (5.16) follows because $\kappa_{\rm H} \geq L$. □

Lemma 5.7 *Suppose that AS1 and AS3 hold, that $g_k \neq 0$, that*

$$\|\nabla_x f(x_k) - g_k\| \leq \tfrac{1}{2}\kappa_{\rm C}(1-\eta)\|g_k\| \tag{5.17}$$

and that

$$\Delta_k \leq \frac{\kappa_{\rm C}}{2\kappa_{\rm H}}(1-\eta)\|g_k\|. \tag{5.18}$$

Then iteration k is successful.

Proof. (See Theorem 8.4.3, page 286 in Conn et al., 2000.) Observe first that AS3, (5.18) and (2.7) imply that

$$m_k(x_k) - m_k(x_k^+) \geq \kappa_{\rm C}\|g_k\| \min\left[\frac{\|g_k\|}{\kappa_{\rm H}}, \Delta_k\right] = \kappa_{\rm C}\|g_k\|\Delta_k$$

Hence, successively using (2.6), this last inequality, (5.16), (5.17) and (5.18), we obtain that

$$|\rho_k - 1| \leq \left|\frac{f(x_k^+) - m_k(x_k^+)}{m_k(x_k) - m_k(x_k^+)}\right| \leq \left|\frac{\|\nabla_x f(x_k) - g_k\|}{\kappa_{\rm C}\|g_k\|}\right| + \left|\frac{\kappa_{\rm H}\Delta_k}{\kappa_{\rm C}\|g_k\|}\right| \leq 1 - \eta.$$

Thus we have that $\rho_k \geq \eta$ and iteration k is successful. □

We are now ready for our final result.

Theorem 5.8 *Suppose that AS1-AS3 hold and that the model is of degree one or higher for all k sufficiently large. Then*

$$\liminf_{k\to\infty} \|\nabla_x f(x_k)\| = 0. \tag{5.19}$$

Proof. Assume, by contradiction that there exists $\kappa_{\rm g} > 0$ such that

$$\|\nabla_x f(x_k)\| \geq \kappa_{\rm g} \tag{5.20}$$

for all k sufficiently large. Lemmas 5.4 and 5.5 show that, for any $\epsilon_i \in (0,1)$, Algorithm 2 will generate an iterate k_i such that $\|g_{k_i}\| \leq \epsilon_i$ (at the beginning of Step 1). The mechanism of Step 1 then implies that the sequence $\{k_i\}$ is infinite and that $\{\epsilon_i\}$ converges to zero. Let us now restrict our attention to i sufficiently large to ensure that

$$\epsilon_i \leq \frac{1}{2}\min\left[\frac{\kappa_{\rm C}(1-\eta)}{\kappa_{\rm eg}\Lambda}, \gamma_1\theta, \frac{\gamma_1\kappa_{\rm C}(1-\eta)}{2\kappa_{\rm H}}\right]\kappa_{\rm g}. \tag{5.21}$$

Then Lemma 2.5 ensures that, after Step 1 is executed at iteration k_i,

$$\|\nabla_x f(x_{k_i}) - g_{k_i}\| \leq \kappa_{\rm eg}\,\Lambda\,\epsilon_i \leq \tfrac{1}{2}\kappa_{\rm C}(1-\eta)\|\nabla_x f(x_{k_i})\| \leq \tfrac{1}{2}\|\nabla_x f(x_{k_i})\|, \tag{5.22}$$

where we used (5.20). and (5.21). Thus, after Step 1 is executed at iteration k_i,

$$\|g_{k_i}\| = \|\nabla_x f(x_{k_i}) + g_{k_i} - \nabla_x f(x_{k_i})\| \geq \|\nabla_x f(x_{k_i})\| - \|\nabla_x f(x_{k_i}) - g_{k_i}\| \geq \tfrac{1}{2}\kappa_{\rm g} \tag{5.23}$$

for i sufficiently large. As a consequence, no loop occurs within Step 1 for i large and we have that

$$\Delta_{k_i} = \theta\|g_{k_i}\| \geq \tfrac{1}{2}\theta\kappa_{\rm g} > \epsilon_i/\gamma_1 > \epsilon_i \tag{5.24}$$

where we used (5.21) to derive the penultimate inequality. Moreover, we have that $v_i = x_{k,i}$ at all iterations between k_i and the next successful iteration if any. This observation together with (5.24) imply that no iteration of type 4b or 4c may occur before the next successful iteration or before the trust-region radius becomes smaller than ϵ_i. Thus, either a successful iteration (type 4a) occurs, or the trust-region radius is decreased without altering the model (type 4d). This last case may happen for $j \geq 0$ as long as $\Delta_{k_i+j} > \kappa_{\rm C}(1-\eta)\kappa_{\rm g}/4\kappa_{\rm H}$, but Lemma 5.7 and (5.23) imply that a successful

iteration must occur as soon as this inequality is violated. As a consequence, a successful iteration $k_i + j_s$ must occur with

$$\Delta_{k_i+j_s} > \frac{\gamma_1 \kappa_{\rm C}(1-\eta)\kappa_{\rm g}}{4\kappa_{\rm H}} \stackrel{\rm def}{=} \Delta_{\rm min} \geq \epsilon_i, \tag{5.25}$$

where the last inequality results from (5.21). Moreover, since the model has not changed between iterations k_i and $k_i + j_s$, we have, from (5.23), that

$$\|g_{k_i+j_s}\| = \|g_{k_i}\| \geq \tfrac{1}{2}\kappa_{\rm g}. \tag{5.26}$$

Inserting (5.25) and (5.26) into (2.7) for the successful iteration $k_i + j_s$, we find, using (2.6) and $\rho_{k_i+j_s} \geq \eta$, that

$$f(x_{k_i+j_s}) - f(x_{k_i+j_s+1}) \geq \tfrac{1}{2}\eta\kappa_{\rm C}\kappa_{\rm g} \min\left[\frac{\frac{1}{2}\kappa_{\rm g}}{\kappa_{\rm H}}, \Delta_{\rm min}\right] > 0.$$

Since this scenario is repeated for all i large enough to ensure (5.21), we conclude, as in Lemma 5.5, that the objective function must be unbounded below, which is impossible in view of AS2. Thus our initial assumption that (5.20) holds for all sufficiently large k is itself impossible, and (5.19) follows. □

6 Discussion

We have shown in Theorem 5.8 that global convergence to first-order critical points may be achieved by a DFO algorithm without specific "geometry iterations". However, the poisedness of the interpolation set needs to be monitored (using Lagrange polynomials) and special care must be exercized to verify the first-order criticality of putative stationary points. Thus, the expenses of computing additional function values and of maximizing Lagrange polynomials in the trust region to find optimal new interpolation points, two main ingredients of "geometry steps", can (essentially) be spared, even if the linear algebra needed to maintain the Lagrange polynomials themselves remains (which does not increase the cost of maintaining an interpolation model).

The convergence guarantee of the new algorithm is a consequence of the remarkable self-correction property that the model's geometry must improve at unsuccessful iterations if the trust-region radius is sufficiently small compared to the model's gradient and all the interpolation points lie within the trust region. This property also throws some light as to why the simple method of Fasano et al. (2009) appears to perform well in practice, although we have seen that convergence cannot be ensured for this latter method. Indeed, one might expect that the numerical behaviour of the new method will be often similar to that of the simpler version. Thus the self-correcting property may also indirectly help the simpler method. This remains to be verified in details, but an in-depth investigation of this behaviour is outside the scope of the present contribution.

The algorithm and theory presented can be extended in many directions, as is possible for standard trust-region methods. The first would be to consider convergence to second-order critical points, which seems to be achievable at the cost of making the algorithm more complex and the assumptions stronger. We could also consider the case where the norms of the model's Hessians is no longer uniformly bounded, but tends to infinity slowly enough to ensure that the series of their inverses is divergent (see Chapter 8 in Conn et al., 2000). The adaptation of the algorithm and theory to optimization with convex constraints using projections (in the spirit of Chapter 12 of Conn et al., 2000) is also of interest, as are variants designed to handle noisy objective functions.

Acknowledgements
The authors are grateful to Jorge Nocedal for bringing the issue of avoiding geometry steps to their attention.

References

P. G. Ciarlet and P. A. Raviart. General Lagrange and Hermite interpolation in $\mathbb{R}^n$ with applications to finite element methods. *Arch. Ration. Mech. Anal.*, **46**, 177–199, 1972.

B. Colson. *Trust-Region Algorithms for Derivative-Free Optimization and Nonlinear Bilevel Programming*. PhD thesis, Department of Mathematics, FUNDP - University of Namur, Namur, Belgium, 2004.

B. Colson and Ph. L. Toint. Exploiting band structure in unconstrained optimization without derivatives. *Optimization and Engineering*, **2**, 349–412, 2001.

B. Colson and Ph. L. Toint. A derivative-free algorithm for sparse unconstrained optimization problems. *in* A. H. Siddiqi and M. Kočvara, eds, 'Trends in Industrial and Applied Mathematics', pp. 131–149, Dordrecht, The Netherlands, 2002. Kluwer Academic Publishers.

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

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 01 *in* 'MPS-SIAM Series on Optimization'. SIAM, Philadelphia, USA, 2000.

A. R. Conn, K. Scheinberg, and Ph. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. *in* A. Iserles and M. Buhmann, eds, 'Approximation Theory and Optimization: Tributes to M. J. D. Powell', pp. 83–108, Cambridge, England, 1997*a*. Cambridge University Press.

A. R. Conn, K. Scheinberg, and Ph. L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming, Series B*, **79**(3), 397–414, 1997*b*.

A. R. Conn, K. Scheinberg, and Ph. L. Toint. A derivative free optimization algorithm in practice. Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4, 1998.

A. R. Conn, K. Scheinberg, and L. N. Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming, Series B*, **111**(1-2), 2008*a*.

A. R. Conn, K. Scheinberg, and L. N. Vicente. Geometry of sample sets in derivative free optimization: polynomial regression and underdetermined interpolation. *IMA Journal of Numerical Analysis*, **28**(4), 721–748, 2008*b*.

A. R. Conn, K. Scheinberg, and L. N. Vicente. Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM Journal on Optimization*, **(to appear)**, 2008*c*.

A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-free Optimization*. MPS-SIAM Optimization series. SIAM, Philadelphia, USA, 2008*d*.

L. Driessen. *Simulation-based Optimization for Product and Process Design*. PhD thesis, University of Tilburg, Tilburg (NL), 2006.

L. Driessen, C. M. Brekelmans, H. Hamers, and D. den Hertog. On D-optimality based trust-regions for black-box optimization problems. *Structural and Multidisciplinary Optimization*, **31**, 40–48, 2006.

G. Fasano, J. Nocedal, and J.-L. Morales. On the geometry phase in model-based algorithms for derivative-free optimization. *Optimization Methods and Software*, **(to appear)**, 2009.

M. Marazzi and J. Nocedal. Wedge trust region methods for derivative free optimization. *Mathematical Programming, Series A*, **91**(2), 289–300, 2002.

J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. Technical Report ANL-MCS-P1471-1207, Mathematics and Computer Science, Argonne National Laboratory, Argonne, Illinois, USA, 2007.

P. Mugunthan, C. A. Shoemaker, and R. G. Regis. Comparison of function approximation, heuristic and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models. *Water Resources Research*, **41**, 2005.

R. Oeuvray. *Trust-Region Methods Based on Radial Basis Functions with Application to Biomedical Imaging*. PhD thesis, Institut de Mathématiques, École Polytechnique Fédérale de Lausanne, Switzerland, 2005.

R. Oeuvray and M. Bierlaire. A new derivative-free algorithm for the medical image restoration problem. *International Journal of Modelling and Simulation*, 2007.

R. Oeuvray and M. Bierlaire. BOOSTERS: A derivative-free algorithm based on radial basis functions. *International Journal of Modelling and Simulation*, **(to appear)**, 2008.

M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *in* S. Gomez and J. P. Hennart, eds, 'Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico', Vol. 275, pp. 51–67, Dordrecht, The Netherlands, 1994*a*. Kluwer Academic Publishers.

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

M. J. D. Powell. A quadratic model trust region method for unconstrained minimization without derivatives. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong, 1998.

M. J. D. Powell. On the use of quadratic models in unconstrained minimization without derivatives. Technical Report NA2003/03, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England, 2003.

M. J .D. Powell. Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming, Series B*, **100**(1), 183–215, 2004.

M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. *in* 'Large-Scale Nonlinear Optimization', Vol. 83, pp. 255–297. Springer, US, 2006.

M. J. D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA Journal of Numerical Analysis*, **28**(4), 649–664, 2008.

F. Vanden Berghen and H. Bersini. Performance of CONDOR, a parallel, constrained extension of Powell's UOBYQA algorithm: experimental results and comparison with the DFO algorithm. *J. Comput. Appl. Math.*, **181**, 157–175, 2005.

S. M. Wild. MNH: A derivative-free optimization algorithm using minimal norm Hessians. 2008.

S. M. Wild, R. G. Regis, and C. A. Shoemaker. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, **(to appear)**, 2008.

D. Winfield. *Function and functional optimization by interpolation in data tables*. PhD thesis, Harvard University, Cambridge, USA, 1969.

D. Winfield. Function minimization by interpolation in a data table. *Journal of the Institute of Mathematics and its Applications*, **12**, 339–347, 1973.