ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Global Convergence of Radial Basis Function Trust Region Derivative-Free Algorithms

Stefan M. Wild* **and Christine Shoemaker**†

Mathematics and Computer Science Division

Preprint ANL/MCS-P1580-0209

February 2009, *revised February 2011*

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. This work was supported by a DOE Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308, by an Argonne Director's Postdoctoral Fellowship, and by DOE under contract DE-AC02-06CH11357.

†School of Civil and Environmental Engineering and School of Operations Research and Information Engineering, Cornell University, Hollister Hall, Ithaca, NY 14853. This work was supported by NSF grants BES-022917, CBET-0756575, CCF-0305583, and DMS-0434390.

Contents

Global Convergence of Radial Basis Function Trust Region Derivative-Free Algorithms

Stefan M. Wild and Christine Shoemaker

Abstract

We analyze globally convergent derivative-free trust region algorithms relying on radial basis function interpolation models. Our results extend the recent work of Conn, Scheinberg, and Vicente to fully linear models that have a nonlinear term. We characterize the types of radial basis functions that fit in our analysis and thus show global convergence to first-order critical points for the ORBIT algorithm of Wild, Regis and Shoemaker. Using ORBIT, we present numerical results for different types of radial basis functions on a series of test problems. We also demonstrate the use of ORBIT in finding local minima on a computationally expensive environmental engineering problem.

1 Introduction

In this paper we analyze trust region algorithms for solving the unconstrained problem

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

using radial basis function (RBF) models. The deterministic real-valued function f is assumed to be continuously differentiable with a Lipschitz gradient ∇f and bounded from below, but we assume that all derivatives of f are either unavailable or intractable to compute. This paper is driven by work on the ORBIT (Optimization by Radial Basis functions In Trust regions) algorithm [31] and provides the key theoretical conditions needed for such algorithms to converge to first-order critical points. We find that the popular thin-plate spline RBFs do not fit in this globally convergent framework. Further, our numerical results show that the Gaussian RBFs popularly used in kriging [9, 16] are not as effective in our algorithms as alternative RBF types.

A keystone of the present work is our assumption that the computational expense of the function evaluation yields a bottleneck for classical techniques (the expense of evaluating the function at a single point outweighing any other expense or overhead of an algorithm). In some applications this could mean that function evaluation requires a few seconds on a state-of-the-art machine, up to functions that, even when parallelized, require several hours on a large cluster. The functions that drive our work usually depend on complex deterministic computer simulations, including those that numerically solve systems of PDEs governing underlying physical phenomena.

Derivative-free optimization of (1.1) has received renewed interest in recent years. Research has focused primarily on developing methods that do not rely on finite-difference estimates of the function's gradient or Hessian. These methods can generally be categorized into those based on systematic sampling of the function along well-chosen directions [1, 15, 17, 18, 19], and those employing a trust region framework with a local approximation of the function [6, 20, 23, 24, 25].

The methods in the former category are particularly popular with engineers for their ease of implementation and include the Nelder-Mead simplex algorithm [19] and pattern search [18]. These methods also admit natural parallel methods [15, 17] where different poll directions are sent to different processors for evaluation; hence these methods have also proved attractive for high-performance computing applications.

Methods in the latter category (including ORBIT) use prior function evaluations to construct a model, which approximates the function in a neighborhood of the current iterate. These models (for example, fully quadratic [6, 20, 24], underdetermined or structured quadratic [25], or radial basis functions [23, 31]) yield computationally attractive derivatives and are hence easy to optimize over within the neighborhood.

ORBIT is a trust region algorithm relying on an interpolating radial basis function model with a *linear polynomial tail* [31]. A primary distinction between ORBIT and the previously proposed RBF-based algorithm in [23] is the management of this interpolation set (Algorithm 3). In contrast to [23], the expense of our objective function allows us to effectively ignore the computational complexity of the overhead of building and maintaining the RBF model.

Our first goal is to show global convergence to first-order critical points for very general interpolation models. In Section 2 we review the multivariate interpolation problem, and show that the local error between the function (and its gradient) and an interpolation model (and its gradient) can be bounded using a simple condition on $n+1$ of the interpolation points. In the spirit of [7], we refer to such interpolation models as *fully linear*. In Section 3 we review derivative-free trust region methods and analyze conditions necessary for global convergence when fully linear models are employed. For this convergence analysis we benefit from the recent results in [8].

Our next goal is to use this analysis to identify the conditions that are necessary for obtaining a globally convergent trust region method using an interpolating RBF-based model. In Section 4 we introduce radial basis functions and the fundamental property of *conditional positive definiteness*, which we rely on in ORBIT to construct uniquely defined RBF models with bounded coefficients. We also give necessary and sufficient conditions for different RBF types to fit within our framework.

In Section 5 we examine the effect of selecting from three different popular radial basis functions covered by the theory by running the resulting algorithm on a set of smooth test functions. We also examine the effect of varying the maximum number of interpolation points. We motivate the use of ORBIT to quickly find locally minima of computationally expensive functions with an application problem (requiring nearly 1 CPU-hour per evaluation on a Pentium 4 machine) arising from detoxification of contaminated groundwater. We note that additional computational results, both on a set of test problems and on two applications from environmental engineering, as well as more practical considerations, are addressed in [31].

2 Interpolation Models

We begin our discussion on models that interpolate a set of scattered data with an introduction to the polynomial models that are heavily utilized by derivative-free trust region methods in the literature [6, 20, 24, 25].

2.1 Notation

We first collect the notation conventions used throughout the paper. $\mathbb{N}_0^n$ will denote n-tuples from the natural numbers including zero. A vector $x \in \mathbb{R}^n$ will be written in component form as $x = [\chi_1, \ldots, \chi_n]^T$ to differentiate it from a particular point $x_i \in \mathbb{R}^n$. For $d \in \mathbb{N}_0$, let $\mathcal{P}_{d-1}^n$ denote the space of n-variate polynomials of total degree no more than $d-1$, with the convention that $\mathcal{P}_{-1}^n = \emptyset$. Let $\mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\} \subset \mathbb{R}^n$ denote an interpolation set of $|\mathcal{Y}|$ points where (y_i, f_i) is known. For ease of notation, we will often assume interpolation relative to some base point $x_b \in \mathbb{R}^n$, made clear from the context, and will employ the set notation $x_b + \mathcal{Y} = \{x_b + y : y \in \mathcal{Y}\}$. We will work with a general norm $\|\cdot\|_k$ that we relate to the 2-norm $\|\cdot\|$ through a constant c_1, depending only on n, satisfying

$$\|\cdot\| \leq c_1 \|\cdot\|_k \quad \forall k. \tag{2.1}$$

The polynomial interpolation problem is to find a polynomial $P \in \mathcal{P}_{d-1}^n$ such that

$$P(y_i) = f_i, \qquad \forall y_i \in \mathcal{Y}, \tag{2.2}$$

for arbitrary values $f_1, \ldots, f_{|\mathcal{Y}|} \in \mathbb{R}$. Spaces where unique polynomial interpolation is always possible given an appropriate number of distinct data points are called *Haar spaces*. A classic theorem of Mairhuber and Curtis (cf. [28, p. 19]) states that Haar spaces do not exist when $n \geq 2$. Hence additional conditions are necessary for the multivariate problem (2.2) to be well-posed. We use the following definition.

Definition 2.1. *The points $\mathcal{Y}$ are $\mathcal{P}_{d-1}^n$-unisolvent if the only polynomial in $\mathcal{P}_{d-1}^n$ that vanishes at all points in $\mathcal{Y}$ is the zero polynomial.*

The monomials $\{\chi_1^{\alpha_1} \cdots \chi_n^{\alpha_n} : \alpha \in \mathbb{N}_0^n, \sum_{i=1}^n \alpha_i \leq d-1\}$ form a basis for $\mathcal{P}_{d-1}^n$, and hence any polynomial $P \in \mathcal{P}_{d-1}^n$ can be written as a linear combination of such monomials. In general, for a basis $\pi : \mathbb{R}^n \to \mathbb{R}^{\hat{p}}$ we will use the representation $P(x) = \sum_{i=1}^{\hat{p}} \nu_i \pi_i(x)$, where $\hat{p} = \dim \mathcal{P}_{d-1}^n = \binom{n+d-1}{n}$. Hence finding an interpolating polynomial $P \in \mathcal{P}_{d-1}^n$ is equivalent to finding coefficients $\nu \in \mathbb{R}^{\hat{p}}$ for which (2.2) holds.

Defining $\Pi \in \mathbb{R}^{\hat{p} \times |\mathcal{Y}|}$ by $\Pi_{i,j} = \pi_i(y_j)$, it follows that $\mathcal{Y}$ is $\mathcal{P}_{d-1}^n$-unisolvent if and only if Π is full rank, $\operatorname{rank}\Pi = \hat{p}$. Further, the interpolation in (2.2) is unique for arbitrary right-hand-side values $f_1, \ldots, f_{|\mathcal{Y}|} \in \mathbb{R}$ if and only if $|\mathcal{Y}| = \hat{p}$ and Π is invertible. In this case, the unique polynomial is defined by the coefficients $\nu = \Pi^{-T} f$.

One can easily see that existence and uniqueness of an interpolant are independent of the particular basis π employed. However, the conditioning of the corresponding matrix Π depends strongly on the basis chosen, as noted (for example) in [7].

Based on these observations, we see that in order to uniquely fit a polynomial of degree $d-1$ to a function, at least $\hat{p} = \dim \mathcal{P}_{d-1}^n = \binom{n+d-1}{n}$ function values must be known. When n is not very small, the computational expense of evaluating f to repeatedly fit even a quadratic (with $\hat{p} = \frac{(n+1)(n+2)}{2}$) is large.

2.2 Fully Linear Models

We now explore a class of *fully linear* interpolation models, which can be formed using as few as a linear (in the dimension, n) number of function values. Since such models are heavily tied to

Taylor-like error bounds, we will require assumptions on the function f as in this definition from [7].

Definition 2.2. *Suppose that* $\mathcal{B} = \{x \in \mathbb{R}^n : \|x - x_b\|_k \leq \Delta\}$ *and* $f \in C^1[\mathcal{B}]$. *For fixed* $\kappa_f, \kappa_g > 0$, *a model* $m \in C^1[\mathcal{B}]$ *is said to be* fully linear on $\mathcal{B}$ *if for all* $x \in \mathcal{B}$

$$|f(x) - m(x)| \leq \kappa_f \Delta^2, \tag{2.3}$$

$$\|\nabla f(x) - \nabla m(x)\| \leq \kappa_g \Delta. \tag{2.4}$$

This definition ensures that first-order Taylor-like bounds exist for the model within the compact neighborhood $\mathcal{B}$. For example, if $f \in C^1[\mathbb{R}]$, ∇f has Lipschitz constant γ_f, and m is the derivative-based linear model $m(x_b+s) = f(x_b)+\nabla f(x_b)^T s$, then m is fully linear with constants $\kappa_g = \kappa_f = \gamma_f$ on any bounded region $\mathcal{B}$.

Since the function's gradient is unavailable in our setting, our focus is on models that interpolate the function at a set of points:

$$m(x_b + y_i) = f(x_b + y_i) \qquad \text{for all } y_i \in \mathcal{Y} = \{y_1 = 0, y_2, \ldots, y_{|\mathcal{Y}|}\} \subset \mathbb{R}^n. \tag{2.5}$$

While we may have interpolation at more points, for the moment we work with a subset of exactly $n+1$ points and always enforce interpolation at the base point x_b so that $y_1 = 0 \in \mathcal{Y}$. The remaining n (nonzero) points compose the square matrix $Y = \left[\begin{array}{ccc} y_2 & \cdots & y_{n+1} \end{array}\right]$.

We can now state error bounds, similar to those in [7], for our models of interest.

Theorem 2.3. *Suppose that* f *and* m *are continuously differentiable in* $\mathcal{B} = \{x : \|x - x_b\|_k \leq \Delta\}$ *and that* ∇f *and* ∇m *are Lipschitz continuous in* $\mathcal{B}$ *with Lipschitz constants* γ_f *and* γ_m, *respectively. Further suppose that* m *satisfies the interpolation conditions in (2.5) at a set of points* $\{y_1 = 0, y_2, \ldots, y_{n+1}\} \subseteq \mathcal{B} - x_b$ *such that* $\|Y^{-1}\| \leq \frac{\Lambda_Y}{c_1 \Delta}$, *for a fixed constant* $\Lambda_Y < \infty$ *and* c_1 *from (2.1). Then for any* $x \in \mathcal{B}$,

$$|f(x) - m(x)| \leq \sqrt{n} c_1^2 (\gamma_f + \gamma_m) \left(\frac{5}{2}\Lambda_Y + \frac{1}{2}\right) \Delta^2 = \kappa_f \Delta^2, \tag{2.6}$$

$$\|\nabla f(x) - \nabla m(x)\| \leq \frac{5}{2}\sqrt{n}\Lambda_Y c_1 (\gamma_f + \gamma_m) \Delta = \kappa_g \Delta. \tag{2.7}$$

Proved in [29], Theorem 2.3 provides the constants $\kappa_f, \kappa_g > 0$ such that conditions (2.3) and (2.4) are satisfied, and hence m is fully linear in a neighborhood $\mathcal{B}$ containing the $n+1$ interpolation points. This result holds for very general interpolation models, requiring only a minor degree of smoothness and conditions on the points being interpolated. The conditions on the interpolation points are equivalent to requiring that the points $\{y_1, y_2, \ldots, y_{n+1}\}$ are sufficiently affinely independent (or equivalently, that the set $\{y_2 - y_1, \ldots, y_{n+1} - y_1\}$ is sufficiently linearly independent), with Λ_Y quantifying the degree of independence.

It is easy to iteratively construct a set of such points given a set of candidates (e.g. the points at which the f has been evaluated), $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\} \subset \mathcal{B} = \{x \in \mathbb{R}^n : \|x - x_b\|_k \leq \Delta\}$, using LU- and QR-like algorithms as noted in [7].

For example, in ORBIT, points are added to the interpolation set $\mathcal{Y}$ one at a time using a QR-like variant described in [31]. The crux of the algorithm is to add a candidate from $\mathcal{D}$ to $\mathcal{Y}$ if its projection onto the subspace orthogonal to span$\mathcal{Y}$ is sufficiently large (as measured by a constant $\theta \in (0,1]$). If the set of candidates $\mathcal{D}$ are not sufficiently affinely independent, such algorithms also produce points belonging to $\mathcal{B}$ that are perfectly conditioned with respect to the projection so that m can be easily made fully linear in fewer than n function evaluations.

We conclude this section by stating a lemma from [31] that ensures a QR like procedure like the one mentioned yields a set of points in $\mathcal{Y}$ satisfying $\left\|Y^{-1}\right\| \leq \frac{\Lambda_Y}{c_1\Delta}$.

Lemma 2.4. *Let $QR = \frac{1}{c_1\Delta}Y$ denote a QR factorization of a matrix $\frac{1}{c_1\Delta}Y$ whose columns satisfy $\left\|\frac{Y_j}{c_1\Delta}\right\| \leq 1$, $j = 1, \ldots, n$. If $r_{ii} \geq \theta > 0$ for $i = 1, \ldots, n$, then $\left\|Y^{-1}\right\| \leq \frac{\Lambda_Y}{c_1\Delta}$ for a constant Λ_Y depending only on n and θ.*

3 Derivative-Free Trust Region Methods

The interpolation models of the previous section were constructed to approximate a function in a local neighborhood of a point x_b. The natural algorithmic extensions of such models are trust region methods (given full treatment in [5]), whose general form we now briefly review.

Trust region methods generate a sequence of iterates $\{x_k\}_{k\geq 0} \subseteq \mathbb{R}^n$ by employing a surrogate model $m_k : \mathbb{R}^n \to \mathbb{R}$, assumed to approximate f within a neighborhood of the current x_k. For a (center, radius) pair $(x_k, \Delta_k > 0)$ we define the *trust region*

$$\mathcal{B}_k = \{x \in \mathbb{R}^n : \|x - x_k\|_k \leq \Delta_k\}, \tag{3.1}$$

where we distinguish the trust region norm (at iteration k), $\|\cdot\|_k$, from other norms used here. New points are obtained by solving subproblems of the form

$$\min_s \left\{m_k(x_k + s) : x_k + s \in \mathcal{B}_k\right\}. \tag{3.2}$$

The pair (x_k, Δ_k) is then updated according to the ratio of actual to predicted improvement. Given a maximum radius $\Delta_{\max}$, the design of the trust region algorithm ensures that f is sampled only within the relaxed level set:

$$\mathcal{L}(x_0) = \{y \in \mathbb{R}^n : \|x - y\|_k \leq \Delta_{\max} \text{ for some } x \text{ with } f(x) \leq f(x_0)\}. \tag{3.3}$$

Hence one really requires only that f be sufficiently smooth within $\mathcal{L}(x_0)$.

When exact derivatives are unavailable, smoothness of the function f is no longer sufficient for guaranteeing that a model m_k approximates the function locally. Hence the main difference between classical and derivative-free trust region algorithms is the addition of safeguards to account for and improve models of poor quality.

Historically (see [6, 20, 24, 25]), the most frequently used model is a quadratic,

$$m_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s, \tag{3.4}$$

the coefficients g_k and H_k being found by enforcing interpolation as in (2.5). As discussed in Section 2, these models rely heavily on results from multivariate interpolation. Quadratic models are attractive in practice because the resulting subproblem in (3.2), for a 2-norm trust region, is one of the only nonlinear programs for which *global* solutions can be efficiently computed.

A downside of quadratic models in our computationally expensive setting is that the number of interpolation points (and hence function evaluations) required is quadratic in the dimension of the problem. Noting that it may be more efficient to use function evaluations for forming subsequent models, Powell designed his NEWUOA code [25] to rely on least-change quadratic models interpolating fewer than $\frac{(n+1)(n+2)}{2}$ points. Recent work in [10, 12] has also explored loosening the restrictions of a quadratic number of geometry conditions.

3.1 Fully Linear Derivative-Free Models

Recognizing the difficulty (and possible inefficiency) of maintaining geometric conditions on a quadratic number of points, we will focus on using the fully linear models introduced in Section 2. These models can be formed with a linear number of points while still maintaining the local approximation bounds in (2.3) and (2.4).

We will follow the recent general trust region algorithmic framework introduced for linear models by Conn et al. [8] in order to arrive at a similar convergence result for the types of models considered here. Given standard trust region inputs $0 \leq \eta_0 < \eta_1 < 1$, $0 < \gamma_0 < 1 < \gamma_1$, $0 < \Delta_0 \leq \Delta_{\max}$, and $x_0 \in \mathbb{R}^n$ and constants $\kappa_d \in (0,1), \kappa_f > 0, \kappa_g > 0, \epsilon > 0, \mu > \beta > 0, \alpha \in (0,1)$, the general first-order derivative-free trust region algorithm is shown in Algorithm 1. This algorithm is discussed in [8], and we note that it forms an infinite loop, a recognition that termination in practice is a result of exhausting a budget of expensive function evaluations.

A benefit of working with more general fully linear models is that they allow for nonlinear modeling of f. Hence, we will be interested primarily in models with nontrivial Hessians, $\nabla^2 m_k \neq 0$, which are uniformly bounded by some constant κ_H.

The sufficient decrease condition that we will use in Step 1.2 then takes the form

$$m_k(x_k) - m_k(x_k + s) \geq \frac{\kappa_d}{2} \|\nabla m_k(x_k)\| \min\left\{\frac{\|\nabla m_k(x_k)\|}{\kappa_H}, \frac{\|\nabla m_k(x_k)\|}{\|\nabla m_k(x_k)\|_k}\Delta_k\right\} \tag{3.7}$$

for some prespecified constant $\kappa_d \in (0,1)$. This condition is similar to those found in the trust region setting when general norms are employed [5]. We note that the following lemma guarantees we will always be able to find an approximate solution, s_k, to the subproblem (3.2) that satisfies condition (3.7).

Lemma 3.1. *If $m_k \in C^2(\mathcal{B}_k)$ and $\kappa_H > 0$ satisfies*

$$\infty > \kappa_H \geq \max_{x \in \mathcal{B}_k} \left\|\nabla^2 m_k(x)\right\|, \tag{3.8}$$

then for any $\kappa_d \in (0,1)$ there exists an $s \in \mathcal{B}_k - x_k$ satisfying (3.7).

Lemma 3.1 (proved in [29]) is our variant of similar ones in [5] and describes a back-tracking line search algorithm to obtain a step that yields a model reduction at least a fraction of that achieved by the Cauchy point. As an immediate corollary we have that there exists $s \in \mathcal{B}_k - x_k$ satisfying (3.7) such that

$$\|s\|_k \geq \min\left\{\Delta_k, \kappa_d \frac{\|\nabla m_k(x_k)\|_k}{\kappa_H}\right\}, \tag{3.9}$$

and hence the size of this step is bounded from zero if $\|\nabla m_k(x_k)\|_k$ and Δ_k are.

Algorithm 1 Iteration k of a first-order (fully linear) derivative-free algorithm [8].

1.1. Criticality test If $\|\nabla m_k(x_k)\| \leq \epsilon$ and either m_k is not fully linear in $\mathcal{B}_k$ or $\Delta_k > \mu \|\nabla m_k(x_k)\|$:

Set $\tilde{\Delta}_k = \Delta_k$ and make m_k fully linear on $\{x : \|x - x_k\|_k \leq \tilde{\Delta}_k\}$.

While $\tilde{\Delta}_k > \mu \|\nabla m_k(x_k)\|$:

Set $\tilde{\Delta}_k \leftarrow \alpha\tilde{\Delta}_k$ and make m_k fully linear on $\{x : \|x - x_k\|_k \leq \tilde{\Delta}_k\}$.

Update $\Delta_k = \max\{\tilde{\Delta}_k, \beta \|\nabla m_k(x_k)\|\}$.

1.2. Obtain trust region step s_k satisfying a sufficient decrease condition (eg.- (3.7)).

1.3. Evaluate $f(x_k + s_k)$.

1.4. Adjust trust region according to ratio $\rho_k = \frac{f(x_k)-f(x_k+s_k)}{m_k(x_k)-m_k(x_k+s_k)}$:

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta_1 \text{ and } \Delta_k < \beta \|\nabla m_k(x_k)\| \\ \Delta_k & \text{if } \rho_k \geq \eta_1 \text{ and } \Delta_k \geq \beta \|\nabla m_k(x_k)\| \\ \Delta_k & \text{if } \rho_k < \eta_1 \text{ and } m_k \text{ is not fully linear} \\ \gamma_0\Delta_k & \text{if } \rho_k < \eta_1 \text{ and } m_k \text{ is fully linear,} \end{cases} \tag{3.5}$$

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k \geq \eta_1 \\ x_k + s_k & \text{if } \rho_k > \eta_0 \text{ and } m_k \text{ is fully linear} \\ x_k & \text{else.} \end{cases} \tag{3.6}$$

1.5. Improve m_k if $\rho_k < \eta_1$ and m_k is not fully linear.

1.6. Form new model m_{k+1}.

Reluctance to use nonpolynomial models in practice can be attributed to the difficulty of solving the subproblem (3.2). However, using the sufficient decrease guaranteed by the Lemma 3.1, we are still able to guarantee convergence to first-order critical points. This result is independent of the number of local or global minima that the subproblem may have as a result of using multimodal models.

Further, we assume that the twice continuously differentiable model used in practice will have first- and second-order derivatives available to solve (3.2). Using a more sophisticated solver may be especially attractive when this expense is negligible relative to evaluation of f at the subproblem solution.

We now state the convergence result for our models of interest and Algorithm 1.

Theorem 3.2. *Suppose that the following two assumptions hold:*

(AF) $f \in C^1[\Omega]$ *for some open* $\Omega \supset \mathcal{L}(x_0)$ *(with* $\mathcal{L}(x_0)$ *defined in* (3.3)*),* ∇f *is Lipschitz continuous on* $\mathcal{L}(x_0)$*, and* f *is bounded on* $\mathcal{L}(x_0)$*.*

(AM) *For all* $k \geq 0$ *we have* $m_k \in C^2[\mathcal{B}_k]$*,* $\infty > \kappa_H \geq \max_{x\in\mathcal{B}_k} \left\|\nabla^2 m_k(x)\right\|$*, and* m_k *can be made (and verified to be) fully linear by some finite procedure.*

Then for the sequence of iterates generated by Algorithm 1, we have

$$\lim_{k \to \infty} \nabla f(x_k) = 0. \tag{3.10}$$

Proof. This follows in large part from the lemmas in [8] with minor changes made to accommodate our sufficient decrease condition and the trust region norm employed. These lemmas, and further explanation where needed, are provided in [29]. □

4 Radial Basis Functions and ORBIT

Having outlined the fundamental conditions in Theorem 3.2 needed to show convergence of Algorithm 1, in this section we analyze which radial basis function models satisfy these conditions. We also show how the ORBIT algorithm fits in this globally convergent framework.

Throughout this section we drop the dependence of the model on the iteration number, k, but we intend for the model m and base point x_b to be the kth model and iterate, m_k and x_k, in the trust region algorithm of the previous section.

An alternative to polynomials is an interpolating surrogate that is a linear combination of nonlinear nonpolynomial basis functions. One such model is of the form

$$m(x_b + s) = \sum_{j=1}^{|\mathcal{Y}|} \lambda_j \phi(\|s - y_j\|) + P(s), \tag{4.1}$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a univariate function and $P \in \mathcal{P}^n_{d-1}$ is a polynomial as in Section 2. Such models are called *radial basis functions* (RBFs) because $m(x_b + s) - P(s)$ is a linear combination of shifts of a function that is constant on spheres in $\mathbb{R}^n$.

Interpolation by RBFs on scattered data has only recently gained popularity in practice [4]. In the context of optimization, RBF models have been used primarily for global optimization [3, 13, 26] because they are able to model multimodal/nonconvex functions and interpolate a large number of points in a numerically stable manner.

To our knowledge, Oeuvray was the first to employ RBFs in a local optimization algorithm. In his 2005 dissertation [22], he introduced BOOSTERS, a derivative-free trust region algorithm using a cubic RBF model with a linear tail. Oeuvray was motivated by medical image registration problems and was particularly interested in "doping" his algorithm with gradient information [23]. When the number of interpolation points is fixed from one iteration to the next, Oeuvray also showed that the RBF model parameters λ and ν can be updated in the same complexity as the underdetermined quadratics from [25] (interpolating the same number of points).

4.1 Conditionally Positive Definite Functions

We now define the fundamental property we rely on, using the notation of Wendland [28].

Definition 4.1. *Let π be a basis for $\mathcal{P}^n_{d-1}$, with the convention that $\pi = \emptyset$ if $d = 0$. A function ϕ is said to be conditionally positive definite of order d if for all sets of distinct points $\mathcal{Y} \subset \mathbb{R}^n$ and all $\lambda \neq 0$ satisfying $\sum_{j=1}^{|\mathcal{Y}|} \lambda_j \pi(y_j) = 0$, the quadratic form $\sum_{i,j=1}^{|\mathcal{Y}|} \lambda_i \lambda_j \phi(\|y_i - y_j\|)$ is positive.*

Table 4.1: Popular twice continuously differentiable RBFs and order of conditional positive definiteness

$\phi(r)$	Order	Parameters	Example
r^β	2	$\beta \in (2,4)$	Cubic, r^3
$(\gamma^2+r^2)^\beta$	2	$\gamma>0, \beta \in (1,2)$	Multiquadric I, $(\gamma^2+r^2)^{3/2}$
$-(\gamma^2+r^2)^\beta$	1	$\gamma>0, \beta \in (0,1)$	Multiquadric II, $-\sqrt{\gamma^2+r^2}$
$(\gamma^2+r^2)^{-\beta}$	0	$\gamma>0, \beta>0$	Inv. Multiquadric, $(\gamma^2+r^2)^{-1/2}$
$\exp\left(-r^2/\gamma^2\right)$	0	$\gamma>0$	Gaussian, $\exp\left(-r^2/\gamma^2\right)$

Table 4.1 lists examples of popular radial functions and their orders of conditional positive definiteness. Note that if a radial function ϕ is conditionally positive definite of order d, then it is also conditionally positive definite of order $\hat{d} \geq d$ [28, p. 98].

We now use the property of conditional positive definiteness to uniquely determine an RBF model that interpolates data on a set $\mathcal{Y}$. Let $\Phi_{i,j} = \phi(\|y_i - y_j\|)$ define the square matrix $\Phi \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, and let Π be the polynomial matrix $\Pi_{i,j} = \pi_i(y_j)$ as in Section 2 so that $P(s) = \sum_{i=1}^{\hat{p}} \nu_i \pi_i(s)$. Provided that $\mathcal{Y}$ is $\mathcal{P}_{d-1}^n$-unisolvent (as in Definition 2.1), we have the equivalent nonsingular symmetric linear system:

$$\begin{bmatrix} \Phi & \Pi^T \\ \Pi & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \nu \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{4.2}$$

The top set of equations corresponds to the interpolation conditions in (2.5) for the RBF model in (4.1), while the lower set ensures uniqueness of the solution.

As in Section 2 for polynomial models, for conditionally positive definite functions of order d, a sufficient condition for the nonsingularity of (4.2) is that the points in $\mathcal{Y}$ be distinct and yield a Π^T of full column rank. Clearly this condition is *geometric*, depending only on the location of (but not function values at) the data points.

The saddle point problem in (4.2) will generally be indefinite. However, we employ a null-space method that directly relies on the conditional positive definiteness of ϕ. If Π^T is full rank, then $R \in \mathbb{R}^{(n+1)\times(n+1)}$ is nonsingular from the truncated QR factorization $\Pi^T = QR$. By the lower set of equations in (4.2) we must have $\lambda = Z\omega$ for $\omega \in \mathbb{R}^{|\mathcal{Y}|-n-1}$ and any orthogonal basis Z for $\mathcal{N}(\Pi)$. Hence (4.2) reduces to

$$Z^T \Phi Z \omega = Z^T f \tag{4.3}$$

$$R\nu = Q^T(f - \Phi Z \omega). \tag{4.4}$$

Given that Π^T is full rank and the points in $\mathcal{Y}$ are distinct, Definition 4.1 directly implies that $Z^T\Phi Z$ is positive definite for any ϕ that is conditionally positive definite of at most order d. Positive definiteness of $Z^T\Phi Z$ guarantees the existence of a nonsingular lower triangular Cholesky factor L such that

$$Z^T \Phi Z = LL^T, \tag{4.5}$$

and the isometry of Z gives the bound

$$\|\lambda\| = \left\|ZL^{-T}L^{-1}Z^T f\right\| \leq \left\|L^{-1}\right\|^2 \|f\|. \tag{4.6}$$

4.2 Fully Linear RBF Models

Thus far we have maintained a very general RBF framework. In order for the convergence results in Section 3 to apply, we now focus on a more specific set of radial functions that satisfy two additional conditions:

- $\phi \in C^2[\mathbb{R}_+]$ and $\phi'(0) = 0$
- ϕ conditionally positive definite of order 2 or less.

The first condition ensures that the resulting RBF model is twice continuously differentiable. The second condition is useful for restricting ourselves to models of the form (4.1) with a linear tail $P \in \mathcal{P}_1^n$.

For RBF models that are twice continuously differentiable and have a linear tail,

$$\nabla m(x_b + s) = \sum_{\{y_i \in \mathcal{Y}: y_i \neq s\}} \lambda_i \phi'(\|s - y_i\|) \frac{s - y_i}{\|s - y_i\|} + \nabla P(s), \tag{4.7}$$

$$\nabla^2 m(x_b + s) = \sum_{y_i \in \mathcal{Y}} \lambda_i \Theta(\|s - y_i\|), \tag{4.8}$$

with

$$\Theta(r) = \begin{cases} \frac{\phi'(\|r\|)}{\|r\|} I_n + \left(\phi''(\|r\|) - \frac{\phi'(\|r\|)}{\|r\|}\right) \frac{r}{\|r\|} \frac{r^T}{\|r\|}, & \text{if } r \neq 0, \\ \phi''(0) I_n & \text{if } r = 0, \end{cases} \tag{4.9}$$

where we have explicitly defined these derivatives for the special case when s is one of the interpolation knots in $\mathcal{Y}$.

The following lemma is a consequence of an unproven statement in Oeuvray's dissertation [22], which we could not locate in the literature. It provides necessary and sufficient conditions on ϕ for the RBF model m to be twice continuously differentiable.

Lemma 4.2. *The model m defined in* (4.1) *is twice continuously differentiable on $\mathbb{R}^n$ if and only if $\phi \in C^2[\mathbb{R}_+]$ and $\phi'(0) = 0$.*

Proof. We begin by noting that the polynomial tail P and composition with the sum over $\mathcal{Y}$ are both smooth. Moreover, away from any of the points in $\mathcal{Y}$, m is clearly twice continuously differentiable if and only if $\phi \in C^2[\mathbb{R}_+]$. It now remains only to treat the case when $s = y_j \in \mathcal{Y}$.

If ϕ' is continuous but $\phi'(0) \neq 0$, then since $\frac{s-y_j}{\|s-y_j\|}$ is always of bounded magnitude but does not exist as $s \to y_j$, we have that ∇m in (4.7) is not continuous at y_j.

We conclude by noting that $\phi'(0) = 0$ is sufficient for the continuity of $\nabla^2 m$ at y_j. To see this, recall from L'Hôpital's rule in calculus that $\lim_{a \to 0} \frac{g(a)}{a} = g'(0)$, provided $g(0) = 0$ and g is differentiable at 0. Applying this result with $g = \phi'$, we have that

$$\lim_{s \to y_j} \frac{\phi'(\|s - y_j\|)}{\|s - y_j\|} = \phi''(0).$$

Hence the second term in the expression for Θ in (4.9) vanishes as $r \to 0$, leaving only the first; that is, $\lim_{r \to 0} \Theta(r) = \phi''(0) I_n$ exists. □

Table 4.2: Upper bounds on RBF components (assumes $\gamma > 0$, $r \in [0, \Delta]$, β as in Table 4.1)

$\phi(r)$	$\lvert\phi(r)\rvert$	$\left\lvert\frac{\phi'(r)}{r}\right\rvert$	$\lvert\phi''(r)\rvert$
r^β	Δ^β	$\beta\Delta^{\beta-2}$	$\beta(\beta-1)\Delta^{\beta-2}$
$(\gamma^2+r^2)^\beta$,	$(\gamma^2+\Delta^2)^\beta$	$2\beta(\gamma^2+\Delta^2)^{\beta-1}$	$2\beta(\gamma^2+\Delta^2)^{\beta-1}\left(1+\frac{2(\beta-1)\Delta^2}{\gamma^2+\Delta^2}\right)$
$-(\gamma^2+r^2)^\beta$,	$(\gamma^2+\Delta^2)^\beta$	$2\beta\gamma^{2(\beta-1)}$	$2\beta\gamma^{2(\beta-1)}$
$(\gamma^2+r^2)^{-\beta}$,	$\gamma^{-2\beta}$	$2\beta\gamma^{-2(\beta+1)}$	$2\beta\gamma^{-2(\beta+1)}$
$\exp(-r^2/\gamma^2)$	1	$2/\gamma^2$	$2/\gamma^2$

We note that this result implies that models using the thin-plate spline radial function $\phi(r) = r^2\log(r)$ are not twice continuously differentiable and hence do not fit in our framework.

Having established conditions for the twice differentiability of the radial portion of m in (4.1), we now focus on the linear tail P. Without loss of generality, we assume that the base point x_b is an interpolation point so that $y_1 = 0 \in \mathcal{Y}$. Employing the standard linear basis and permuting the points, we then have that the polynomial matrix $\Pi_{i,j} = \pi_i(y_j)$ is of the form

$$\Pi = \begin{bmatrix} Y & 0 & y_{n+2} & \cdots & y_{|\mathcal{Y}|} \\ e^T & 1 & 1 & \cdots & 1 \end{bmatrix}, \tag{4.10}$$

where e is the vector of ones and Y denotes a matrix of n particular nonzero points in $\mathcal{Y}$.

Recall that, in addition to the distinctness of the points in $\mathcal{Y}$, a condition for the nonsingularity of the RBF system (4.2) is that the first $n+1$ columns of Π in (4.10) are linearly independent. This is exactly the condition needed for the fully linear interpolation models in Section 2, where bounds for the matrix Y were provided.

In order to fit RBF models with linear tails into the globally convergent trust region framework of Section 3, it remains only to show that the model Hessians are bounded by some fixed constant κ_H.

From (4.8) and (4.9), it is clear that the magnitude of the Hessian depends only on the quantities λ, $\left|\frac{\phi'(r)}{r}\right|$, and $|\phi''(r)|$. As an example, Table 4.2 provides bounds on the last two quantities for the radial functions in Table 4.1 when r is restricted to lie in the interval $[0, \Delta]$. In particular, these bounds provide an upper bound for

$$h_\phi(\Delta) = \max\left\{2\left|\frac{\phi'(r)}{r}\right| + \left|\phi''(r)\right| : r \in [0, \Delta]\right\}. \tag{4.11}$$

From (4.6) we also have a bound on λ provided that the appropriate Cholesky factor L is of bounded norm. We bound $\left\|L^{-1}\right\|$ inductively by building up the interpolation set $\mathcal{Y}$ one point at a time. This inductive method lends itself well to a practical implementation and was inspired by the development in [3].

To start this inductive argument, we assume that $\mathcal{Y}$ consists of $n+1$ points that are $\mathcal{P}_1^n$-unisolvent. With only these $n+1$ points, $\lambda = 0$ is the unique solution to (4.2), and hence the RBF model is linear. To include an additional point $y \in \mathbb{R}^n$ in the interpolation set $\mathcal{Y}$ (beyond the initial $n+1$ points), we appeal to the following lemma (derived in [31]).

Algorithm 2 Algorithm for adding additional interpolation points.

2.0. Input $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\} \subset \mathbb{R}^n$, $\mathcal{Y}$ consisting of $n+1$ sufficiently affinely independent points, constants $\theta_2 > 0$, $\Delta > 0$, and $p_{\max} \geq n+1$.

2.1. Using $\mathcal{Y}$, compute the Cholesky factorization $LL^T = Z^T \Phi Z$ as in (4.5).

2.2. For all $y \in \mathcal{D}$ such that $\|y\|_k \leq \Delta$:

If $\tau(y) \geq \theta_2$,

$\mathcal{Y} \leftarrow \mathcal{Y} \cup \{y\}$,

Update $Z \leftarrow Z_y$, $L \leftarrow L_y$,

If $|\mathcal{Y}| = p_{\max}$, `return`.

Lemma 4.3. *Let $\mathcal{Y}$ be such that Π is full rank and $LL^T = Z^T \Phi Z$ is invertible as in (4.5). If $y \in \mathbb{R}^n$ is added to $\mathcal{Y}$, then the new Cholesky factor L_y has an inverse*

$$L_y^{-1} = \begin{bmatrix} L^{-1} & 0 \\ \frac{-v_y^T L^{-T} L^{-1}}{\tau(y)} & \frac{1}{\tau(y)} \end{bmatrix}, \qquad \text{with} \quad \tau(y) = \sqrt{\sigma_y - \|L^{-1} v_y\|^2}, \tag{4.12}$$

provided that the constant $\tau(y)$ is positive.

Here we see that only the last row of L_y^{-1} is affected by the addition of the new point y. As noted in [31], the constant σ_y and vector v_y in Lemma 4.3 appear in the reduced $Z_y^T \Phi_y Z_y = L_y L_y^T$ when y is added, and can be obtained by applying $n+1$ Givens rotations to Π_y^T. The following lemma bounds the resulting Cholesky factor L_y^{-1} as a function of the previous factor L^{-1}, v_y, and $\tau(y)$.

Lemma 4.4. *If $\|L^{-1}\| \leq \kappa$ and $\tau(y) \geq \theta > 0$, then*

$$\|L_y^{-1}\|^2 \leq \kappa + \frac{1}{\theta^2}\left(1 + \|v_y\| \kappa^2\right)^2. \tag{4.13}$$

Proof. Let $w_y = (w, \tilde{w}) \in R^{|\mathcal{Y}|+1}$ be an arbitrary vector with $\|w_y\| = 1$. Then

$$\begin{aligned} \|L_y^{-1} w_y\|^2 &= \|L^{-1} w\|^2 + \frac{1}{\tau(y)^2}\left(\tilde{w} - v_y^T L^{-T} L^{-1} w\right)^2 \\ &\leq \kappa + \frac{1}{\theta^2}\left(\tilde{w}^2 - 2\tilde{w} v_y^T L^{-T} L^{-1} w + \left(v_y^T L^{-T} L^{-1} w\right)^2\right) \\ &\leq \kappa + \frac{1}{\theta^2}\left(1 + 2\|L^{-1} v_y\| \|L^{-1} w\| + \left(\|L^{-1} v_y\| \|L^{-1} w\|\right)^2\right) \\ &\leq \kappa + \frac{1}{\theta^2}\left(1 + \|v_y\| \kappa^2\right)^2. \end{aligned}$$

□

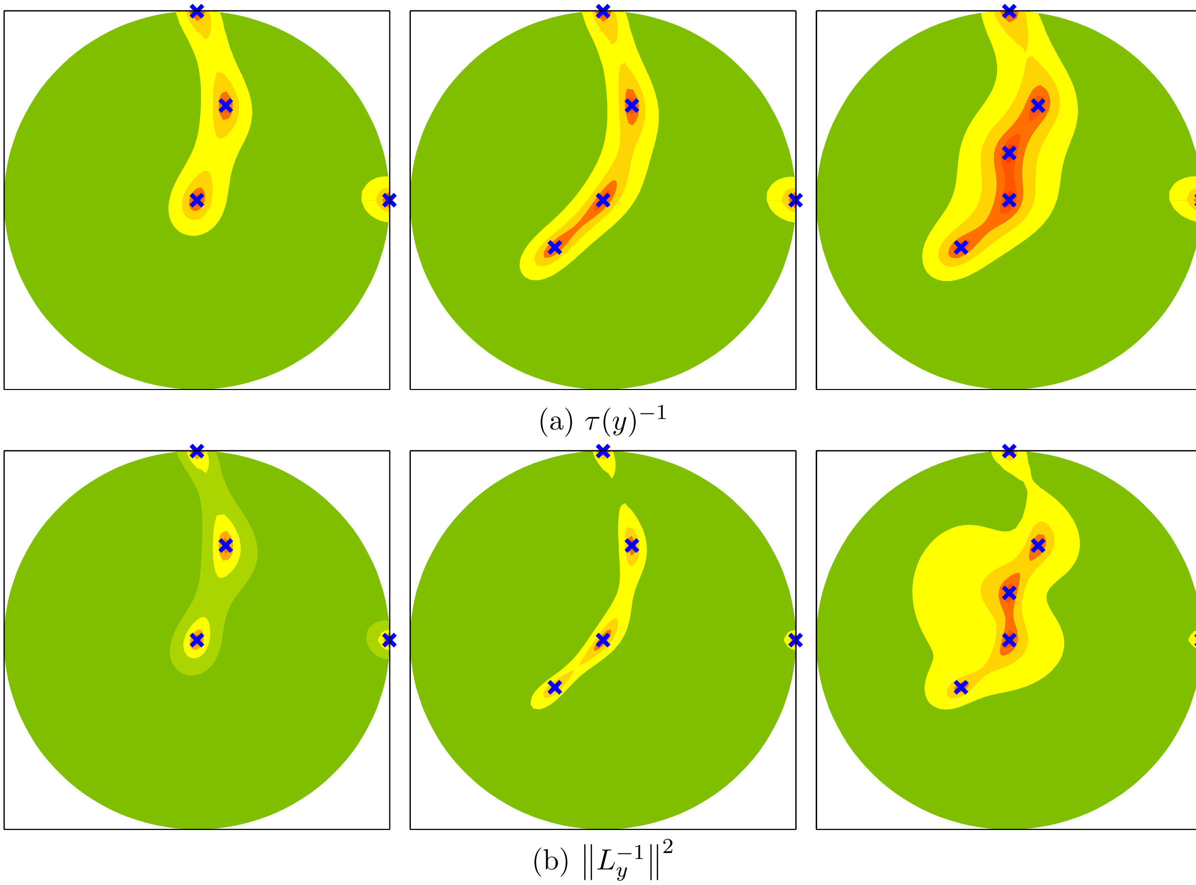

Figure 4.1: Contours for $\tau(y)^{-1}$ and $\left\|L_y^{-1}\right\|^2$ values (4.12) for a multiquadric RBF interpolating 4, 5, and 6 points in $\mathbb{R}^2$ (log-scale). The quantities grow as the interpolation points are approached.

Lemma 4.4 suggests the procedure given in Algorithm 2, which we use in ORBIT to iteratively add previously evaluated points to the interpolation set $\mathcal{Y}$. Before this algorithm is called, we assume that $\mathcal{Y}$ consists of $n+1$ sufficiently affinely independent points generated as described in Section 2 and hence the initial L matrix is empty.

Figure 4.1 (a) gives an example of $\tau(y)^{-1}$ values for different interpolation sets in $\mathbb{R}^2$. In particular we note that $\tau(y)$ approaches zero as y approaches any of the points already in the interpolation set $\mathcal{Y}$. Figure 4.1 (b) shows the behavior of $\left\|L_y^{-1}\right\|^2$ for the same interpolation sets and illustrates the relative correspondence between the values of $\tau(y)^{-1}$ and $\left\|L_y^{-1}\right\|^2$.

We now assume that both $\mathcal{Y}$ and the point y being added to the interpolation set belong to some bounded domain $\{x \in \mathbb{R}^n : \|x\|_k \leq \Delta\}$. Thus the quantities $\{\|x-z\| : x, z \in \mathcal{Y} \cup y\}$ are all of magnitude no more than $2c_1\Delta$, since $\|\cdot\| \leq c_1 \|\cdot\|_k$. The elements in $\Phi_{i,j} = \phi(\|y_i - y_j\|)$ and $\phi_y = [\phi(\|y-y_1\|), \ldots, \phi(\left\|y - y_{|\mathcal{Y}|}\right\|)]^T$ are bounded by $k_\phi(2c_1\Delta)$, where

$$k_\phi(2c_1\Delta) = \max\{|\phi(r)| : r \in [0, 2c_1\Delta]\}. \tag{4.14}$$

Bounds for the specific ϕ functions of the radial basis functions of interest are provided in Table 4.2. Using the isometry of Z_y we hence have the bound

$$\|v_y\| \leq \sqrt{|\mathcal{Y}|(|\mathcal{Y}|+1)}\, k_\phi(2c_1\Delta), \tag{4.15}$$

Algorithm 3 Algorithm for constructing model m_k.

3.0. Input $\mathcal{D} \subset \mathbb{R}^n$, constants $\theta_2 > 0$, $\theta_4 \geq \theta_3 \geq 1$, $\theta_1 \in (0, \frac{1}{\theta_3}]$, $\Delta_{\max} \geq \Delta_k > 0$, and $p_{\max} \geq n+1$.

3.1. Seek affinely independent interpolation set $\mathcal{Y}$ within distance $\theta_3 \Delta_k$.

Save z_1 as a model-improving direction for use in Step 1.5 of Algorithm 1.

If $|\mathcal{Y}| < n+1$ (and hence m_k is not fully linear):

Seek $n+1-|\mathcal{Y}|$ additional points in $\mathcal{Y}$ within distance $\theta_4 \Delta_{\max}$

If $|\mathcal{Y}| < n+1$, evaluate f at remaining $n+1-|\mathcal{Y}|$ model points so that $|\mathcal{Y}| = n+1$.

3.2. Use up to $p_{\max} - n - 1$ additional points within $\theta_4 \Delta_{\max}$ using Algorithm 2.

3.3. Obtain model parameters by (4.3) and (4.4).

---

independent of where in $\{x \in \mathbb{R}^n : \|x\|_k \leq \Delta\}$ the point y lies, which can be used in (4.13) to bound $\left\|L_y^{-1}\right\|$. The following theorem gives the resulting bound.

Theorem 4.5. *Let $\mathcal{B} = \{x \in \mathbb{R}^n : \|x - x_b\|_k \leq \Delta\}$. Let $\mathcal{Y} \subset \mathcal{B} - x_b$ be a set of distinct interpolation points, $n+1$ of which are affinely independent and $|f(x_b + y_i)| \leq f_{\max}$ for all $y_i \in \mathcal{Y}$. Then for a model of the form (4.1), with a bound h_ϕ as defined in (4.11), interpolating f on $x_b + \mathcal{Y}$, we have that for all $x \in \mathcal{B}$*

$$\left\|\nabla^2 m(x)\right\| \leq |\mathcal{Y}| \left\|L^{-1}\right\|^2 h_\phi(2c_1\Delta) f_{\max} =: \kappa_H. \tag{4.16}$$

Proof. Let $r_i = s - y_i$, and note that when s and $\mathcal{Y}$ both belong to $\mathcal{B} - x_b$, $\|r_i\| \leq c_1 \|r_i\|_k \leq 2c_1\Delta$ for $i = 1, \ldots, |\mathcal{Y}|$. Thus for an arbitrary w with $\|w\| = 1$,

$$\begin{aligned}
\left\|\nabla^2 m(x_b + s)w\right\| &\leq \sum_{i=1}^{|\mathcal{Y}|} |\lambda_i| \left\| \frac{\phi'(\|r_i\|)}{\|r_i\|} w + \left(\phi''(\|r_i\|) - \frac{\phi'(\|r_i\|)}{\|r_i\|}\right) \frac{r_i^T w}{\|r_i\|} \frac{r_i}{\|r_i\|} \right\|, \\
&\leq \sum_{i=1}^{|\mathcal{Y}|} |\lambda_i| \left[2 \left| \frac{\phi'(\|r_i\|)}{\|r_i\|} \right| + |\phi''(\|r_i\|)| \right] \\
&\leq \|\lambda\|_1 h(2c_1\Delta) \leq \sqrt{|\mathcal{Y}|} \left\|L^{-1}\right\|^2 \|f\| \, h(2c_1\Delta),
\end{aligned}$$

where the last two inequalities follow from (4.11) and (4.6), respectively. Noting that $\|f\| \leq \sqrt{|\mathcal{Y}|} f_{\max}$ gives the desired result. □

4.3 RBF Models in ORBIT

Having shown how RBFs fit into the globally convergent framework for fully linear models, we collect some final details of ORBIT, consisting of Algorithm 1 and the RBF model formation summarized in Algorithm 3.

Algorithm 3 requires that the interpolation points in $\mathcal{Y}$ lie within some constant factor of the largest trust region $\Delta_{\max}$. This region, $\mathcal{B}_{\max} = \{y \in \mathbb{R}^n : \|y\|_k \leq \theta_4 \Delta_{\max}\}$, is chosen to be larger than the current trust region so that the algorithm can make use of more points previously evaluated in the course of the optimization.

In Algorithm 3 we certify a model to be fully linear if $n+1$ points within $\{y \in \mathbb{R}^n : \|y\|_k \leq \theta_3 \Delta_k\}$ result in pivots larger than θ_1, where the constant θ_1 is chosen so as to be attainable by the model directions (scaled by Δ_k) discussed in Section 2.

If not enough points are found, the model will not be fully linear; thus, we must expand the search for affinely independent points within the larger region $\mathcal{B}_{\max}$. If still fewer than $n+1$ points are available, we must evaluate f along a set of the model-improving directions Z to ensure that $\mathcal{Y}$ is $\mathcal{P}_1^n$-unisolvent.

Additional available points within $\mathcal{B}_{\max}$ are added to the interpolation set $\mathcal{Y}$ provided that they keep $\tau(y) \geq \theta_2 > 0$, until a maximum of $p_{\max}$ points are in $\mathcal{Y}$.

Since we have assumed that f is bounded on $\mathcal{L}(x_0)$ and that $\mathcal{Y} \subset \mathcal{B}_{\max}$, the bound (4.16) holds for all models used by the algorithm, regardless of whether they are fully linear. Provided that the radial function ϕ is chosen to satisfy the requirements of Lemma 4.2, m will be twice continuously differentiable. Hence ∇m is Lipschitz continuous on $\mathcal{B}_{\max}$, and κ_H in (3.8) is one possible Lipschitz constant. When combined with the results of Section 2 showing that such interpolation models can be made fully linear in a finite procedure, Theorem 3.2 guarantees that $\lim_{k\to\infty} \nabla f(x_k) = 0$ for trust region algorithms using these RBFs, and ORBIT in particular.

5 Computational Experiments

We now present numerical results aimed at determining the effect of selecting different types of RBF models. We follow the benchmarking procedures in [21], with the derivative-free convergence test

$$f(x_0) - f(x) \geq (1-\tau)(f(x_0) - f_L), \tag{5.1}$$

where $\tau > 0$ is a tolerance, x_0 is the starting point, and f_L is the smallest value of f obtained by any tested solver within a fixed number, μ_f, of function evaluations. We note that in (5.1), a problem is "solved" when the achieved reduction from the initial value, $f(x_0) - f(x)$, is at least $1-\tau$ times the best possible reduction, $f(x_0) - f_L$.

For each solver $s \in \mathcal{S}$ and problem $p \in \mathcal{P}$, we define $t_{p,s}$ as the number of function evaluations required by s to satisfy the convergence test (5.1) on p, with the convention that $t_{p,s} = \infty$ if s does not satisfy the convergence test on p within μ_f evaluations.

If we assume that (i) the differences in times for solvers to determine a point for evaluation of $f(x)$ are negligible relative to the time to evaluate the function, and (ii) the function requires the same amount of time to evaluate at any point in its domain. then differences in the measure $t_{p,s}$ roughly correspond to differences in computing time. Assumption (i) is reasonable for the computationally expensive simulation-based problems motivating this work.

Given this measure, we define the *data profile* $d_s(\alpha)$ for solver $s \in \mathcal{S}$ as

$$d_s(\alpha) = \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \alpha \right\} \right|, \tag{5.2}$$

where n_p is the number of variables in problem $p \in \mathcal{P}$. We note that the data profile $d_s : \mathbb{R} \to [0,1]$ is a nondecreasing step function that is independent of the data profiles of the other solvers $\mathcal{S}\backslash\{s\}$, provided that f_L is fixed. By this definition, $d_s(\kappa)$ is the percentage of problems that can be solved within κ *simplex gradient estimates.*

5.1 Smooth Test Problems

We begin by considering the test set $\mathcal{P}_S$ of 53 smooth nonlinear least squares problems defined in [21]. Each unconstrained problem is defined by a starting point x_0 and a function $f(x) = \sum_{i=1}^{k} f_i(x)^2$, comprised of a set of smooth components. The functions vary in dimension from $n = 2$ to $n = 12$, with the 53 problems being roughly uniformly distributed across these dimensions. The maximum number of function evaluations is set to $\mu_f = 1300$ so that at least the equivalent of 100 simplex gradient estimates can be obtained on all the problems in $\mathcal{P}_S$. The initial trust region radius is set to $\Delta_0 = \max\{1, \|x_0\|_\infty\}$ for each problem.

The ORBIT implementation illustrated here relies on a 2-norm trust region with parameter values as in [31]: $\eta_0 = 0$, $\eta_1 = .2$, $\gamma_0 = \frac{1}{2}$, $\gamma_1 = 2$, $\Delta_{\max} = 10^3\Delta_0$, $\epsilon = 10^{-10}$, $\kappa_d = 10^{-4}$, $\alpha = .9$, $\mu = 2000$, $\beta = 1000$, $\theta_1 = 10^{-3}$, $\theta_2 = 10^{-7}$, $\theta_3 = 10$, $\theta_4 = \max(\sqrt{n}, 10)$. In addition to the backtracking line search detailed here, we use an augmented Lagrangian method to approximately solve the trust region subproblem.

The first solver set we consider is the set $\mathcal{S}_A$ consisting of four different radial basis function types for ORBIT:

Multiquadric : $\phi(r) = -\sqrt{1 + r^2}$, with $p_{\max} = 2n + 1$.

Cubic : $\phi(r) = r^3$, with $p_{\max} = 2n + 1$.

Gaussian : $\phi(r) = \exp(-r^2)$, with $p_{\max} = 2n + 1$.

Thin Plate : $\phi(r) = r^2 \log(r)$, with $p_{\max} = 2n + 1$.

The common theme among these models is that they interpolate at most $p_{\max} = 2n + 1$ points, chosen because this is the number of interpolation points recommended by Powell for the NEWUOA algorithm [25]. We tested other values of the parameter γ used by multiquadric and Gaussian RBFs but found that $\gamma = 1$ worked well for both.

In our testing, we examined accuracy levels of $\tau = 10^{-k}$ for several k. For the sake of brevity, in Figure 5.1 we present the data profiles for $k = 1$ and $k = 5$. Recall that $\tau = 0.1$ corresponds to a 90% reduction relative to the best possible reduction in $\mu_f = 1300$ function evaluations. As discussed in [21], data profiles are used to see which solver is likely to achieve a given reduction of the function within a specific computational budget. For example, given the equivalent of 15 simplex gradients ($15(n + 1)$ function evaluations), we see that the cubic, multiquadric, Gaussian, and thin plate spline variants respectively solve 38%, 30%, 27%, and 30% of problems to $\tau = 10^{-5}$ accuracy.

For the accuracy levels shown, the cubic variant is generally best (especially given small budgets), while the Gaussian and thin plate spline variants are generally worst. The differences are smaller than those seen in [21], where $\mathcal{S}$ consisted of three very different solvers.

The second solver set, $\mathcal{S}_B$, consists of the same four radial basis function types:

Multiquadric : $\phi(r) = -\sqrt{1 + r^2}$, with $p_{\max} = \frac{(n+1)(n+2)}{2}$.

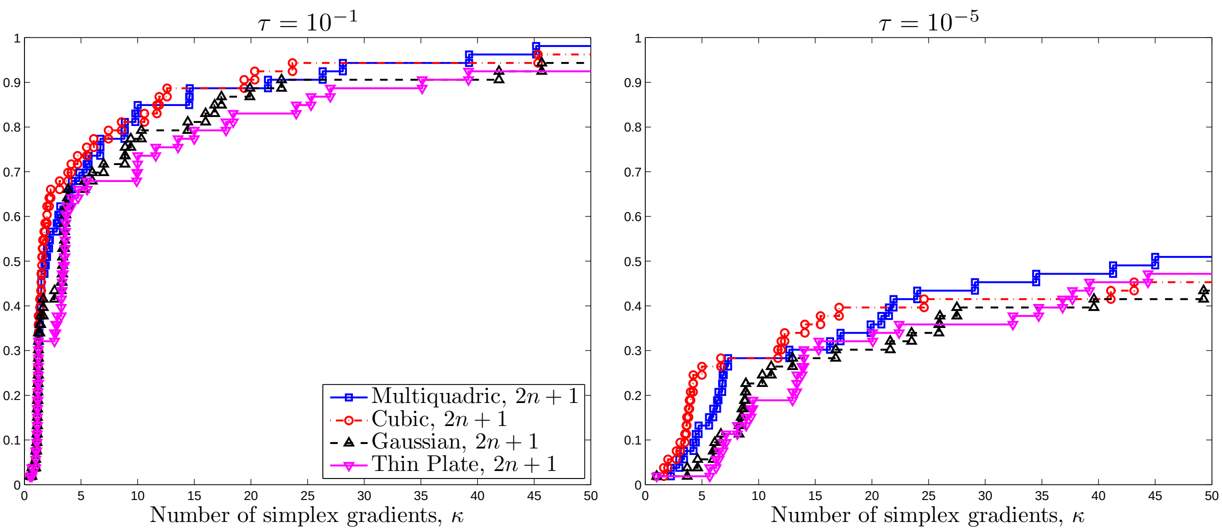

Figure 5.1: Data profiles $d_s(\kappa)$ for different RBF types with $p_{\max} = 2n+1$ on the smooth problems $\mathcal{P}_S$. These profiles show the percentage of problems solved as a function of a computational budget of simplex gradients ($\kappa(n+1)$ function evaluations).

Cubic : $\phi(r) = r^3$, with $p_{\max} = \frac{(n+1)(n+2)}{2}$.

Gaussian : $\phi(r) = \exp(-r^2)$, with $p_{\max} = \frac{(n+1)(n+2)}{2}$.

Thin Plate : $\phi(r) = r^2 \log(r)$, with $p_{\max} = \frac{(n+1)(n+2)}{2}$.

Here, the maximum number of points being interpolated corresponds to the number of points needed to uniquely fit an interpolating quadratic model, and this choice made solely to give an indication of how the performance changes with a larger number of interpolation points.

Figure 5.2 shows the data profiles for the accuracy levels $\tau \in \{10^{-1}, 10^{-5}\}$. The cubic variant is again generally best (especially given small budgets) but there are now larger differences among the variants. When the equivalent of 15 simplex gradients are available, we see that the cubic, multiquadric, Gaussian, and thin plate spline variants are respectively able to now solve 37%, 28%, 16%, 11% of problems to an accuracy level of $\tau = 10^{-5}$. We note that the raw data in Figure 5.2 should not be quantitatively compared against that in Figure 5.1 because the best function value found for each problem is obtained from only the solvers tested (in $\mathcal{S}_A$ or $\mathcal{S}_B$) and hence the convergence tests differ.

Our final test on these test problems compares the best variants for the two different maximum numbers of interpolation points. The solver set $\mathcal{S}_C$ consists of:

Cubic A : $\phi(r) = r^3$, with $p_{\max} = 2n+1$.

Cubic B : $\phi(r) = r^3$, with $p_{\max} = \frac{(n+1)(n+2)}{2}$.

Figure 5.3 shows that these two variants perform comparably, with differences smaller than those seen in Figures 5.1 and 5.2. As expected, as the number of function evaluations grows, the variant that is able to interpolate more points performs better. This variant also performs

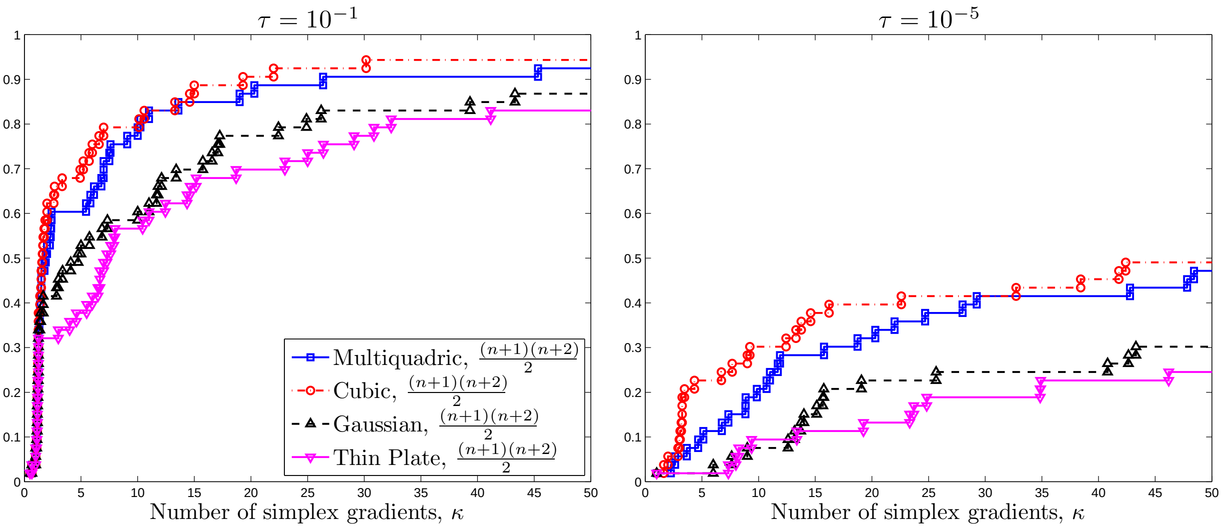

Figure 5.2: Data profiles $d_s(\kappa)$ for different RBF types with $p_{\max} = \frac{(n+1)(n+2)}{2}$ on the smooth problems $\mathcal{P}_S$. These profiles show the percentage of problems solved as a function of a computational budget of simplex gradients ($\kappa(n+1)$ function evaluations).

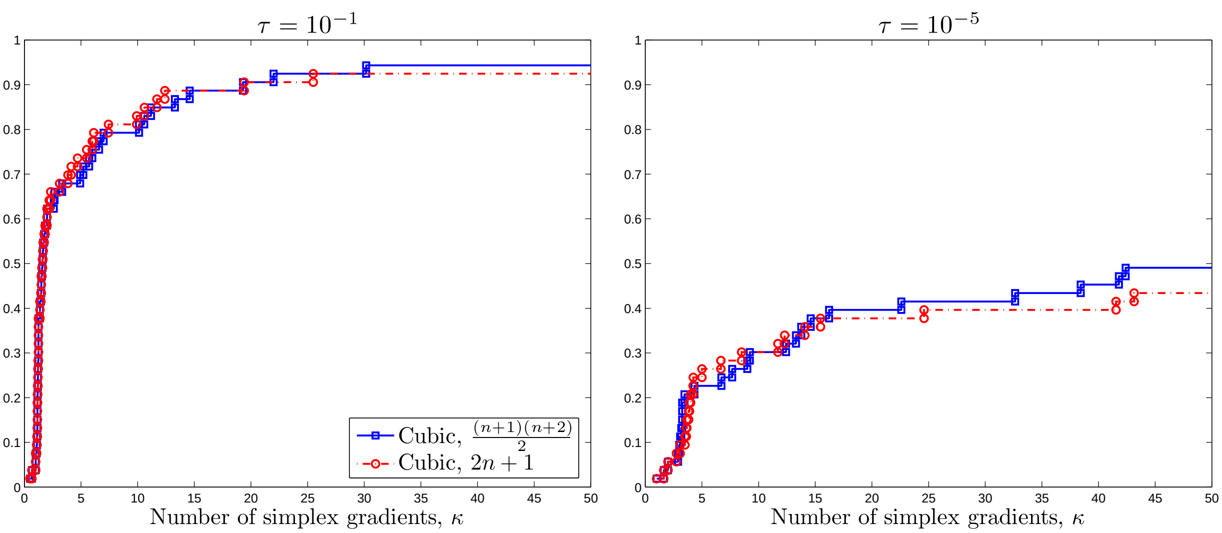

Figure 5.3: The effect of changing the maximum number of interpolation points, $p_{\max}$, on the data profiles $d_s(\kappa)$ for the smooth problems $\mathcal{P}_S$.

better when higher accuracy levels are demanded, and we attribute this to the fact that the model interpolating more points is generally a better approximation of the function f. The main downside of interpolating more points is that the linear systems in Section 4 will also grow, resulting in a higher linear algebra cost per iteration. As we will see in the next set of tests, for many applications, this cost may be viewed as negligible relative to the cost of evaluating the function f.

We are, however, surprised to see that the $2n+1$ variant performs better for some smaller budgets. For example, this variant performs slightly better between 5 and 15 simplex gradient estimates when $\tau = 10^{-1}$, and between 4 and 9 simplex gradient estimates when $\tau = 10^{-5}$. Since the initial $n+1$ evaluations are common to both variants and the parameter $p_{\max}$ has no effect on the subroutine determining the sufficiently affinely independent points, we might expect that the variant interpolating more points would do at least as well as the variant interpolating fewer points.

Further results comparing ORBIT (in 2-norm and ∞-norm trust regions) against NEWUOA on a set of noisy test problems are provided in [31].

5.2 An Environmental Application

We now illustrate the use of RBF models on a computationally expensive application problem.

The Blaine Naval Ammunition Depot comprises 48,800 acres just east of Hastings, Nebraska. In the course of producing nearly half of the naval ammunition used in World War II, much toxic waste was generated and disposed of on the site. Among other contaminants, both trichloroethylene (TCE), a probable carcinogen, and trinitrotoluene (TNT), a possible carcinogen, are present in the groundwater.

As part of a collaboration [2, 32] among environmental consultants, academic institutions, and governmental agencies, several optimization problems were formulated. Here we focus on one of the simpler formulations, where we have control over 15 injection and extraction wells located at fixed positions in the site. At each of these wells we can either inject clean water or extract contaminated water, which is then treated. Each instance of the decision variables hence corresponds to a pumping strategy that will run over a 30-year time horizon. For scaling purposes, each variable is scaled so that range of realistic pumping rates maps to the interval $[0,1]$.

The objective is to minimize the cost of the pumping strategy (the electricity needed to run the pumps) plus a penalty associated with exceeding the constraints on maximum allowable concentration of TCE and TNT over the 30-year planning horizon. For each pumping strategy, these concentrations are obtained by running a pair of coupled simulators, MODFLOW 2000 [27] and MT3D [33], which simulate the underlying contaminant transport and transformation. For a given set of pumping rates, this process required more than 45 minutes on a Pentium 4 dual-core desktop.

In the spirit of [21], in addition to ORBIT we considered three solvers designed to solve unconstrained serial optimization problems using only function values.

NMSMAX is an implementation of the Nelder-Mead method and is due to Higham [14]. We specified that the initial simplex have sides of length Δ_0. Since NMSMAX is defined for maximization problems, it was given $-f$.

SID-PSM is a pattern search solver due to Custódio and Vicente [11]. It is especially designed to make use of previous function evaluations. We used version 0.4 with an initial step size set to Δ_0. We note that the performance of the tested version has since been improved with the incorporation of interpolating models (as reported in [10]), but we have reported the originally tested version as an example of an industrial strength pattern search method not incorporating such models.

NEWUOA is a trust region solver using a quadratic model and is due to Powell [25]. The number of interpolation points was fixed at the recommended value of $p_{\max}=2n+1$, and the initial trust region radius was set to Δ_0.

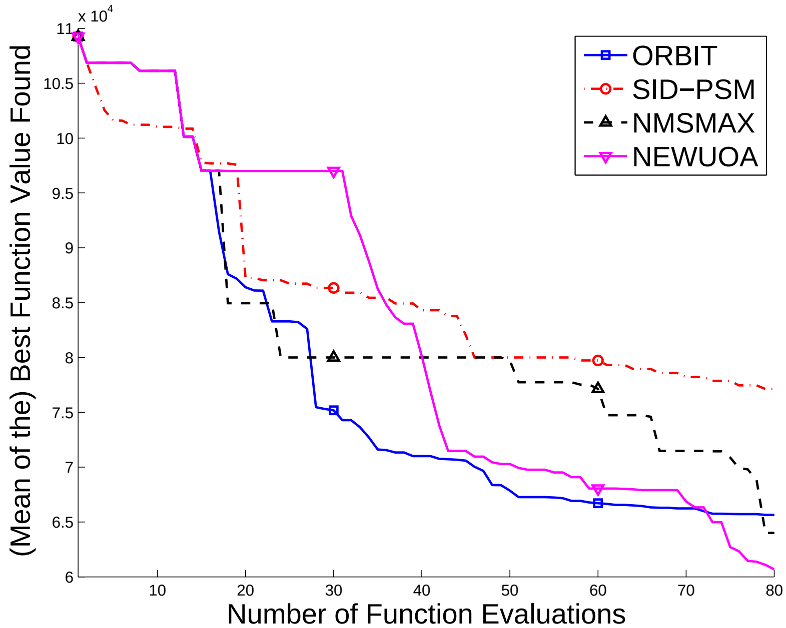

Figure 5.4: Mean (in 8 trials) of the best function value found for the first 80 evaluations on the Blaine problem. All ORBIT runs found a local minimum within 80 evaluations, while NEWUOA obtained a lower function value after 72 evaluations.

ORBIT used the same parameter values as used on the test functions, with a cubic RBF, initial trust region radius Δ_0, and a maximum number of interpolation points taken to be larger than the number of function evaluations, $p_{\max} \geq \mu_f$.

Each of these solvers also requires a starting point x_0 and a maximum number of allowable function evaluations, μ_f. A common selection of $\Delta_0 = 0.1$ was made to standardize the initial evaluations across the collection of solvers. Hence each solver except SID-PSM evaluated the same initial $n+1$ points. SID-PSM moves off this initial pattern once it sees a reduction. All other inputs were set to their default values except that we effectively set all termination parameters to zero to ensure that the solvers terminate only after exhausting the budget μ_f function evaluations.

We set $\mu_f = 5(n+1) = 80$, and since each evaluation (i.e., an environmental model simulation) requires more than 45 minutes, a single run of one solver thus requires nearly 3 CPU-days. As this problem is noisy and has multiple local minima, we chose to run each solver from the same eight starting points generated uniformly at random within the hypercube $[0,1]^{15}$ of interest. Thus, running four solvers over these eight starting points required roughly 3 CPU-months to obtain.

Figure 5.4 shows the average of the best function value obtained over the course of the first 80 function evaluations. By design, all solvers start from the same function value. The ORBIT solver does best initially, obtaining a function value of 70,000 in 46 evaluations. The ORBIT trajectory quickly flattens out as it is the first solver to find a local minima, with an average value of 65,600. In this case, however, the local minimum found most quickly by ORBIT has (on average) a higher function value than the point (not yet a local minimum) found by the NEWUOA and NMSMAX solvers after $\mu_f = 80$ evaluations. Hence, in these tests, NEWUOA and NMSMAX are especially good at finding a good minimum for a noisy function. On average, given $\mu_f = 80$ evaluations, NEWUOA finds a point with $f \approx 60,700$. None of these algorithms are designed to be global optimization solvers, so the comparison should focus more on the time to find the first local minimum.

The Blaine problem highlights the fact that solvers will have different performance on different functions and that many application problems contain computational noise and multiple distinct local minima, which can prevent globally convergent local methods from finding good solutions. Comparisons between ORBIT and other derivative-free algorithms on two different problems from environmental engineering can be found in [31]. The results in [31] found that two variants of ORBIT outperformed the three other solvers tested on these two environmental problems.

6 Conclusions and Perspectives

In this paper we have introduced and analyzed first-order derivative-free trust region algorithms based on radial basis functions, which are globally convergent. We first showed that, provided a function and a model are sufficiently smooth, interpolation on a set of sufficiently affinely independent points is enough to guarantee Taylor-like error bounds for both the model and its gradient. In Section 3 we extended the recent derivative-free trust region framework in [8] to include nonlinear fully linear models. In Section 4 we showed how RBFs can fit in this framework, and we introduced procedures for bounding an RBF model's Hessian. In particular, these results show that the ORBIT algorithm introduced in [31] converges to first-order critical points.

The central element of an RBF is the radial function. We have illustrated the results with a few different types of radial functions. However, the results presented here are wide-reaching, requiring only the following conditions on ϕ:

1. ϕ is twice continuously differentiable on $[0, u)$, for some $u > 0$,

2. $\phi'(0) = 0$, and

3. ϕ is conditionally positive definite of order 2.

While the last condition seems to be the most restrictive, only the first condition eliminates the thin-plate spline, popular in other applications of RBFs, from our analysis. Indeed, the numerical results show that the thin plate spline performed worst among the tested variants. We anticipate that this very general framework will be useful to researchers developing new optimization algorithms based on RBFs. Indeed, this theory extends to both the BOOSTERS algorithm [23] and ORBIT algorithm [31].

Our numerical results are aimed at illustrating the effect of using different types of radial functions ϕ in the ORBIT algorithm [31]. We saw that the cubic radial function slightly outperformed the multiquadric radial function, while the Gaussian radial function performed worse. These results are interesting because Gaussian radial basis functions are the only ones among those tested that are conditionally positive definite of order 0, requiring neither a linear nor a constant term to uniquely interpolated scattered data. Gaussian RBFs are usually used in kriging [9], which forms the basis for the global optimization methods such as [16]. We also found that the performance differences are greater when the RBF type is changed than when the maximum number of interpolation points is varied.

We also ran ORBIT on a computationally expensive environmental engineering problem, requiring 3 CPU-days for a single run of 80 evaluations. On this problem ORBIT quickly found a local minimum and obtained a good solution within 50 expensive evaluations.

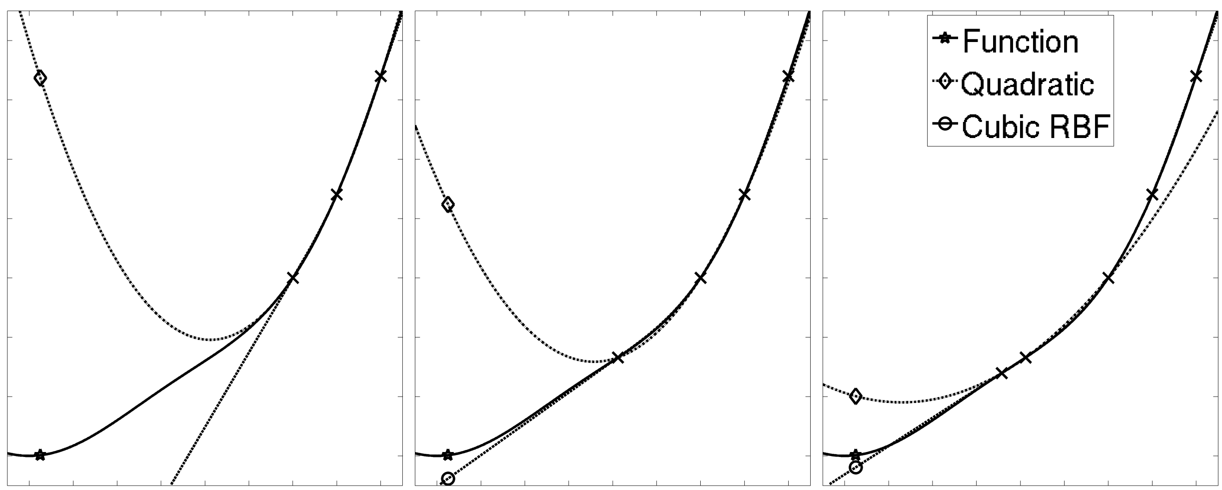

Figure 6.1: The function $f(x) = x\sin(x\pi/4) + x^2$ approximated by a quadratic interpolating $\frac{(n+1)(n+2)}{2} = 3$ points and a cubic RBF interpolating (from left to right) 3, 4, and 5 points.

Not surprisingly, there is no "free lunch:" while a method using RBFs outperformed methods using quadratics on the two application problems in [31], a quadratic method found the best solution on the application considered here when given a large enough budget of evaluations. Determining when to use a quadratic and when to use an RBF remains an open research problem. Our experience suggests that RBFs can be especially useful when f is nonconvex and has nontrivial higher-order derivatives.

An example of how this difference is amplified as more interpolation points are allowed is shown in Figure 6.1. As the number of points interpolated grow, the RBF model exhibits better extrapolation than the quadratic with a fixed number of points. Similar behavior is seen even when the additional points are incorporated using a regression quadratic or a higher-order polynomial.

The present work focused primarily on the theoretical implications needed to ensure that methods using radial basis function models fit in a globally convergent trust region framework. The results on the Blaine problem and the behavior seen in Figure 6.1 have motivated our development of global optimization methods in [29], and we intend to pursue "large-step" variants of ORBIT designed to step over computational noise.

We note that the theory presented here can be extended to models of other forms. We mention quadratics in [30], but we could also have used higher-order polynomial tails for better approximation bounds. For example, methods using a suitably conditioned quadratic tail could be expected to converge to second-order local minima. In fact, we attribute the quadratic-like convergence behavior RBF methods exhibit when at least $\frac{(n+1)(n+2)}{2}$ points are interpolated to the fact that the RBF models are *fully quadratic* with probability one, albeit with theoretically large Taylor constants. We leave the extensive numerical testing needed when many points are interpolated as future work.

Acknowledgments

We are grateful to Amandeep Singh for providing the application problem code, and to Rommel Regis and two anonymous referees for comments on an earlier draft.

References

[1] C. Audet and J.E. Dennis, Jr., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.

[2] D. Becker, B. Minsker, R. Greenwald, Y. Zhang, K. Harre, K. Yager, C. Zheng, and R. Peralta, *Reducing long-term remedial costs by transport modeling optimization*, Ground Water, 44 (2006), pp. 864–875.

[3] M. Björkman and K. Holmström, *Global optimization of costly nonconvex functions using radial basis functions*, Optim. Eng., 1 (2000), pp. 373–397.

[4] M.D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge, England, 2003.

[5] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust-region methods*, SIAM, Philadelphia, PA, 2000.

[6] A.R. Conn, K. Scheinberg, and P.L. Toint, *Recent progress in unconstrained nonlinear optimization without derivatives*, Math. Prog., 79 (1997), pp. 397–414.

[7] A.R. Conn, K. Scheinberg, and L.N. Vicente, *Geometry of interpolation sets in derivative free optimization*, Math. Prog., 111 (2008), pp. 141–172.

[8] ——, *Global convergence of general derivative-free trust-region algorithms to first and second order critical points*, SIAM J. Optim., 20 (2009), pp. 387–415.

[9] N.A. Cressie, *Statistics for spatial data*, Wiley, New York, 1993.

[10] A.L. Custódio, H. Rocha, and L.N. Vicente, *Incorporating minimum frobenius norm models in direct search*, Comput. Optim. Appl., 46 (2009), pp. 265–278.

[11] A.L. Custódio and L.N. Vicente, *Using sampling and simplex derivatives in pattern search methods*, SIAM J. Optim., 18 (2007), pp. 537–555.

[12] G. Fasano, J.L. Morales, and J. Nocedal, *On the geometry phase in model-based algorithms for derivative-free optimization*, Optim. Methods Softw., 24 (2009), pp. 145–154.

[13] H.-M. Gutmann, *A radial basis function method for global optimization*, J. Global Optim., 19 (2001), pp. 201–227.

[14] N.J. Higham, *The Matrix Computation Toolbox*. `www.ma.man.ac.uk/~higham/mctoolbox`.

[15] P.D. Hough, T.G. Kolda, and V.J. Torczon, *Asynchronous parallel pattern search for nonlinear optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 134–156.

[16] D.R. Jones, M. Schonlau, and W.J. Welch, *Efficient global optimization of expensive black-box functions*, J. Global Optim., 13 (1998), pp. 455–492.

[17] T.G. Kolda, *Revisiting asynchronous parallel pattern search for nonlinear optimization*, SIAM J. Optim., 16 (2005), pp. 563–586.

[18] T.G. Kolda, R.M. Lewis, and V.J. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[19] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright, *Convergence properties of the Nelder-Mead simplex algorithm in low dimensions*, SIAM J. Optim., 9 (1998), pp. 112–147.

[20] M. Marazzi and J. Nocedal, *Wedge trust region methods for derivative free optimization*, Math. Prog., 91 (2002), pp. 289–305.

[21] J.J. Moré and S.M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 20 (2009), pp. 172–191.

[22] R. Oeuvray, *Trust-Region Methods Based on Radial Basis Functions with Application to Biomedical Imaging*, PhD thesis, EPFL, Lausanne, Switzerland, 2005.

[23] R. Oeuvray and M. Bierlaire, *Boosters: A derivative-free algorithm based on radial basis functions*, Int. J. Model. Simul., 29 (2009), pp. 26–36.

[24] M.J.D. Powell, *UOBYQA: unconstrained optimization by quadratic approximation*, Math. Prog., 92 (2002), pp. 555–582.

[25] ——, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-Scale Nonlinear Optimization, G. Di Pillo and M. Roma, eds., Springer, 2006, pp. 255–297.

[26] R.G. Regis and C.A. Shoemaker, *A stochastic radial basis function method for the global optimization of expensive functions*, INFORMS J. Comput., 19 (2007), pp. 457–509.

[27] U.S. Geological Survey, *MODFLOW 2000*, 2000. See `http://water.usgs.gov/nrp/gwsoftware/modflow2000/modflow2000.html`.

[28] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, England, 2005.

[29] S.M. Wild, *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*, PhD thesis, Cornell University, Ithaca, NY, USA, 2008.

[30] ——, *MNH: A derivative-free optimization algorithm using minimal norm hessians*, in Tenth Copper Mountain Conference On Iterative Methods, April 2008. Available at `http://grandmaster.colorado.edu/~copper/2008/SCWinners/Wild.pdf`.

[31] S.M. Wild, R.G. Regis, and C.A. Shoemaker, *ORBIT: Optimization by radial basis function interpolation in trust-regions*, SIAM J. Sci. Comput., 30 (2008), pp. 3197–3219.

[32] Y. Zhang, R. Greenwald, B. Minsker, R. Peralta, C. Zheng, K. Harre, D. Becker, L. Yeh, and K. Yager, *Final cost and performance report application of flow and transport optimization codes to groundwater pump and treat systems*, Tech. Report TR-2238-ENV, Naval Facilities Engineering Service Center, Port Hueneme, California, January 2004.

[33] C. Zheng and P.P. Wang, *MT3DMS, a modular three-dimensional multi-species transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems – documentation and users guide*, Tech. Report Contract Report SERDP-99-1, U.S. Army Engineer Research and Development Center, 1999.