Computation of Sparse Low Degree Interpolating Polynomials and their Application to Derivative-Free Optimization

A. S. Bandeira* K. Scheinberg† L. N. Vicente‡

May 4, 2012

Abstract

Interpolation-based trust-region methods are an important class of algorithms for Derivative-Free Optimization which rely on locally approximating an objective function by quadratic polynomial interpolation models, frequently built from less points than there are basis components.

Often, in practical applications, the contribution of the problem variables to the objective function is such that many pairwise correlations between variables are negligible, implying, in the smooth case, a sparse structure in the Hessian matrix. To be able to exploit Hessian sparsity, existing optimization approaches require the knowledge of the sparsity structure. The goal of this paper is to develop and analyze a method where the sparse models are constructed automatically.

The sparse recovery theory developed recently in the field of compressed sensing characterizes conditions under which a sparse vector can be accurately recovered from few random measurements. Such a recovery is achieved by minimizing the ℓ_1-norm of a vector subject to the measurements constraints. We suggest an approach for building sparse quadratic polynomial interpolation models by minimizing the ℓ_1-norm of the entries of the model Hessian subject to the interpolation conditions. We show that this procedure recovers accurate models when the function Hessian is sparse, using relatively few randomly selected sample points.

Motivated by this result, we developed a practical interpolation-based trust-region method using deterministic sample sets and minimum ℓ_1-norm quadratic models. Our computational results show that the new approach exhibits a promising numerical performance both in the general case and in the sparse one.

Keywords: Derivative-free optimization, interpolation-based trust-region methods, random sampling, sparse recovery, compressed sensing, ℓ_1-minimization.

*Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA (ajsb@math.princeton.edu).

†Department of Industrial and Systems Engineering, Lehigh University, Harold S. Mohler Laboratory, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA (katyas@lehigh.edu). The work of this author is partially supported by AFOSR under grant FA9550-11-1-0239.

‡CMUC, Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal (lnv@mat.uc.pt). Support for this author was provided by FCT under grant PTDC/MAT/098214/2008.

1 Introduction

The wide range of applications of mathematical optimization have been recently enriched by the developments in emerging areas such as Machine Learning and Compressed Sensing, where a structure of a model needs to be recovered from some observations. Specially designed optimization methods have been developed to handle the new applications that often give rise to large scale, but convex and well structured problems. However, in many real-world applications, the objective function is calculated by some costly black-box simulation which does not provide information about its derivatives. Although one could estimate the derivatives, e.g., by finite differences, such a process is often too expensive and can produce misleading results in the presence of noise. An alternative is to consider methods that do not require derivative information, and such methods are the subject of study in Derivative-Free Optimization (DFO). In this paper we propose a reverse relationship between optimization and compressed sensing — instead of using optimization methods to solve compressed sensing problems we use the results of compressed sensing to improve optimization methods by recovering and exploiting possible structures of the black-box objective functions.

An important class of methods in DFO are interpolation-based trust-region methods. At each iteration, these methods build a model of the objective function that locally approximates it in some *trust region* centered at the current iterate. The model is then minimized in the trust region, and the corresponding minimizer is, hopefully, a better candidate for being a minimizer of the objective function in the trust region, and thus, possibly, is taken as the next iterate. It is usually preferable that minimization of the model in the trust region is an easy task, hence the models should be simple. The simplest yet meaningful class of models is the class of linear functions. Their drawback is that they do not capture the curvature of the objective function and thus slow down the convergence of the methods. A natural and convenient non-linear class of models, which is often efficiently used, is the quadratic class. Determined quadratic interpolation requires sample sets whose cardinality is approximately equal to the square of the dimension, which may turn out to be too costly if the objective function is expensive to evaluate. An alternative is to consider underdetermined quadratic models, using sample sets of smaller size than the ones needed for determined interpolation. However, in this case, the quality of the model may deteriorate.

In many applications, the objective function has structure, such as sparsity of the Hessian, which one may exploit to improve the efficiency of an optimization method. In DFO, since derivatives are not known, typically neither is their sparsity structure. If the structure is known in advance, such as in group partial separability, it can be exploited as it is proposed in [7]. The main idea of our work is to implicitly and automatically take advantage of the sparsity of the Hessian in the cases when the sparsity structure is not known in advance, to build accurate models from relatively small sample sets. This goal is achieved by minimizing the ℓ_1-norm of the Hessian model coefficients.

Our work relies on the sparse solution recovery theory developed recently in the field of compressed sensing, where one characterizes conditions under which a sparse signal can be accurately recovered from few random measurements. Such type of recovery is achieved by minimizing the ℓ_1-norm of the unknown signal subject to measurement constraints and can be accomplished in polynomial time.

The contribution of this paper is twofold. First, we show that it is possible to compute fully quadratic models (i.e., models with the same accuracy as second order Taylor models)

for functions defined on $\mathbb{R}^n$ with sparse Hessians based on randomly selected sample sets with only $\mathcal{O}(n(\log n)^4)$ sample points (instead of the $\mathcal{O}(n^2)$ required for the determined quadratic case) when the number of non-zero elements in the Hessian of the function is $\mathcal{O}(n)$. Second, we introduce a practical interpolation-based trust-region DFO algorithm exhibiting competitive numerical performance.

The state-of-the-art approach is to build quadratic interpolation models, based on sample sets of any size between $n+1$ and $(n+1)(n+2)/2$, taking up the available degrees of freedom by choosing the models with the smallest Frobenius norm of the Hessian [11] or Hessian change [22], and this approach has been shown to be robust and efficient in practice (see also the recent paper [17] where the models are always determined, varying thus the number of basis components). In the approach proposed in this paper, the degrees of freedom are taken up by minimizing the ℓ_1-norm of the Hessian of the model. We have tested the practical DFO algorithm using both minimum Frobenius and minimum ℓ_1-norm models. Our results demonstrate the ability of the ℓ_1-approach to improve the results of the Frobenius one in the presence of some form of sparsity in the Hessian of the objective function.

This paper is organized as follows. In Section 2, we introduce background material on interpolation models. We give a brief introduction to compressed sensing in Section 3, introducing also concepts related to *partially* sparse recovery (the details are left to a separate paper [3]). In Section 4, we obtain the main result mentioned above for sparse recovery of models for functions with sparse Hessians, using an orthogonal basis for the space of polynomials of degree ≤ 2. The proof of this result is based on sparse bounded orthogonal expansions which are briefly described in the beginning of Section 4. In Section 5, we introduce our practical interpolation-based trust-region method and present numerical results for the two underdetermined quadratic model variants, defined by minimum Frobenius and ℓ_1-norm minimization. Finally, in Section 6 we draw some conclusions and discuss possible avenues for future research.

The paper makes extensive use of vector, matrix, and functional norms. We will use ℓ_p or $\|\cdot\|_p$ for vector and matrix norms, without ambiguity. The notation $B_p(x;\Delta)$ will represent a closed ball in $\mathbb{R}^n$, centered at x and of radius Δ, in the ℓ_p-norm, i.e., $B_p(x;\Delta) = \{y \in \mathbb{R}^n : \|y-x\|_p \leq \Delta\}$. For norms of functions on normed spaces L, we will use $\|\cdot\|_L$.

2 Use of models in DFO trust-region methods

2.1 Fully linear and fully quadratic models

One of the main techniques used in DFO consists of locally modeling the objective function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ by models that are "simple" enough to be optimized easily and sufficiently "complex" to approximate f well. If a reliable estimate of the derivatives of the function is available, then one typically uses Taylor approximations of first and second order as polynomial models of $f(x)$. In DFO one has no access to derivatives or their accurate estimates, and hence other model classes are considered. However, the essential approximation quality of the Taylor models is required to be sustained when necessary by the models used in convergent DFO frameworks. For instance, the simplest first order approximation is provided by, the so-called, fully linear models, whose definition requires f to be smooth up to the first order.

Assumption 2.1 *Assume that f is continuously differentiable with Lipschitz continuous gradient (on an open set containing D). (For simplicity we will assume that all the balls and*

neighborhoods considered in this paper are contained in D.)

The following definition is essentially the same as given in [11, Definition 6.1] stated using balls in an arbitrary ℓ_p-norm, with $p \in (0, +\infty]$.

Definition 2.1 *Let a function $f : D \to \mathbb{R}$ satisfying Assumption 2.1 be given. A set of model functions $\mathcal{M} = \{m : \mathbb{R}^n \to \mathbb{R},\, m \in C^1\}$ is called a fully linear class of models if the following hold:*

1. *There exist positive constants κ_{ef}, κ_{eg}, and ν_1^m, such that for any $x_0 \in D$ and $\Delta \in (0, \Delta_{max}]$ there exists a model function m in $\mathcal{M}$, with Lipschitz continuous gradient and corresponding Lipschitz constant bounded by ν_1^m, and such that*

   - *the error between the gradient of the model and the gradient of the function satisfies*

   $$\|\nabla f(x) - \nabla m(x)\|_2 \;\leq\; \kappa_{eg}\,\Delta, \quad \forall x \in B_p(x_0; \Delta),$$

   - *and the error between the model and the function satisfies*

   $$|f(x) - m(x)| \;\leq\; \kappa_{ef}\,\Delta^2, \quad \forall x \in B_p(x_0; \Delta).$$

   Such a model m is called fully linear on $B_p(x_0; \Delta)$.

2. *For this class $\mathcal{M}$ there exists an algorithm, which we will call a 'model-improvement' algorithm, that in a finite, uniformly bounded (with respect to x_0 and Δ) number of steps can*

   - *either provide a certificate for a given model $m \in \mathcal{M}$ that it is fully linear on $B_p(x_0; \Delta)$,*
   - *or fail to provide such a certificate and find a model $\tilde{m} \in \mathcal{M}$ fully linear on $B_p(x_0; \Delta)$.*

It is important to note that this definition does not restrict fully linear models to linear functions, but instead considers models that approximate f *as well* as the linear Taylor approximations. Linear models such as linear interpolation (and first order Taylor approximation) do not capture the curvature information of the function that they are approximating. To achieve better practical local convergence rates in general it is essential to consider nonlinear models. In this paper we focus on quadratic interpolation models, which ultimately aim at a higher degree of approximation accuracy. We call such approximation models fully quadratic, following [11], and note that, as in the linear case, one can consider a wider class of models not necessarily quadratic. We now require the function f to exhibit smoothness up to the second order.

Assumption 2.2 *Assume that f is twice differentiable with Lipschitz continuous Hessian (on an open set containing D).*

Below we state the definition of fully quadratic models given in [11, Definition 6.2], again using balls in an ℓ_p-norm, with arbitrary $p \in (0, +\infty]$.

Definition 2.2 *Let a function $f : D \to \mathbb{R}$ satisfying Assumption 2.2 be given. A set of model functions $\mathcal{M} = \{m : \mathbb{R}^n \to \mathbb{R},\, m \in C^2\}$ is called a fully quadratic class of models if the following hold:*

1. *There exist positive constants κ_{ef}, κ_{eg}, κ_{eh}, and ν_2^m, such that for any $x_0 \in D$ and $\Delta \in (0, \Delta_{max}]$ there exists a model function m in $\mathcal{M}$, with Lipschitz continuous Hessian and corresponding Lipschitz constant bounded by ν_2^m, and such that*

   - *the error between the Hessian of the model and the Hessian of the function satisfies*

$$\|\nabla^2 f(x) - \nabla^2 m(x)\|_2 \;\leq\; \kappa_{eh}\,\Delta, \quad \forall x \in B_p(x_0; \Delta),$$

   - *the error between the gradient of the model and the gradient of the function satisfies*

$$\|\nabla f(x) - \nabla m(x)\|_2 \;\leq\; \kappa_{eg}\,\Delta^2, \quad \forall x \in B_p(x_0; \Delta),$$

   - *and the error between the model and the function satisfies*

$$|f(x) - m(x)| \;\leq\; \kappa_{ef}\,\Delta^3, \quad \forall x \in B_p(x_0; \Delta).$$

   Such a model m is called fully quadratic on $B_p(x_0; \Delta)$.

2. *For this class $\mathcal{M}$ there exists an algorithm, which we will call a 'model-improvement' algorithm, that in a finite, uniformly bounded (with respect to x_0 and Δ) number of steps can*

   - *either provide a certificate for a given model $m \in \mathcal{M}$ that it is fully quadratic on $B_p(x_0; \Delta)$,*
   - *or fail to provide such a certificate and find a model $\tilde{m} \in \mathcal{M}$ fully quadratic on $B_p(x_0; \Delta)$.*

This definition of a fully quadratic class requires that given a model from the class one can either prove that it is a fully quadratic model of f on a given $B_p(x_0; \Delta)$, and for given κ_{ef}, κ_{eg}, κ_{eh}, and ν_2^m, independent of x_0 and Δ, or provide such a model. It is shown in [11, Chapter 6] that model-improvement algorithms exist for quadratic interpolation and regression models. Hence quadratic interpolation models form a fully quadratic class of models. They also exist for a fully linear class of models, where a model-improvement algorithm in [11] checks if a quadratic interpolation model is built using a well-poised set of at least $n + 1$ interpolation points. To certify that a model is fully quadratic, a model-improvement algorithm in [11] requires that the set of the interpolation points is well poised and contains $(n + 1)(n + 2)/2$ points in the proximity of x_0. Thus, using a model-improvement algorithm often implies considerable computational cost: it may be prohibitive to maintain sets of $(n + 1)(n + 2)/2$ sample points near the current iterate due to the cost of obtaining the function values and the dimension of the problem. Moreover, verifying that the sample set is well poised may require a factorization of a matrix with $(n+1)(n+2)/2$ rows and columns resulting in $\mathcal{O}(n^6)$ complexity. For small n, this additional cost may be negligible, but it becomes substantial as n grows beyond a few dozen.

In this paper we show that for any given function f, there exist constants κ_{ef}, κ_{eg}, κ_{eh}, and ν_2^m and the corresponding fully quadratic class of quadratic models $\mathcal{M}$ for which, given x_0 and Δ, we can construct a fully quadratic model of f on $B_p(x_0; \Delta)$ from $\mathcal{M}$, with high probability, using, possibly, less than $(n + 1)(n + 2)/2$ sample points.

Note that Definition 2.2 requires the existence of an algorithm which can *deterministically* certify that a given model is fully quadratic. This requirement is imposed because it enables the

deterministic convergence analysis of an algorithmic framework, provided in [10] (see also [11, Chapter 10]), based on fully quadratic (or fully linear) models. In contrast, in this work, we consider an algorithm which cannot certify that a given model is fully quadratic, but can construct such models with high probability, hopefully, at a considerable computational saving. To adapt this approach in a convergent algorithmic framework, a stochastic version of such a framework has to be designed and analyzed. This work is a subject of future research (see [15] for some relevant theoretical results).

Finally, we want to point out that while the full convergence theory for the new model-based algorithmic framework is under development, the practical implementation reported in this paper shows that in a simple trust-region framework the new method works as well as or better than other methods discussed in [11].

2.2 Quadratic polynomial interpolation models

In model based DFO fully quadratic models of f are often obtained from the class of quadratic polynomials by interpolating f on some sample set of points Y. A detailed description of this process and related theory is given in [11]. Here we present briefly the basic ideas and necessary notation.

Let $\mathcal{P}_n^2$ be the space of polynomials of degree less than or equal to 2 in $\mathbb{R}^n$. The dimension of this space is $q = (n+1)(n+2)/2$. A basis ϕ for $\mathcal{P}_n^2$ will be denoted by $\phi = \{\phi_l(x)\}$ with $l = 1, \ldots, q$. The most natural basis for polynomial spaces is the one consisting of the monomials, or the *canonical basis*. This basis appears naturally in Taylor models and is given for $\mathcal{P}_n^2$ by

$$\bar{\phi} \;=\; \left\{ \frac{1}{2}x_1^2, ..., \frac{1}{2}x_n^2, x_1x_2, ..., x_{n-1}x_n, x_1, ..., x_n, 1 \right\}. \tag{1}$$

We say that the quadratic function m interpolates f at a given point y if $m(y) = f(y)$. Assume that we are given a set $Y = \{y^1, ..., y^p\} \subset \mathbb{R}^n$ of interpolation points. A quadratic function m that interpolates f at the points in Y, written as

$$m(x) \;=\; \sum_{l=1}^{q} \alpha_l \phi_l(x),$$

must satisfy the following p interpolation conditions $\sum_{l=1}^{q} \alpha_l \phi_l(y^i) = f(y^i)$, $i = 1, ..., p$. These conditions form a linear system,

$$M(\phi, Y)\alpha \;=\; f(Y), \tag{2}$$

where $M(\phi, Y)$ is the interpolation matrix and $f(Y)_i = f(y^i),\ i = 1, ..., p$.

A sample set Y is poised for (determined) quadratic interpolation if the corresponding interpolation matrix $M(\phi, Y)$ is square (p=q) and non-singular, guaranteeing that there exists a unique quadratic polynomial m such that $m(Y) = f(Y)$. It is not hard to prove that this definition of poisedness and the uniqueness of the interpolant do not depend either on f or on the basis ϕ (see [11, Chapter 3]).

In [11, Chapters 3 and 6] rigorous conditions on Y are derived which ensure "well poisedness" for quadratic interpolation. Under these conditions it is shown that if $Y \subset B_2(x_0; \Delta)$ is a well-poised sample set for quadratic interpolation, then the quadratic function m that interpolates f on Y is a fully quadratic model for f on $B_2(x_0; \Delta)$ for some fixed positive constants κ_{ef}, κ_{eg}, κ_{eh}, and ν_2^m.

One of the conditions imposed in [11] on a sample set Y to guarantee fully quadratic interpolation model, is that Y has to contain $p = (n+1)(n+2)/2$ points. However, building such a sample set costs $(n+1)(n+2)/2$ evaluations of the function f which is too expensive for many applications. A typical efficient approach is to consider smaller sample sets, which makes the linear system in (2) underdetermined.

2.2.1 Underdetermined quadratic interpolation

We will now consider the case where the size of the sample set Y satisfies $n+1 < p < (n+1)(n+2)/2$, in other words, when there are more points than is required for linear interpolation but fewer than is necessary for determined quadratic interpolation. If we consider the class of all quadratic functions that interpolate f on Y, then we can choose a model from this class that for one reason or other seems the most suitable. In particular approaches in [9] and [29] select a quadratic model with the smallest possible Frobenius norm of the Hessian matrix, while in [22] a model is chosen to minimize the Frobenius norm of the *change* of the Hessian from one iteration to the next. The former approach is studied in detail in [11, Chapter 5]. Let us introduce the basic ideas here.

To properly introduce the underdetermined models that we wish to consider we split the basis $\bar{\phi}$ in (1) into its linear and quadratic components: $\bar{\phi}_L = \{x_1, ..., x_n, 1\}$ and $\bar{\phi}_Q = \bar{\phi} \setminus \bar{\phi}_L$.

An essential property of a sample set Y with $|Y| > n+1$ is that the matrix $M(\bar{\phi}_L, Y)$ must have sufficiently linearly independent columns (in [11, Section 4.4] it is said that Y is well poised for linear regression). Roughly speaking, well poisedness means that under a suitable scaling of Y, $M(\bar{\phi}_L, Y)$ has a relatively small condition number, see [11, Section 4.4]. In that case for *any* quadratic model which interpolates f on Y the following holds (see [11, Theorem 5.4] for a rigorous statement and proof).

Theorem 2.1 *For any $x_0 \in D$ and $\Delta \in (0, \Delta_{max}]$, let m be a quadratic function that interpolates f in Y, where $Y \subset B_2(x_0; \Delta)$ is a sample set well poised for linear regression. Then, m is fully linear (see Definition 2.1) for f on $B_2(x_0; \Delta)$ where the constants κ_{ef} and κ_{eg} are $\mathcal{O}(1 + \|\nabla^2 m\|_2)$ and depend on the condition number of $M(\bar{\phi}_L, Y)$ (where Y here is suitably scaled).*

Theorem 2.1 suggests that one should build underdetermined quadratic models with "small" model Hessians, thus motivating minimizing its Frobenius norm subject to (2) as in [9] and [29]. Recalling the split of the basis $\bar{\phi}$ into the linear and the quadratic parts, one can write the interpolation model as

$$m(x) \;=\; \alpha_Q^T \bar{\phi}_Q(x) + \alpha_L^T \bar{\phi}_L(x),$$

where α_Q and α_L are the corresponding parts of the coefficient vector α. The minimum Frobenius norm solution [11, Section 5.3] can now be defined as the solution to the following optimization problem

$$\begin{array}{ll} \min & \frac{1}{2}\|\alpha_Q\|_2^2 \\ \text{s.t.} & M(\bar{\phi}_Q, Y)\alpha_Q + M(\bar{\phi}_L, Y)\alpha_L \;=\; f(Y). \end{array} \tag{3}$$

(If $|Y| = (n+1)(n+2)/2$ and $M(\bar{\phi}, Y)$ is nonsingular, this reduces to determined quadratic interpolation.) Note that (3) is a convex quadratic program with a closed form solution.

In [11, Section 5.3] it is shown that under some additional conditions of well poisedness on Y, the minimum Frobenius norm (MFN) interpolating model can be fully linear with uniformly

bounded error constants κ_{ef} and κ_{eg}. Hence, the MFN quadratic models provide at least as accurate interpolation as linear models.

On the other hand, it has not been shown so far that any class of underdetermined quadratic interpolation models provide provably better approximation of f than fully linear models. The purpose of this paper is to show how to construct, with high probability, underdetermined quadratic interpolation models that are fully quadratic.

2.2.2 Sparse quadratic interpolation

It is clear that without any additional assumptions on f we cannot guarantee a fully quadratic accuracy by an interpolation model based on less than $(n+1)(n+2)/2$ points. We will thus consider the structure that is most commonly observed and exploited in large-scale derivative based optimization: the (approximate) sparsity of the Hessian of f. Special structure, in particular group partial separability of f, has been exploited in DFO before, see [7]. However, it was assumed that the specific structure is known in advance. In the derivative-free setting, however, while the sparsity structure of the Hessian may be known in some cases, it is often unavailable. Moreover, we do not need to assume that there exists a fixed sparsity structure of the Hessian.

What we assume in this paper is that the Hessian of f is "approximately" sparse in the domain where the model is built. In other words we assume the existence of a sparse fully quadratic model (a rigorous definition is provided in Section 4.3). In the case where $\nabla^2 f$ is itself sparse, a Taylor expansion may serve as such a model. The main focus of our work is to recover sparse quadratic models from the interpolation conditions.

Instead of solving (3) we construct quadratic models from the solution to the following optimization problem

$$\begin{aligned} \min \quad & \|\alpha_Q\|_1 \\ \text{s.t.} \quad & M(\bar{\phi}_Q, Y)\alpha_Q + M(\bar{\phi}_L, Y)\alpha_L \;=\; f(Y), \end{aligned} \tag{4}$$

where α_Q, α_L, $\bar{\phi}_Q$, and $\bar{\phi}_L$ are defined as in (3). Solving (4) is tractable, since it is a linear program. Note that minimizing the ℓ_1-norm of the entries of the Hessian model indirectly controls its ℓ_2-norm and therefore is an appealing approach from the perspective of Theorem 2.1. This makes the new approach a reasonable alternative to building MFN models. As we will show in this paper, this approach is advantageous when the Hessian of f has zero entries (in other words, when there is no direct interaction between some of the variables of the objective function f). In such cases, as we will show in Section 4, we are able to recover, with high probability, fully quadratic models with much less than $(n+1)(n+2)/2$ random points. This is the first result where a fully quadratic model is constructed from an underdetermined interpolation system. To prove this result we will rely on sparse vector recovery theory developed in the field of compressed sensing. In the next section we introduce the basic concepts and results that are involved.

3 Compressed sensing

Compressed sensing is a field concerned with the recovery of a sparse vector $\bar{z} \in \mathbb{R}^N$ satisfying $b = A\bar{z}$, given a vector $b \in \mathbb{R}^k$ and a matrix $A \in \mathbb{R}^{k \times N}$ with significantly fewer rows than columns

($k \ll N$). The desired sparse vector $\bar{z} \in \mathbb{R}^N$ can be recovered by minimizing the number of non-zero components by solving

$$\min \operatorname{card}(z) \quad \text{s. t.} \quad Az = b, \tag{5}$$

where $\operatorname{card}(z) = |\{i \in \{1, \ldots, n\} : z_i \neq 0\}|$. Since this problem is generally NP-Hard, one considers a more tractable approximation by substituting its objective function by a relatively close convex one:

$$\min \|z\|_1 \quad \text{s. t.} \quad Az = b, \tag{6}$$

which is a linear program. The main results of compressed sensing show that, under certain conditions on the (possibly random) matrix A, the solution of (6) is in fact $\bar{z}$ and coincides with the optimal solution of (5) (possibly, with high probability). We will now discuss the compressed sensing results that are useful for our purposes.

3.1 General concepts and properties

One says that a vector z is $s-$sparse if $\operatorname{card}(z) \leq s$. In compressed sensing, one is interested in matrices A such that, for every $s-$sparse vector $\bar{z}$, the information given by $b = A\bar{z}$ is sufficient to recover $\bar{z}$ and, moreover, that such recovery can be accomplished by solving problem (6). The following definition of the *Restricted Isometry Property* (RIP) is introduced in [5].

Definition 3.1 (Restricted Isometry Property) *One says that $\delta_s > 0$ is the Restricted Isometry Property Constant, or* RIP *constant, of order s of the matrix $A \in \mathbb{R}^{k \times N}$ if δ_s is the smallest positive real such that:*

$$(1 - \delta_s) \|z\|_2^2 \leq \|Az\|_2^2 \leq (1 + \delta_s) \|z\|_2^2$$

for every $s-$sparse vector z.

The following theorem (see, e.g., [6, 24]) provides a useful sufficient condition for successful recovery by (6) with $b = A\bar{z}$.

Theorem 3.1 *Let $A \in \mathbb{R}^{k \times N}$ and $2s < N$. If $\delta_{2s} < \frac{1}{3}$, where δ_{2s} is the RIP constant of A of order $2s$, then, for every $s-$sparse vector $\bar{z}$, problem (6) with $b = A\bar{z}$ has a unique solution and it is given by $\bar{z}$.*

Although the RIP provides useful sufficient conditions for sparse recovery, it is a difficult and still open problem to find deterministic matrices which satisfy such a property when the underlying system is highly underdetermined (see [2, 27] for a discussion on this topic). It turns out that random matrices provide a better ground for this analysis (see for instance, one of the results in [4]).

3.2 Partially sparse recovery

To be able to apply sparse recovery results of compressed sensing to our setting we first observe that problem (4) is similar to problem (6), however, it differs in that only a part of the solution vector α is expected to be sparse and appears in the objective function. We hence need to consider an extended recovery result for partial sparsity.

Formally, one has $z = (z_1, z_2)$, where $z_1 \in \mathbb{R}^{N-r}$ is $(s-r)-$sparse and $z_2 \in \mathbb{R}^r$. A natural generalization of problem (6) to this setting of partially sparse recovery is given by

$$\min \|z_1\|_1 \quad \text{s.t.} \quad A_1 z_1 + A_2 z_2 \;=\; b, \tag{7}$$

where $A = (A_1, A_2)$ and A_1 has the first $N-r$ columns of A and A_2 the last r. One can easily see that problem (4) fits into this formulation by setting $z_1 = \alpha_Q$, $z_2 = \alpha_L$, $A_1 = M(\bar{\phi}_Q, Y)$, $A_2 = M(\bar{\phi}_L, Y)$, and $r = n+1$.

We can define an extension of the RIP to the partially sparse recovery setting. Under the assumption that A_2 is full column rank (which in turn is implied by the RIP; see [3]), let

$$\mathcal{P} \;=\; I - A_2 \left(A_2^\top A_2\right)^{-1} A_2^\top \tag{8}$$

be the matrix representing the projection from $\mathbb{R}^N$ onto $\mathcal{R}\left(A_2\right)^\perp$. Then, the problem of recovering $(\bar{z}_1, \bar{z}_2)$, where $\bar{z}_1$ is an $(s-r)$–sparse vector satisfying $A_1\bar{z}_1 + A_2\bar{z}_2 = b$, can be stated as the problem of recovering an $(s-r)-$sparse vector $\bar{z}_1$ satisfying $(\mathcal{P}A_1)\, z_1 = \mathcal{P}b$ and then recovering $\bar{z}_2$ satisfying $A_2 z_2 = b - A_1\bar{z}_1$. The latter task results in solving a linear system given that A_2 has full column rank and $(\mathcal{P}A_1)\bar{z}_1 = \mathcal{P}b$. Note that the former task reduces to the classical setting of compressed sensing. These considerations motivate the following definition of RIP for partially sparse recovery.

Definition 3.2 (Partial RIP Property) *We say that $\delta^r_{s-r} > 0$ is the Partial Restricted Isometry Property Constant of order $s-r$ for recovery of size $N-r$ of the matrix $A = (A_1, A_2) \in \mathbb{R}^{k\times N}$ (with $A_1 \in \mathbb{R}^{k\times(N-r)}$, $A_2 \in \mathbb{R}^{k\times r}$, and $r \le s$) if A_2 is full column rank and δ^r_{s-r} is the RIP constant of order $s-r$ (see Definition 3.1) of the matrix $\mathcal{P}A_1$, where $\mathcal{P}$ is given by (8).*

When $r = 0$ the Partial RIP reduces to the RIP of Definition 3.1. In [3] we show a simple proof of the fact that if a matrix A satisfies RIP for $s-$sparse recovery with δ_s constant, then it also satisfies Partial RIP with $\delta^r_{s-r} = \delta_s$. A very similar result has been independently proved in [18]. It is also shown in [3] that Partial RIP implies that the solution of (7) is the original $s-$sparse solution $\bar{z} = (\bar{z}_1, \bar{z}_2)$. Hence to be able to apply sparse recovery results to problem (4), which is of interest to us, it suffices to construct matrices $M(\bar{\phi}, Y)$ for which the RIP property holds. In [28] a specific sufficient condition for partially sparse recovery is given, but it remains to be seen if we can use such a result to strengthen the bounds on the sample set size which we derive in Section 4. To establish these bounds, we will rely on results on random matrices which apply to our specific setting. We discuss these results in the next section.

4 Recovery of Sparse Hessians

4.1 Sparse recovery using orthonormal bases

For the purposes of building quadratic models based on sparse Hessians we are interested in solving (4) which is equivalent to (7), where $A_1 = M(\phi_Q, Y)$, $A_2 = M(\phi_L, Y)$, $z_1 = \alpha_Q$, $z_2 = \alpha_L$, $b = f(Y)$, and $r = n+1$. In this case ϕ is a basis in the space $\mathcal{P}^2_n$ of polynomials of degree ≤ 2 of dimension $N = (n+1)(n+2)/2$ and the resulting quadratic model m is constructed as

$$m(x) \;=\; \sum_{l=1}^{N} \alpha_l \phi_l(x)$$

where α is the vector of coefficients which is presumed to be sparse (with partially known support since α_L is not necessarily sparse).

Let us now consider a general setting of a finite dimensional space of functions (defined in some domain $\mathcal{D}$) spanned by a basis $\phi = \{\phi_1, ..., \phi_N\}$ of functions (not necessarily polynomial). Let us also consider a function $g : \mathcal{D} \to \mathbb{R}$ which belongs to that space, in other words g can be written as

$$g \;=\; \sum_{j=1}^{N} \alpha_j \phi_j,$$

for some expansion coefficients $\alpha_1, ..., \alpha_N$. We are interested in the problem of recovering g from its values in some finite subset $Y = \{y^1, ..., y^k\} \subset \mathcal{D}$ with $k \leq N$, with the additional assumption that g is $s-$sparse, meaning that the expansion coefficient vector α is $s-$sparse. The purpose of this section is to provide conditions under which such recovery occurs with high probability. Although the results of this section hold also for complex valued functions, we will restrict ourselves to the real case, because the functions we are interested in DFO are real valued. We consider a probability measure μ defined in $\mathcal{D}$ (having in mind that $\mathcal{D} \subset \mathbb{R}^n$). The basis ϕ will be required to satisfy the following orthogonality property [24].

Definition 4.1 (K-bounded orthonormal basis) *A set of functions $\phi = \{\phi_1, ..., \phi_N\}$, spanning a certain function space, is said to be an orthonormal basis satisfying the K-boundedness condition (in the domain $\mathcal{D}$ for the measure μ) if*

$$\int_{\mathcal{D}} \phi_i(x)\phi_j(x)d\mu(x) \;=\; \delta_{ij},$$

(here δ_{ij} is the Kronecker delta) and $\|\phi_j\|_{L^\infty(\mathcal{D})} \leq K$, for all $i, j \in \{1, \ldots, N\}$.

The following theorem (see [24, Theorem 4.4]) shows that by selecting the sample set Y randomly we can recover the sparse coefficient vector with fewer sample points than basis coefficients.

Theorem 4.1 *Let $M(\phi, Y) \in \mathbb{R}^{k\times N}$ be the interpolation matrix associated with an orthonormal basis satisfying the K-boundedness condition. Assume that the sample set $Y = \{y^1, ..., y^k\} \subset \mathcal{D}$ is chosen randomly where each point is drawn independently according to the probability measure μ. Further assume that*

$$\frac{k}{\log k} \;\geq\; c_1 K^2 s (\log s)^2 \log N, \tag{9}$$

$$k \;\geq\; c_2 K^2 s \log\left(\frac{1}{\varepsilon}\right), \tag{10}$$

where $c_1, c_2 > 0$ are universal constants, $\varepsilon \in (0, 1)$, and $s \in \{1, \ldots, N\}$. Then, with probability at least $1-\varepsilon$, $\frac{1}{\sqrt{k}}A = \frac{1}{\sqrt{k}}M(\phi, Y)$ satisfies the RIP property (Definition 3.1) with constant $\delta_{2s} < \frac{1}{3}$.

From the classical results in compressed sensing (see Theorem 3.1 and the paragraph afterwards), this result implies that every $s-$sparse vector $\bar{z} \in \mathbb{R}^N$ is the unique solution to the ℓ_1-minimization problem (6), with $A = M(\phi, Y)$ and $b = M(\phi, Y)\bar{z} = g(Y)$. However, it also implies, by [3, Theorem 4.2], that every $s-$sparse vector $(\bar{z}_1, \bar{z}_2)$ with $(s-r)-$sparse $\bar{z}_1 \in \mathbb{R}^{N-r}$

and possibly dense $\bar{z}_2 \in \mathbb{R}^r$, is the unique solution to the ℓ_1-minimization problem (7), with $A = M(\phi, Y)$ and $b = M(\phi, Y)\bar{z} = g(Y)$. Note that it is a scaled version of A, given by $A/\sqrt{k}$ and not A itself, that satisfies the RIP property but this does not affect the recovery results in the exact setting. As we will see below the scaling has an effect in the noisy case.

It is worth noting that an optimal result is obtained if one sets $\varepsilon = e^{-\frac{k}{c_2 K^2 s}}$ in the sense that (10) is satisfied with equality. Also, from (9) we obtain $k \geq (\log k) c_1 K^2 \ s(\log s)^2 \log N$, and so, using $\log s \geq 1$, $1 - e^{-\frac{k}{c_2 K^2 s}} \geq 1 - N^{-\gamma \log k}$, for the universal constant $\gamma = c_1/c_2$. Thus, ε can be set such that the probability of success $1 - \varepsilon$ satisfies

$$1 - \varepsilon \ \geq \ 1 - N^{-\gamma \log k}, \tag{11}$$

showing that this probability grows polynomially with N and k.

As we observe later in this section, we are not interested in satisfying the interpolation conditions exactly, hence we need to consider instead of $b = g(Y)$ a perturbed version $b = g(Y) + \epsilon$, with a known bound on the size of ϵ. In order to extend the results we just described to the case of noisy recovery, some modifications of problem (6) are needed. In the case of full noisy recovery it is typical to consider, instead of the formulation (6), the following optimization problem:

$$\min \|z\|_1 \quad \text{s. t.} \quad \|Az - b\|_2 \ \leq \ \eta, \tag{12}$$

where η is a positive number. We now present a recovery result based on the formulation (12) and thus appropriate to the noisy case. A proof is available in [6].

Theorem 4.2 *Under the same assumptions of Theorem 4.1, with probability at least $1 - \varepsilon$, $\varepsilon \in (0, 1)$, the following holds for every $s-$sparse vector $\bar{z}$:*

Let noisy samples $b = M(\phi, Y)\bar{z} + \epsilon$ with

$$\|\epsilon\|_2 \ \leq \ \eta$$

be given, for any η non-negative, and let z^* be the solution of the ℓ_1-minimization problem (12) with $A = M(\phi, Y)$. Then,

$$\|z^* - \bar{z}\|_2 \ \leq \ \frac{c_{total}}{\sqrt{k}} \, \eta \tag{13}$$

for some universal constant $c_{total} > 0$.

Since we are interested in the partially sparse recovery case, we need to consider instead

$$\min \|z_1\|_1 \quad \text{s. t.} \quad \|Az - b\|_2 \ \leq \ \eta. \tag{14}$$

The extension of Theorem 4.2 to partially recovery for the noisy case is obtained from the full noisy recovery, analogously to the exact case (see [3, Theorem 5.2] for a proof).

Theorem 4.3 *Under the same assumptions of Theorem 4.1, with probability at least $1 - \varepsilon$, $\varepsilon \in (0, 1)$, the following holds for every vector $\bar{z} = (\bar{z}_1, \bar{z}_2)$ with $r \leq s$ and $\bar{z}_1$ an $(s - r)-$sparse vector:*

Let noisy samples $b = M(\phi, Y)\bar{z} + \epsilon$ with

$$\|\epsilon\|_2 \ \leq \ \eta$$

be given, for any η non-negative, and let $z^* = (z_1^*, z_2^*)$ be the solution of the ℓ_1-minimization problem (14) with $A = M(\phi, Y)$. Then,

$$\|z^* - \bar{z}\|_2 \leq \frac{c_{partial}}{\sqrt{k}}\,\eta, \tag{15}$$

for some universal constant $c_{partial} > 0$.

Note that it is possible to extend these results to approximately sparse vectors (see [3]), however we do not include such an extension in the present paper for the sake of clarity of the exposition.

4.2 Sparse recovery using polynomial orthonormal expansions

As described in Section 2, we are interested in recovering a local quadratic model of the objective function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ near a point x_0. Therefore we consider the space of quadratic functions defined in $B_p(x_0; \Delta)$. To apply the results in Theorems 4.1 and 4.2 we need to build an appropriate orthonormal basis for the space of quadratic functions in $B_p(x_0; \Delta)$. In addition we require that the models we recover are expected to be sparse in such a basis. In this paper we consider models that reconstruct sparse Hessians of f, and thus it is natural to include into the basis polynomials of the forms $c_{ij}\,(x_i - x_0)\,(x_j - x_0)$, with some constant c_{ij} (we will henceforth set $x_0 = 0$ in this section without lost of generality). It is then required that these elements of the basis do not appear as parts of other basis polynomials. The orthonormal basis should satisfy the K-boundedness condition for some constant K independent[1] of the dimension n.

We will now build such an orthonormal basis on the domain $\mathcal{D} = B_\infty(0; \Delta) = [-\Delta, \Delta]^n$ (the ℓ_∞-ball centered at the origin and of radius Δ), using the uniform probability measure μ and the corresponding L^2 inner product.

4.2.1 An orthonormal basis on hypercubes

Let μ be the uniform probability measure on $B_\infty(0; \Delta)$. Note that due to the geometric properties of $B_\infty(0; \Delta) = [-\Delta, \Delta]^n$, one has

$$\int_{[-\Delta,\Delta]^n} g(x_i)h(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)dx \;= \tag{16}$$

$$= \int_{-\Delta}^{\Delta} g(x_i)dx_i \int_{[-\Delta,\Delta]^{n-1}} h(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)dx_1 \cdots dx_{i-1}dx_{i+1} \cdots dx_n,$$

for appropriate functions g and h satisfying the conditions of Fubini's Theorem.

We want to find an orthonormal basis, with respect to μ, of the second degree polynomials on $B_\infty(0; \Delta)$ that contains the polynomials $\{c_{ij}x_ix_j\}_{i\neq j}$. We are considering, first, the off-diagonal part of the Hessian since this is the part which is expected to be sparse (indicating the lack of direct variable interactions). It is easy to see that the $n(n-1)/2$ polynomial functions $\{c_{ij}x_ix_j\}_{i\neq j}$ are all orthogonal, and that due to symmetry all c_{ij} constants are equal, to say,

[1] Otherwise the results in Theorems 4.1 and 4.2 become weaker. Recently, progress has been made in addressing the case when K grows with the dimension, where the main idea is to precondition the interpolation matrix (see [25]).

k_2 (a normalizing constant). Hence we have $n(n-1)/2$ elements of the basis. Now, note that from (16), for different indices i, j, l,

$$\int_{B_\infty(0;\Delta)} x_i x_j x_l d\mu \;=\; \int_{B_\infty(0;\Delta)} x_i x_j d\mu \;=\; \int_{B_\infty(0;\Delta)} x_i x_j^2 d\mu \;=\; 0.$$

As a result, we can add to the set $\{k_2 x_i x_j\}_{i\neq j}$ the polynomials $\{k_1 x_i\}_{1\leq i\leq n}$ and the polynomial k_0, where k_1 and k_0 are normalizing constants, forming a set of $n(n-1)/2 + (n+1)$ orthogonal polynomials.

It remains to construct n quadratic polynomials, which have to contain terms x_i^2 but should not contain terms $x_i x_j$. We choose to consider n terms of the form $k_3(x_i^2 - \alpha_1 x_i - \alpha_0)$. We will select the constants α_0 and α_1 in such a way that these polynomials are orthogonal to the already constructed ones. From the orthogonality with respect to $k_i x_i$, i.e.,

$$\int_{B_\infty(0;\Delta)} x_i(x_i^2 - \alpha_1 x_i - \alpha_0) d\mu \;=\; 0,$$

we must have $\alpha_1 = 0$. Then, orthogonality with respect to the constant polynomial k_0 implies

$$\int_{B_\infty(0;\Delta)} (x_i^2 - \alpha_0) d\mu \;=\; 0.$$

Thus,

$$\alpha_0 \;=\; \frac{1}{2\Delta}\int_{-\Delta}^{\Delta} x^2 dx \;=\; \frac{1}{2\Delta}\left(\frac{2}{3}\Delta^3\right) \;=\; \frac{1}{3}\Delta^2.$$

Hence we have a set of orthogonal polynomials that span the set of quadratic functions on $B_\infty(0;\Delta)$. What remains is the computation of the normalization constants to ensure normality of basis elements. From

$$\int_{B_\infty(0;\Delta)} k_0^2 d\mu \;=\; 1$$

we set $k_0 = 1$. From the equivalent statements

$$\begin{aligned}
\int_{B_\infty(0;\Delta)} (k_1 x_i)^2 \, d\mu \;&=\; 1,\\
\frac{k_1^2}{(2\Delta)^n}\int_{-\Delta}^{\Delta} x^2 dx \int_{[-\Delta,\Delta]^{n-1}} 1 dx \;&=\; 1,\\
k_1^2 \int_{-\Delta}^{\Delta} x^2 \frac{dx}{2\Delta} \;&=\; 1,
\end{aligned}$$

we obtain $k_1 = \sqrt{3}/\Delta$. From the equivalent statements

$$\begin{aligned}
\int_{B_\infty(0;\Delta)} (k_2 x_i x_j)^2 \, d\mu \;&=\; 1,\\
k_2^2 \left(\int_{-\Delta}^{\Delta} x^2 \frac{dx}{2\Delta}\right)^2 \;&=\; 1,
\end{aligned}$$

we conclude that $k_2 = 3/\Delta^2$. And from the equivalent statements

$$\int_{B_\infty(0;\Delta)} \left(k_3 \left(x_i^2 - \frac{1}{3}\Delta^2 \right) \right)^2 d\mu \;=\; 1,$$
$$k_3^2 \int_{-\Delta}^{\Delta} \left(x^2 - \frac{1}{3}\Delta^2 \right)^2 \frac{1}{2\Delta} dx \;=\; 1,$$

we obtain

$$k_3 \;=\; \frac{3\sqrt{5}}{2\Delta^2}.$$

We have thus constructed the desirable basis, which we will denote by ψ. We will abuse the notation to define ψ using indices $(0), (1,i)$, $(2,ij)$ or $(2,i)$ for the elements of ψ in place of the single index l. The expressions of these sophisticated indices should simplify the understanding. For instance, $(2,ij)$ index stands for the element of the basis ψ which involves the term $x_i x_j$, similarly $\alpha_{(2,i)}$ is the term corresponding to x_i^2, and so on.

Definition 4.2 *We define the basis ψ as the set of the following $(n+1)(n+2)/2$ polynomials:*

$$\left\{ \begin{array}{rcl} \psi_{2,i}(x) & = & \frac{3\sqrt{5}}{2\Delta^2} x_i^2 - \frac{\sqrt{5}}{2}, \\ \psi_{2,ij}(x) & = & \frac{3}{\Delta^2} x_i x_j, \\ \psi_{1,i}(x) & = & \frac{\sqrt{3}}{\Delta} x_i, \\ \psi_0(x) & = & 1. \end{array} \right. \tag{17}$$

The basis ψ satisfies the assumptions of Theorems 4.1 and 4.2, as stated in the following theorem.

Theorem 4.4 *The basis ψ (see Definition 4.2) is orthonormal and satisfies the K-boundedness condition (see Definition 4.1) in $B_\infty(0;\Delta)$ for the uniform probability measure with $K = 3$.*

Proof. From the above derivation and (16) one can easily show that ψ is orthonormal in $B_\infty(0;\Delta)$ with respect to the uniform probability measure. So, it remains to prove the boundedness condition with $K = 3$. In fact, it is easy to check that

$$\left\{ \begin{array}{rcll} \|\psi_{2,i}\|_{L^\infty(B_\infty(0;\Delta))} & = & \sqrt{5} & \leq 3, \\ \|\psi_{2,ij}\|_{L^\infty(B_\infty(0;\Delta))} & = & 3 & \leq 3, \\ \|\psi_{1,i}\|_{L^\infty(B_\infty(0;\Delta))} & = & \sqrt{3} & \leq 3, \\ \|\psi_0\|_{L^\infty(B_\infty(0;\Delta))} & = & 1 & \leq 3, \end{array} \right. \tag{18}$$

where $\|g\|_{L^\infty(B_\infty(0;\Delta))} = \max_{x \in B_\infty(0;\Delta)} |g(x)|$. ■

We will consider ψ_Q, the subset of ψ consisting of the polynomials of degree 2, and ψ_L, the ones of degree 1 or 0, as we did in Section 2 for $\bar{\phi}$.

We are interested in quadratic functions $m = \sum_l \alpha_l \psi_l$ (see Definition 4.2) with an $h-$sparse coefficient subvector α_Q, i.e., only h coefficients corresponding to the polynomials in ψ_Q in the representation of m are non-zero, where h is a number between 1 and $n(n+1)/2$. In such cases, the corresponding full vector α of coefficients is $(h+n+1)-$sparse. We now state a corollary of Theorem 4.2 for sparse recovery in the orthonormal basis ψ, with $k = p$ (number of sample points) and $N = q$ (number of elements in ψ), which will be used in the next section to establish results on sparse quadratic model recovery. Note that we write the probability of successful recovery of a sparse solution in the form $1 - n^{-\gamma \log p}$ which can be derived from (11) using $q = \mathcal{O}(n^2)$ and a simple modification of the universal constant γ.

Corollary 4.1 *Let $M(\psi, Y) \in \mathbb{R}^{p\times q}$ be the matrix with entries $[M(\psi, Y)]_{ij} = \psi_j(y^i)$, $i = 1, ..., p$, $j = 1, ..., q$, with $q = (n + 1)(n + 2)/2$.*

Assume that the sample set $Y = \{y^1, ..., y^p\} \subset B_\infty(0; \Delta)$ is chosen randomly where each point is drawn independently according to the uniform probability measure μ in $B_\infty(0; \Delta)$. Further assume that

$$\frac{p}{\log p} \;\geq\; 9c\,(h + n + 1)\,(\log\,(h + n + 1))^2 \log q,$$

for some universal constant $c > 0$ and $h \in \{1, ..., n(n + 1)/2\}$. Then, with probability at least $1 - n^{-\gamma \log p}$, for some universal constant $\gamma > 0$, the following holds for every vector $\bar{z}$, having at most $h + n + 1$ non-zero expansion coefficients in the basis ψ:

Let noisy samples $b = M(\psi, Y)\bar{z} + \epsilon$ with

$$\|\epsilon\|_2 \;\leq\; \eta$$

be given, for any η non-negative, and let z^* be the solution of the ℓ_1-minimization problem (14) with $A = M(\psi, Y) = (M(\psi_Q, Y), M(\psi_L, Y))$. Then,

$$\|z^* - \bar{z}\|_2 \;\leq\; \frac{c_{partial}}{\sqrt{p}}\,\eta$$

for some universal constant $c_{partial} > 0$.

Remark 4.1 *It would be natural to consider the interpolation domain to be the ball $B_2(0; \Delta)$ in the classical ℓ_2-norm. However, our procedure of constructing an orthonormal set of polynomials with desired properties in the hypercube, i.e., using the ℓ_∞-norm ball, does not extend naturally to the ℓ_2 one. One problem with the uniform measure in the ℓ_2-ball is that formulas like (16) no longer hold. A construction of an appropriate basis for the uniform measure on the ℓ_2-ball is a subject for further work.*

4.3 Recovery of a fully quadratic model of a function with sparse Hessian

As we stated earlier our main interest in this paper is to recover a fully quadratic model (see Definition 2.2) of a twice continuously differentiable objective function $f : D \to \mathbb{R}$ near a point x_0 using fewer than $(n+1)(n+2)/2$ sample points. In other words, we want to show that for a given function f there exist constants κ_{ef}, κ_{eg}, and κ_{eh} such that, given any point x_0 and a radius Δ, we can build a model based on a random sample set of p points (with $p < (n + 1)(n + 2)/2$) which, with high probability, is a fully quadratic model of f on $B_p(x_0; \Delta)$ with respect to the given constants κ_{ef}, κ_{eg}, and κ_{eh}. The number p of sample points depends on the sparsity of the Hessian of the model that we are attempting to reconstruct. Hence we need to make some assumption about the sparsity. The simplest (and strongest) assumption we can make is that the function f has a sparse Hessian at any point x_0.

Assumption 4.1 (Hessian sparsity) *Assume that $f : D \to \mathbb{R}$ satisfies Assumption 2.2 and furthermore that for any given $x_0 \in D$ the Hessian $\nabla^2 f(x_0)$ of f at x_0 has at most h non-zero entries, on or above the diagonal, where h is a number between 1 and $n(n + 1)/2$. If this is the case, then $\nabla^2 f$ is said to be $h-$sparse.*

The above assumption implies that for every $x_0 \in D$ there exists a fully quadratic second degree polynomial model q_f of f such that the Hessian $\nabla^2 q_f$ is $h-$sparse, from a fully quadratic class with κ'_{ef}, κ'_{eg}, and κ'_{eh} equal to some multiples of the Lipschitz constant of $\nabla^2 f$. The second order Taylor model at x_0 is, in particular, such a model.

However, we do not need this strong assumption to be able to construct fully quadratic models. Constructing models via random sample sets and ℓ_1-minimization, in the way that we described above, provides fully quadratic models regardless of the amount of sparsity of the Hessian, as we will show in this section. The sparsity of the Hessian affects, however, the number of sample points that are required. Hence, one can consider functions whose Hessian is approximately sparse and the sparsity pattern, or even the cardinality (number of non-zeros), is not constant. The following assumption is weaker than Assumption 4.1 but is sufficient for our purposes.

Assumption 4.2 (Approximate Hessian sparsity) *Assume that $f : D \to \mathbb{R}$ satisfies Assumption 2.2 and furthermore that for any given $x_0 \in D$ and $\Delta > 0$ there exists a second degree polynomial $m(x) = \sum_l \alpha_l \psi_l(x) = \alpha_Q \psi_Q(x) + \alpha_L \psi_L(x)$, with α_Q an $h-$sparse coefficient vector, where h may depend on x_0 and Δ, which is a fully quadratic model of f on $B_p(x_0; \Delta)$ for some constants κ'_{ef}, κ'_{eg}, and κ'_{eh}, independent of x_0 and Δ.*

If in the above assumption h is independent of x_0 and Δ, then the assumption reduces to Assumption 4.1. As it stands, Assumption 4.2 is less restrictive.

Given the result in Section 4.2, we will consider the ℓ_∞-norm in Definition 2.2, thus considering regions of the form $B_\infty(x_0; \Delta)$.

When we state that f has a sparse Hessian, it is understood that the representation of the Taylor second order expansion, or of any other fully quadratic model of f, is a sparse linear combination of the elements of the canonical basis $\bar{\phi}$ (see (1)). However, the basis $\bar{\phi}$ is not orthogonal on $B_\infty(x_0; \Delta)$. Hence we are interested in models that have a sparse representation in the orthonormal basis ψ of Definition 4.2. Fortunately, basis ψ can be obtained from $\bar{\phi}$ through a few simple transformations. In particular, the sparsity of the Hessian of a quadratic model m will be carried over to sparsity in the representation of m in ψ, since the expansion coefficients in ψ_Q will be multiples of the ones in $\bar{\phi}_Q$, thus guaranteeing that if the coefficients in the latter are $h-$sparse, so are the ones in the former.

We are now able to use the material developed in Section 4.2 to guarantee, with high probability, the construction, for each x_0 and Δ, of a fully quadratic model of f in $B_\infty(x_0; \Delta)$ using a random sample set of only $\mathcal{O}(n(\log n)^4)$ points, instead of $\mathcal{O}(n^2)$ points, provided that $h = \mathcal{O}(n)$ (for the given x_0 and Δ, see Assumption 4.2). We find such a fully quadratic model by solving the partially sparse recovery version of problem (12) written now in the form

$$\begin{aligned} \min \quad & \|\alpha_Q\|_1 \\ \text{s.t.} \quad & \|M(\psi_Q, Y)\alpha_Q + M(\psi_L, Y)\alpha_L - f(Y + x_0)\|_2 \ \leq \ \eta, \end{aligned} \tag{19}$$

where η is some appropriate positive quantity and Y is drawn in $B_\infty(0; \Delta)$. Corollary 4.1 can then be used to ensure that only $\mathcal{O}(n(\log n)^4)$ points are necessary for recovery of a sparse model in $B_\infty(0; \Delta)$, when the number of non-zero components of the Hessian of m in Assumption 4.2 is of the order of n.

Note that we are in fact considering "noisy" measurements, because we are only able to evaluate the function f while trying to recover a fully quadratic model, whose values are somewhat

different from those of f. We will say that a function q^* is the solution to the minimization problem (19) if $q^*(x) = \sum_l \alpha_l^* \psi_l(x)$, where α^* is the minimizer of (19).

First we need to prove an auxiliary lemma. Corollary 4.1 provides an estimate on the ℓ_2-norm of the error in the recovered vector of coefficients of the quadratic model. In the definition of fully quadratic models, the error between the quadratic model and the function f is measured in terms of the maximum difference of their function values in $B_\infty(x_0; \Delta)$ and the maximum norms of the differences of their gradients and their Hessians in $B_\infty(x_0; \Delta)$. The following lemma establishes a bound for the value, gradient, and Hessian of quadratic polynomials in $B_\infty(0; \Delta)$ in terms of the norm of their coefficient vector (using the basis ψ).

Lemma 4.1 *Let m be a quadratic function and α be a vector in $\mathbb{R}^{(n+1)(n+2)/2}$ such that*

$$m(x) \;=\; \sum_l \alpha_l \psi_l(x)$$

with $\psi(x)$ defined in (17). Then

$$\begin{aligned} |m(x)| &\leq \left(3\sqrt{\mathrm{card}(\alpha)}\right) \|\alpha\|_2 \\ \|\nabla m(x)\|_2 &\leq \left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha)}\right) \frac{1}{\Delta} \|\alpha\|_2 \\ \left\|\nabla^2 m(x)\right\|_2 &\leq \left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha)}\right) \frac{1}{\Delta^2} \|\alpha\|_2, \end{aligned}$$

for all $x \in B_\infty(0; \Delta)$, where $\mathrm{card}(\alpha)$ is the number of non-zero elements in α.

Proof. We will again use the indices $(0), (1,i)$, $(2,ij)$ or $(2,i)$ for the elements of α in correspondence to the indices used in Definition 4.2.

From the K-boundedness conditions (18) we have

$$|m(x)| \;\leq\; \sum_l |\alpha_l||\psi_l(x)| \;\leq\; 3\|\alpha\|_1 \;\leq\; 3\sqrt{\mathrm{card}(\alpha)}\|\alpha\|_2,$$

for all $x \in B_\infty(0; \Delta)$. Also, from (17),

$$\begin{aligned} \left|\frac{\partial m}{\partial x_i}(x)\right| &\leq \sum_l |\alpha_l| \left|\frac{\partial \psi_l}{\partial x_i}\right| \\ &= |\alpha_{1,i}| \left|\frac{\sqrt{3}}{\Delta}\right| + \sum_{j \in \{1,\ldots,n\}\setminus\{i\}} |\alpha_{2,ij}| \left|\frac{3}{\Delta^2} x_j\right| + |\alpha_{2,i}| \left|\frac{3\sqrt{5}}{\Delta^2} x_i\right| \\ &\leq \frac{\sqrt{3}}{\Delta}|\alpha_{1,i}| + \sum_{j \in \{1,\ldots,n\}\setminus\{i\}} \frac{3}{\Delta}|\alpha_{2,ij}| + \frac{3\sqrt{5}}{\Delta}|\alpha_{2,i}|. \end{aligned}$$

Then, by the known relations between the norms ℓ_1 and ℓ_2,

$$\begin{aligned} \|\nabla m(x)\|_2 \;\leq\; \|\nabla m(x)\|_1 &\leq \sum_{i=1}^n \left(\frac{\sqrt{3}}{\Delta}|\alpha_{1,i}| + \sum_{j \in \{1,\ldots,n\}\setminus\{i\}} \frac{3}{\Delta}|\alpha_{2,ij}| + \frac{3\sqrt{5}}{\Delta}|\alpha_{2,i}| \right) \\ &\leq \sum_{i=1}^n \frac{\sqrt{3}}{\Delta}|\alpha_{1,i}| + \sum_{i,j \in \{1,\ldots,n\}, j>i} \frac{6}{\Delta}|\alpha_{2,ij}| + \sum_{i=1}^n \frac{3\sqrt{5}}{\Delta}|\alpha_{2,i}| \\ &\leq \left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha)}\right) \frac{1}{\Delta}\|\alpha\|_1, \end{aligned}$$

for all $x \in B_\infty(0;\Delta)$.

For the estimation of the Hessian, we need to separate the diagonal from the non-diagonal part. For the non-diagonal part, with $i \neq j$,

$$\left|\frac{\partial^2 m}{\partial x_i \partial x_j}(x)\right| \;\leq\; \sum_l |\alpha_l| \frac{\partial^2 \psi_l(x)}{\partial x_i \partial x_j} \;\leq\; |\alpha_{2,ij}| \frac{3}{\Delta^2}.$$

For the diagonal part, with $i = 1, \ldots, n$,

$$\left|\frac{\partial^2 m}{\partial x_i^2}(x)\right| \;\leq\; \sum_l |\alpha_l| \frac{\partial^2 \psi_l(x)}{\partial x_i^2} \;\leq\; |\alpha_{2,i}| \frac{3\sqrt{5}}{\Delta^2}.$$

Since the upper triangular part of the Hessian has at most $\mathrm{card}(\alpha)$ non-zero components one has

$$\begin{aligned}
\left\|\nabla^2 m(x)\right\|_F &= \sqrt{\sum_{i,j\in\{1,\ldots,n\}} \left(\frac{\partial^2 m}{\partial x_i \partial x_j}(x)\right)^2} \\
&\leq \sum_{i,j\in\{1,\ldots,n\}} \left|\frac{\partial^2 m}{\partial x_i \partial x_j}(x)\right| \\
&\leq \sum_{i,j\in\{1,\ldots,n\}, j>i} |\alpha_{2,ij}| \frac{6}{\Delta^2} + \sum_{i\in\{1,\ldots,n\}} |\alpha_{2,i}| \frac{3\sqrt{5}}{\Delta^2} \\
&\leq \sqrt{\mathrm{card}(\alpha)} \left(\frac{3\sqrt{5}}{\Delta^2}\right) \|\alpha\|_2.
\end{aligned}$$

Thus,

$$\left\|\nabla^2 m(x)\right\|_2 \;\leq\; \left\|\nabla^2 m(x)\right\|_F \;\leq\; \sqrt{\mathrm{card}(\alpha)} \left(\frac{3\sqrt{5}}{\Delta^2}\right) \|\alpha\|_2$$

for all $x \in B_\infty(0;\Delta)$. ■

Remark 4.2 *The dependency of the error bounds in Lemma 4.1 on* $\mathrm{card}(\alpha)$ *cannot be eliminated. In fact, the quadratic function*

$$g(x) \;=\; \sum_{i,j\in\{1,\ldots,n\}, j>i} \sqrt{\frac{2}{n(n-1)}} \frac{3}{\Delta^2} x_i x_j$$

satisfies $g(\Delta, ..., \Delta) = 3\sqrt{(n(n-1))/2} = 3\sqrt{\mathrm{card}(\alpha)}$ *while the vector of coefficients* α *has norm equal to* 1.

Remark 4.3 *Since* ψ *is orthonormal (with respect to* μ*) on* $B_\infty(0;\Delta)$ *we have that* $\|\alpha\|_2 = \|m\|_{L^2(B_\infty(0;\Delta),\mu)}$. *Hence the* ℓ_2*-norm of the vector of the coefficients is simply the* L^2*-norm of the function* m *over* $B_\infty(0;\Delta)$. *If we now consider* $\|m\|_{L^\infty(B_\infty(0;\Delta))}$, *the* L^∞*-norm of* m *which is the maximum absolute value of* $m(x)$ *over* $B_\infty(0;\Delta)$, *we see that Lemma 4.1 establishes a relation between two norms of* m. *By explicitly deriving constants in terms of* $\mathrm{card}(\alpha)$ *we strengthen the bounds in Lemma 4.1 for the cases of sparse models.*

We are now ready to present our main result.

Theorem 4.5 *Let Assumption 4.2 hold (approximate Hessian sparsity). Given Δ and x_0, let h be the corresponding sparsity level of the fully quadratic model guaranteed by Assumption 4.2. Let $Y = \{y^1, ..., y^p\}$ be a given set of p random points, chosen with respect to the uniform measure in $B_\infty(0;\Delta)$, with*

$$\frac{p}{\log p} \;\geq\; 9c_{partial}\,(h+n+1)\,(\log\,(h+n+1))^2\log q, \tag{20}$$

for some universal constant $c_{partial} > 0$, then with probability larger than $1 - n^{-\gamma\log p}$, for some universal constant $\gamma > 0$, the quadratic $m^*(x) = \tilde{m}^*(x - x_0)$, where $\tilde{m}^*$ is the solution to the ℓ_1-minimization problem (19), is a fully quadratic model of f on $B_\infty(x_0;\Delta)$ with $\nu_2^{m^*} = 0$ and constants κ_{ef}, κ_{eg}, and κ_{eh} not depending on x_0 and Δ.

Proof. From Assumption 4.2, there exists a fully quadratic model m for f on $B_\infty(x_0;\Delta)$ with $\nu_2^m = 0$ and some constants κ'_{ef}, κ'_{eg}, and κ'_{eh}. The quadratic polynomial $\tilde{m}(z) = m(z+x_0)$, $z \in B_\infty(0;\Delta)$, satisfies the assumptions of Corollary 4.1 and, for the purpose of the proof, is the quadratic that will be approximately recovered. Now, since m is a fully quadratic model, we have $|f(y^i + x_0) - m(y^i + x_0)| \leq \kappa'_{ef}\Delta^3$, $y^i \in B_\infty(0;\Delta)$. Therefore

$$\|f(Y + x_0) - m(Y + x_0)\|_2 \;\leq\; \sqrt{p}\,\kappa'_{ef}\Delta^3,$$

for any sample set $Y \subset B_\infty(0;\Delta)$ (where κ'_{ef} is independent of x_0 and Δ). Note that one can only recover m approximately given that the values of $m(Y + x_0) \simeq f(Y + x_0)$ are 'noisy'.

Consider again $\tilde{m}(z) = m(z + x_0)$, $z \in B_\infty(0;\Delta)$. Then, by Corollary 4.1, with probability larger than $1 - n^{-\gamma\log p}$, for a universal constant $\gamma > 0$, the solution $\tilde{m}^*$ to the ℓ_1-minimization problem (19) with $\eta = \sqrt{p}\,\kappa'_{ef}\Delta^3$ satisfies

$$\|\alpha^* - \alpha\|_2 \;\leq\; c_{partial}\kappa'_{ef}\Delta^3,$$

where α^* and α are the coefficients of $\tilde{m}^*$ and $\tilde{m}$ in the basis ψ given by (17), respectively. Note that $c_{partial}$ does not depend on x_0. So, by Lemma 4.1,

$$\begin{aligned}
|\tilde{m}^*(z) - \tilde{m}(z)| &\leq c_{partial}\left(3\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef}\Delta^3,\\
\|\nabla\tilde{m}^*(z) - \nabla\tilde{m}(z)\|_2 &\leq c_{partial}\left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef}\Delta^2,\\
\left\|\nabla^2\tilde{m}^*(z) - \nabla^2\tilde{m}(z)\right\|_2 &\leq c_{partial}\left(3\sqrt{10}\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef}\Delta,
\end{aligned}$$

for all $z \in B_\infty(0;\Delta)$. Note that α^* and α depend on x_0 but $\mathrm{card}(\alpha^* - \alpha)$ can be easily bounded independently of x_0. Let now $m^*(x) = \tilde{m}^*(x - x_0)$ for $x \in B_\infty(x_0;\Delta)$. Therefore, from the fact that m is fully quadratic (with constants κ'_{ef}, κ'_{eg}, and κ'_{eh}), one has

$$\begin{aligned}
|m^*(x) - f(x)| &\leq \left(c_{partial}\left(3\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef} + \kappa'_{ef}\right)\Delta^3,\\
\|\nabla m^*(x) - \nabla f(x)\|_2 &\leq \left(c_{partial}\left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef} + \kappa'_{eg}\right)\Delta^2,\\
\left\|\nabla^2 m^*(x) - \nabla^2 f(x)\right\|_2 &\leq \left(c_{partial}\left(3\sqrt{5}\sqrt{\mathrm{card}(\alpha^* - \alpha)}\right)\kappa'_{ef} + \kappa'_{eh}\right)\Delta,
\end{aligned}$$

for all $x \in B_\infty(x_0; \Delta)$.

Since m^* is a quadratic function, its Hessian is Lipschitz continuous with Lipschitz constant 0, so one has that $\nu_2^{m^*} = 0$. Hence m^* is a fully quadratic model of f on $B_\infty(x_0; \Delta)$. ■

Note that the result of Theorem 4.5 is obtained for a number p of sampling points satisfying (see (20) and recall that $q = \mathcal{O}(n^2)$)

$$\frac{p}{\log p} \;=\; \mathcal{O}(n(\log n)^3)$$

when $h = \mathcal{O}(n)$, i.e., when the number of non-zero elements of the Hessian of f at x_0 is of the order of n. Since $p < (n+1)(n+2)/2$, one obtains

$$p \;=\; \mathcal{O}\left(n(\log n)^4\right). \tag{21}$$

Theorem 4.5 does not directly assume Hessian sparsity of f. It is worth observing again that Theorem 4.5 can be established under no conditions on the sparsity *pattern* of the Hessian of f.

Problem (19) is a second order cone programming problem [1] and can, hence, be solved in polynomial time. However it is typically easier in practice to solve linear programming problems. Since the second order Taylor model T satisfies $\|T(Y + x_0) - f(Y + x_0)\|_\infty \leq \eta/\sqrt{p}$ (where $\eta = \sqrt{p}\,\kappa'_{ef}\Delta^3$), because T is fully quadratic for f, instead of (19), one can consider

$$\begin{array}{ll} \min & \left\|\alpha_Q^m\right\|_1 \\ \text{s.t.} & \left\|M(\psi_Q, Y)\alpha_Q^m + M(\psi_L, Y)\alpha_L^m - f(Y + x_0)\right\|_\infty \;\leq\; \frac{1}{\sqrt{p}}\eta, \end{array}$$

which is a linear program. In our implementation, as we will discuss in the next section, we chose to impose the interpolation constraints exactly which corresponds to setting $\eta = 0$ in the above formulations, hence simplifying parameter choices. Also, recent work has provided some insight for why this choice works well (see [30]).

Theorem 4.5 cannot strictly validate a practical setting in DFO like the one discussed in the next section. It serves to provide motivation and insight on the use of ℓ_1-minimization to build underdetermined quadratic models for functions with sparse Hessians. It also is the first result, to our knowledge, that establishes a reasonable approach to building fully quadratic models with underdetermined interpolation, when the sparsity structure of the objective function is not known. However, in the current implementation the sampling is done deterministically in order to be able to reuse existing sample points. This may be lifted in future parallel implementations. Note that the constants in the bound (20) (and thus in (21)) render the current bounds impractical. In fact, the best known upper bound (see [24]) for the universal constant c_{total} appearing in (13) is $c_{total} < 17190$ ($c_{partial}$ is of the same order), making (20) only applicable if n is much greater than the values for which DFO problems are tractable today by deterministic algorithms. However, such a bound is most likely not tight; in fact, similar universal constants appearing in the setting of compressed sensing are known to be much smaller in practice.

5 A practical interpolation-based trust-region method

5.1 Interpolation-based trust-region algorithms for DFO

Trust-region methods are a well known class of algorithms for the numerical solution of nonlinear programming problems [8, 20]. In this section we will give a brief summary of these methods

when applied to the unconstrained minimization of a smooth function $f : D \subset \mathbb{R}^n \to \mathbb{R}$,

$$\min_{x \in \mathbb{R}^n} \; f(x), \tag{22}$$

without using the derivatives of the objective function f. For comprehensive coverage we refer the reader to [11].

At each iteration k, these methods build a model $m_k(x_k + s)$ of the objective function in a *trust region* of the form $B_p(x_k; \Delta_k)$, typically with $p = 2$, around the current iterate x_k. The scalar Δ_k is then called the trust-region radius. A step s_k is determined by solving the trust-region subproblem

$$\min_{s \in B_2(0;\Delta_k)} \; m_k(x_k + s). \tag{23}$$

Then, the value $f(x_k + s_k)$ is computed and the actual reduction in the objective function ($ared_k = f(x_k) - f(x_k + s_k)$) is compared to the predicted reduction in the model ($pred_k = m_k(x_k) - m_k(x_k + s_k)$). If the ratio is big enough ($\rho_k = ared_k/pred_k \geq \eta_1 \in (0, 1)$), then $x_k + s_k$ is accepted as the new iterate and the trust-region radius may be increased. Such iterations are called successful. If the ratio is small ($\rho_k < \eta_1$), then the step is rejected and the trust-region radius is decreased. Such iterations are called unsuccessful.

The global convergence properties of these methods are strongly dependent on the requirement that, as the trust region becomes smaller, the model becomes more accurate, implying in particular that the trust-region radius is bounded away from zero, as long as the stationary point is not reached. Taylor based-models, when derivatives are known, naturally satisfy this requirement. However, in the DFO setting, some provision has to be taken in the model and sample set management to ensure global convergence. These provisions aim at guaranteeing that the models produced by the algorithm are fully linear or fully quadratic, and to guarantee global convergence this must be done in arbitrarily smaller trust regions thus driving the trust-region radius to zero (such a procedure is ensured by the so-called criticality step, which has indeed be shown necessary [26]).

Conn, Scheinberg, and Vicente [10] proved global convergence to first and second order stationary points depending whether fully linear or fully quadratic models are used. The approach proposed in [10] involves special model improving iterations. Scheinberg and Toint [26] have recently shown global convergence to first order stationary points for their *self-correcting geometry* approach which replaces model-improving iterations by an appropriate update of the sample set using only the new trust-region iterates.

Our results derived in Section 4 provide us with a new method to produce (with high probability) fully quadratic models by considering randomly sampled sets, instead of model handling iterations as is done in [10] and [26]. On the other hand, to develop full convergence theory of DFO methods using randomly sampled sets, one needs to adapt the convergence proofs used in [10] and [26] to the case where successful iterations are guaranteed with high probability, rather than deterministically. This is a subject for future research.

5.2 A practical interpolation-based trust-region method

We now introduce a simple practical algorithm which we chose for testing the performance of different underdetermined models. This algorithm follows some of the basic ideas of the approach introduced by Fasano, Morales, and Nocedal [14], which have also inspired the authors in [26]. The quality of the sample sets is maintained in its simplest form — simply ensuring sufficient

number of sample points ($n+1$ or more) in a reasonable proximity from the current iterate. This approach is theoretically weak as shown in [26], but seems to work well in practice.

Unlike [14], we discard the sample point farthest away from the new iterate (rather than the sample point farthest away from the current iterate). Also, in [14], only determined quadratic models were built based on $p_{\max} = (n+1)(n+2)/2$ sample points. We compare approaches that use minimum Frobenius or ℓ_1 norm interpolation to build the models and hence we allow sample sets of any size less than or equal to $p_{\max}$. This poses additional issues to those considered in [14]. For instance, until the cardinality of the sample set reaches $p_{\max}$, we do not discard points from the sample set and always add new trial points independently of whether or not they are accepted as new iterates, in an attempt to be as greedy as possible when taking advantage of function evaluations.

Another difference from [14] is that we discard points that are too far from the current iterate when the trust-region radius becomes small (this can be viewed as a weak criticality condition), expecting that the next iterations will refill the sample set resulting in a similar effect as a criticality step. Thus, the cardinality of our sample set might fall below $p_{\min} = n+1$, the number required to build fully linear models in general. In such situations, we never reduce the trust-region radius.

Algorithm 5.1 (A practical DFO trust-region algorithm)
Step 0: Initialization.

Initial values. *Select values for the constants $\epsilon_g(= 10^{-5}) > 0$, $\delta(= 10^{-5}) > 0$, $0 < \eta_1(= 10^{-3})$, $\eta_2(= 0.75) > \eta_1$, and $0 < \gamma_1(= 0.5) < 1 < \gamma_2(= 2)$. Set $p_{\min} = n+1$ and $p_{\max} = (n+1)(n+2)/2$. Set the initial trust-region radius $\Delta_0(= 1) > 0$. Choose the norm $t = 1$ (the ℓ_1-norm) or $t = 2$ (the Frobenius norm).*

Initial sample set. *Let the starting point x_0 be given. Select as an initial sample set $Y_0 = \{x_0, x_0 \pm \Delta_0 e_i, i = 1, \ldots, n\}$, where the e_i's are the columns of the identity matrix of order n.*

Function evaluations. *Evaluate the objective function at all $y \in Y_0$.*

Set $k = 0$.
Step 1: Model building.

Form a quadratic model $m_k(x_k + s)$ of the objective function from Y_k. Solve the problem

$$\begin{array}{ll} \min & \frac{1}{p}\|\alpha_Q\|_t^t \\ \text{s.t.} & M(\bar{\phi}_Q, Y_k)\alpha_Q + M(\bar{\phi}_L, Y_k)\alpha_L \;=\; f(Y_k), \end{array} \tag{24}$$

where α_Q and α_L are, respectively, the coefficients of order 2 and order less than 2 of the model.

Step 2: Stopping criteria.

Stop if $\|g_k\| \le \epsilon_g$ or $\Delta_k \le \delta$.

Step 3: Step calculation.

Compute a step s_k by solving (approximately) the trust-region subproblem (23).

Step 4: Function evaluation.

Evaluate the objective function at $x_k + s_k$. *Compute* $\rho_k = (f(x_k) - f(x_k + s_k))/(m_k(x_k) - m_k(x_k + s_k))$.

Step 5: Selection of the next iterate and trust radius update.

If $\rho_k < \eta_1$, *reject the trial step, set* $x_{k+1} = x_k$, *and reduce the trust-region radius, if* $|Y_k| \geq p_{\min}$, *by setting* $\Delta_k = \gamma_1 \Delta_k$ (**unsuccessful iteration**).

If $\rho_k \geq \eta_1$, *accept the trial step* $x_{k+1} = x_k + s_k$ (**successful iteration**).

Increase the trust-region radius, $\Delta_{k+1} = \gamma_2 \Delta_k$, *if* $\rho_k > \eta_2$.

Step 6: Update the sample set.

If $|Y_k| = p_{\max}$, *set* $y_k^{out} \in \operatorname{argmax} \|y - x_{k+1}\|_2$ *(break ties arbitrarily).*

If the iteration was successful:

If $|Y_k| = p_{\max}$, $Y_{k+1} = Y_k \cup \{x_{k+1}\} \setminus \{y_k^{out}\}$.
If $|Y_k| < p_{\max}$, $Y_{k+1} = Y_k \cup \{x_{k+1}\}$.

If the iteration was unsuccessful:

If $|Y_k| = p_{\max}$, $Y_{k+1} = Y_k \cup \{x_k + s_k\} \setminus \{y_k^{out}\}$ *if* $\|(x_k + s_k) - x_k\|_2 \leq \|y_k^{out} - x_k\|_2$.
If $|Y_k| < p_{\max}$, $Y_{k+1} = Y_k \cup \{x_k + s_k\}$.

Step 7: Model improvement.

When $\Delta_{k+1} < 10^{-3}$, *discard from* Y_{k+1} *all the points outside* $B(x_{k+1}; r\Delta_{k+1})$, *where* r *is chosen as the smallest number in* $\{100, 200, 400, 800, ...\}$ *for which at least three sample points from* Y_{k+1} *are contained in* $B(x_{k+1}; r\Delta_{k+1})$.

Increment k *by* 1 *and return to Step 1.*

We note that relying on the model gradient to stop might not be a reliable stopping criterion, as we need to resample the function inside a smaller trust region and construct a new model before safely quitting (criticality step). However, in a practical method, one avoids doing it.

5.3 Numerical results

In this section we describe the numerical experiments which test the performance of Algorithm 5.1 implemented in MATLAB. In particular we are interested in testing two variants of Algorithm 5.1 defined by the norm used to compute the model in (24). The first variant makes use of the ℓ_2-norm and leads to minimum Frobenius norm models. The solution of (24) with $t = 2$ is a convex quadratic problem subject to equality constraints and hence is equivalent to solving the following linear system

$$\begin{bmatrix} M(\bar{\phi}_Q, Y_k) M(\bar{\phi}_Q, Y_k)^T & M(\bar{\phi}_L, Y_k) \\ M(\bar{\phi}_L, Y_k)^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha_L \end{bmatrix} = \begin{bmatrix} f(Y_k) \\ 0 \end{bmatrix}$$

with $\alpha_Q = M(\bar{\phi}_Q, Y_k)^T \lambda$.

We solved this system using SVD, regularizing extremely small singular values after the decomposition and before performing the backward solves, in an attempt to remediate extreme ill-conditioning caused by nearly ill-poised sample sets. The second approach consisted in using $t = 1$, leading to minimum ℓ_1-norm models and attempting to recover sparsity in the Hessian of the objective function. To solve problem (24) with $t = 1$ we formulated it first as a linear program. In both cases, $t = 1, 2$, we first scaled the corresponding problems by shifting the sample set to the origin (i.e., translating all the sample points such that the current iterate coincides with the origin) and then scaling the points so that they lie in $B_2(0; 1)$ with at least one scaled point at the border of this ball. This procedure, suggested in [11, Section 6.3], leads to an improvement of the numerical results, especially in the minimum Frobenius norm case.

The trust-region subproblems (23) have been solved using the routine `trust.m` from the MATLAB Optimization Toolbox which corresponds essentially to the algorithm of Moré and Sorensen [19]. To solve the linear programs (24), with $t = 1$, we have used the routine `linprog.m` from the same MATLAB toolbox. In turn, `linprog.m` uses in most of the instances considered in our optimization runs the interior-point solver `lipsol.m`, developed by Zhang [31].

In the first set of experiments, we considered the test set of unconstrained problems from the CUTEr collection [16] used in [17], and in [14]. We used the same dimension choices as in [17] but we removed all problems considered there with less than 5 variables. This procedure resulted in the test set described in Table 1. Most of these problems exhibit some form of sparsity in the Hessian of the objective function, for instance, a banded format.

In order to present the numerical results for all problems and all methods (and variants) considered, we have used the so-called performance profiles, as suggested in [12]. Performance profiles are, essentially, plots of cumulative distribution functions $\rho(\tau)$ representing a performance ratio for the different solvers. Let $\mathcal{S}$ be the set of solvers and $\mathcal{P}$ the set of problems. Let $t_{p,s}$ denote the performance of the solver $s \in \mathcal{S}$ on the problem $p \in \mathcal{P}$ — lower values of $t_{p,s}$ indicate better performance. This performance ratio $\rho(\tau)$ is defined by first setting $r_{p,s} = t_{p,s} / \min\{t_{p,\bar{s}} : \bar{s} \in \mathcal{S}\}$, for $p \in \mathcal{P}$ and $s \in \mathcal{S}$. Then, one defines $\rho_s(\tau) = (1/|\mathcal{P}|)|\{p \in \mathcal{P} : r_{p,s} \leq \tau\}|$. Thus, $\rho_s(1)$ is the probability that solver s has the best performance among all solvers. If we are only interested in determining which solver is the most efficient (is the fastest on most problems), then we should compare the values of $\rho_s(1)$ for all the solvers. On the other hand, solvers with the largest value of $\rho_s(\tau)$ for large τ are the ones which solve the largest number of problems in $\mathcal{P}$, hence are the most robust. We are interested in considering a wide range of values for τ, hence, we plot the performance profiles in a log-scale (now, the value at 0 represents the probability of winning over the other solvers).

In our experiments, we took the *best* objective function value from [17] (obtained by applying a derivative-based Non-Linear Programming solver), as a benchmark to detect whether a problem was successfully solved up to a certain accuracy 10^{-acc}. The number $t_{p,s}$ is then the number of function evaluations needed to achieve an objective function value within an absolute error of 10^{-acc} of the best objective function value; otherwise a failure occurs and the value of $r_{p,s}$ used to build the profiles is set to a large number (see [12]). Other measures of performance could be used for $t_{p,s}$ but the number of function evaluations is the most appropriate for expensive objective functions. In Figure 1, we plot performance profiles for the two variants of Algorithm 5.1 mentioned above and for the state-of-the-art solver NEWUOA [21, 23]. Following [13], and in order to provide a fair comparison, solvers are run first with their own default stopping criterion and if convergence can not be declared another run is repeated with tighter tolerances. In the

problem	n	NNZH	DFO-TR Frob ($acc = 6$)	DFO-TR l1 ($acc = 6$)
ARGLINB	10	55	57	59
ARGLINC	8	21	56	57
ARWHEAD	15	29	195	143
BDQRTIC	10	40	276	257
BIGGS6	6	21	485	483
BROWNAL	10	55	437	454
CHNROSNB	15	29	993	1004
CRAGGLVY	10	19	548	392
DIXMAANC	15	44	330	515
DIXMAANG	15	44	395	451
DIXMAANI	15	44	429	361
DIXMAANK	15	44	727	527
DIXON3DQ	10	18	–	–
DQDRTIC	10	10	25	25
FREUROTH	10	19	249	252
GENHUMPS	5	9	1449	979
HILBERTA	10	55	8	8
MANCINO	10	55	106	73
MOREBV	10	27	111	105
OSBORNEB	11	66	1363	1023
PALMER1C	8	36	–	–
PALMER3C	8	36	56	53
PALMER5C	6	21	29	29
PALMER8C	8	36	60	55
POWER	10	55	466	428
VARDIM	10	55	502	314

Table 1: The test set used in the first set of experiments and the corresponding dimensions (first three columns). The third column reports the upper bound provided by CUTEr on the number of nonzero elements of the Hessian stored using the coordinate format. The last two columns report the total number of function evaluations required by Algorithm 5.1 to achieve an accuracy of 10^{-6} on the objective function value (versions `DFO-TR Frob` and `DFO-TR l1`). Both approaches failed to solve two of the problems.

case of Algorithm 5.1, this procedure led to $\epsilon_g = \delta = 10^{-7}$ and a maximum number of 15000 function evaluations. For NEWUOA we used the data prepared for [17] also for a maximum number of 15000 function evaluations.

Note that NEWUOA requires an interpolation of fixed cardinality in the interval $[2n + 1, (n + 1)(n + 2)/2]$ throughout the entire optimization procedure. We looked at the extreme possibilities, $2n + 1$ and $(n + 1)(n + 2)/2$, and are reporting results only with the latter one (`NEWUOA quad` in the plots) since it was the one which gave the best results. The two variants of Algorithm 5.1, are referred to as `DFO-TR Frob` (minimum Frobenius norm models) and `DFO-TR l1` (minimum ℓ_1-norm models). Two levels of accuracy (10^{-4} and 10^{-6}) are considered in Figure 1. One can observe that `DFO-TR l1` is the most efficient version ($\tau = 0$ in the log scale) and basically as robust as the `DFO-TR Frob` version (large values of τ), and that both versions of the Algorithm 5.1 seem to outperform `NEWUOA quad` in efficiency and robustness.

In the second set of experiments we ran Algorithm 5.1 for the two variants (minimum Frobenius and ℓ_1 norm models) on the test set of CUTEr unconstrained problems used in the paper [7]. These problems are known to have a significant amount of sparsity in the Hessian (this information as well as the dimensions selected is described in Table 2). We used $\epsilon_g = \delta = 10^{-5}$ and a maximum number of 5000 function evaluations. In Table 3, we report the number of objective

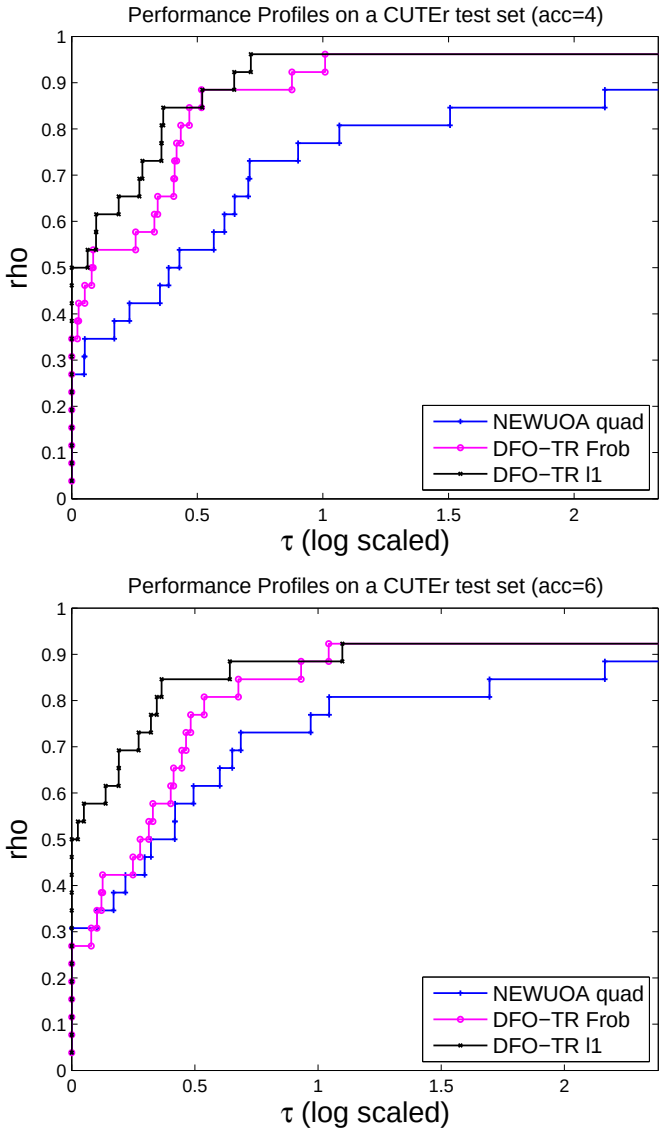

Figure 1: Performance profiles comparing Algorithm 5.1 (minimum Frobenius and ℓ_1 norm versions) and NEWUOA [21, 23], on the test set of Table 1, for two levels of accuracy (10^{-4} above and 10^{-6} below).

function evaluations taken as well as the final objective function value obtained. In terms of function evaluations, one can observe that `DFO-TR l1` wins in approximately 8/9 cases, when compared to the `DFO-TR Frob` version, suggesting that the former is more efficient than the latter in the presence of Hessian sparsity. Another interesting aspect of the `DFO-TR l1` version is some apparent ability to produce the final model with gradient of smaller norm.

problem	n	type of sparsity	NNZH
ARWHEAD	20	sparse	39
BDQRTIC	20	banded	90
CHNROSNB	20	banded	39
CRAGGLVY	22	banded	43
DQDRTIC	20	banded	20
EXTROSNB	20	sparse	39
GENHUMPS	20	sparse	39
LIARWHD	20	sparse	39
MOREBV	20	banded	57
POWELLSG	20	sparse	40
SCHMVETT	20	banded	57
SROSENBR	20	banded	30
WOODS	20	sparse	35

Table 2: The test set used in the second set of experiments. For each problem we included the number of variables and the type of sparsity, as described in [7]. The last column reports the upper bound provided by CUTEr on the number of nonzero elements of the Hessian stored using the coordinate format.

6 Conclusion

Since compressed sensing emerged, it has been deeply connected to optimization, using optimization as a fundamental tool (in particular, to solve ℓ_1-minimization problems). In this paper, however, we have shown that compressed sensing methodology can also serve as a powerful tool for optimization, in particular for Derivative-Free Optimization (DFO), where structure recovery can improve the performance of optimization methods. Namely, our goal was to construct fully quadratic models (essentially models with an accuracy as good as second order Taylor models; see Definition 2.2) of a function with sparse Hessian using underdetermined quadratic interpolation on a sample set with potentially much fewer than $\mathcal{O}(n^2)$ points. We were able to achieve this as is shown in Theorem 4.5, by considering an appropriate polynomial basis and random sample sets of only $\mathcal{O}(n(\log n)^4)$ points when the number of non-zero components of the Hessian is $\mathcal{O}(n)$. The corresponding quadratic interpolation models were built by minimizing the ℓ_1-norm of the entries of the Hessian model. We then tested the new model selection approach in a deterministic setting, by using the minimum ℓ_1-norm quadratic models in a practical interpolation-based trust-region method (see Algorithm 5.1). Our algorithm was able to outperform state-of-the-art DFO methods as shown in the numerical experiments reported in Section 5.3.

One possible way of solving the ℓ_1-minimization problem (4) in the context of interpolation-based trust-region methods is to rewrite it as a linear program. This approach was used to numerically test Algorithm 5.1 when solving problems (24) for $t = 1$. For problems of up to $n = 20, 30$ variables, this way of solving the ℓ_1-minimization problems has produced excellent results in terms of the derivative-free solution of the original minimization problems (22) and is reasonable in terms of the overall CPU time.

However, for larger values of n, the repeated solution of the linear programs introduces significant overhead. Besides the increase in the dimension, one also has to consider possible ill-conditioning arising due to badly poised sample sets. Although related linear programming problems are solved in consecutive iterations, it is not trivial to use warmstart. In fact, the number of rows in the linear programs change frequently, making it difficult to warmstart simplex-

problem	DFO-TR Frob/l1	# f eval	final f value	final ∇m norm
ARWHEAD	Frob	338	3.044e-07	3.627e-03
ARWHEAD	l1	218	9.168e-11	7.651e-07
BDQRTIC	Frob	794	5.832e+01	5.419e+05
BDQRTIC	l1	528	5.832e+01	6.770e-02
CHNROSNB	Frob	2772	3.660e-03	2.025e+03
CHNROSNB	l1	2438	2.888e-03	1.505e-01
CRAGGLVY	Frob	1673	5.911e+00	1.693e+05
CRAGGLVY	l1	958	5.910e+00	8.422e-01
DQDRTIC	Frob	72	8.709e-11	6.300e+05
DQDRTIC	l1	45	8.693e-13	1.926e-06
EXTROSNB	Frob	1068	6.465e-02	3.886e+02
EXTROSNB	l1	2070	1.003e-02	6.750e-02
GENHUMPS	Frob	5000	4.534e+05	7.166e+02
GENHUMPS	l1	5000	3.454e+05	3.883e+02
LIARWHD	Frob	905	1.112e-12	9.716e-06
LIARWHD	l1	744	4.445e-08	2.008e-02
MOREBV	Frob	539	1.856e-04	2.456e-03
MOREBV	l1	522	1.441e-04	3.226e-03
POWELLSG	Frob	1493	1.616e-03	2.717e+01
POWELLSG	l1	5000	1.733e-04	2.103e-01
SCHMVETT	Frob	506	-5.400e+01	1.016e-02
SCHMVETT	l1	434	-5.400e+01	7.561e-03
SROSENBR	Frob	456	2.157e-03	4.857e-02
SROSENBR	l1	297	1.168e-02	3.144e-01
WOODS	Frob	5000	1.902e-01	8.296e-01
WOODS	l1	5000	1.165e+01	1.118e+01

Table 3: Results obtained by DFO-TR Frob and DFO-TR l1 on the problems of Table 2 (number of evaluations of the objective function, final value of the objective function, and the norm of the final model gradient).

based methods. An alternative is to attempt to approximately solve problem (4) by solving $\min \|M(\bar{\phi}, Y)\alpha - f(Y)\|_2 + \tau\|\alpha_Q\|_1$ for appropriate values of $\tau > 0$. We conducted preliminary testing along this avenue but did not succeed in outperforming the linear programming approach in any respect. However, it is out of the scope of this paper a deeper study of the numerical solution of the ℓ_1-minimization problem (4) in the context of interpolation-based trust-region methods.

Although we only considered the most common type of sparsity in unconstrained optimization (sparsity in the Hessian), it is straightforward to adapt our methodology to the case where sparsity also appears in the gradient. In particular, if we aim at only recovering a sparse gradient or a sparse fully linear model, one can show that the required number of sample points to do so would be less that $\mathcal{O}(n)$ and tighten to the level of sparsity.

Finally, we would like to stress that building accurate quadratic models for functions with sparse Hessians from function samples could be of interest outside the field of Optimization. The techniques and theory developed in Section 4 could also be applicable in other settings of Approximation Theory and Numerical Analysis.

Acknowledgments

We would like to thank Rachel Ward (Courant Institute of Mathematical Sciences, NYU) for interesting discussions on compressed sensing and to Anke Tröltzsch (CERFACS, Toulouse) for providing us assistance with the testing environment of Section 5.3. We are also grateful to the

anonymous referees for their helpful comments on the earlier version of the manuscript.

References

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Program.*, 95:3–51, 2003.

[2] A. S. Bandeira, M. Fickus, D. G. Mixon, and P. Wong. The road to deterministic matrices with the restricted isometry property. *Submitted, available online: arXiv:1202.1234 [math.FA]*, 2012.

[3] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. On partially sparse recovery. Technical Report 11-13, Dept. Mathematics, Univ. Coimbra, 2011.

[4] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx.*, 28:253–263, 2008.

[5] E. Candès and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inform. Theory*, 52:5406–5425, 2006.

[6] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, 589–592, 2009.

[7] B. Colson and Ph. L. Toint. Optimizing partially separable functions without derivatives. *Optim. Methods Softw.*, 20:493–508, 2005.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

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

[10] 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 J. Optim.*, 20:387–415, 2009.

[11] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[12] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[13] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM J. Optim.*, 16:891–909, 2006.

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

[15] M. C. Ferris and G. Deng. Classification-based global search: An application to a simulation for breast cancer. In *Proceedings of 2008 NSF Engineering Research and Innovation Conference*, Knoxville, Tennessee, 2008.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29:373–394, 2003.

[17] S. Gratton, Ph. L. Toint, and A. Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optim. Methods Softw.*, 26:873–894, 2011.

[18] L. Jacques. A short note on compressed sensing with partially known signal support. *Signal Processing*, 90:3308–3312, 2010.

[19] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Comput.*, 4:553–572, 1983.

[20] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, second edition, 2006.

[21] M. J. D. Powell. On trust region methods for unconstrained minimization without derivatives. *Math. Program.*, 97:605–623, 2003.

[22] M. J. D. Powell. Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Math. Program.*, 100:183–215, 2004.

[23] M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Nonconvex Optim. Appl.*, volume 83, pages 255–297. Springer-Verlag, Berlin, 2006.

[24] H. Rauhut. Compressive sensing and structured random matrices. In M. Fornasier, editor, *Theoretical Foundations and Numerical Methods for Sparse Recovery*, Radon Series Comp. Appl. Math., pages 1–92. 2010.

[25] H. Rauhut and R. Ward. Sparse Legendre expansions via ℓ_1-minimization. *J. Approx. Theory*, 2012, to appear.

[26] K. Scheinberg and Ph. L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM J. Optim.*, 20:3512–3532, 2010.

[27] T. Tao. Open question: Deterministic UUP matrices: `http://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices`, 2007.

[28] N. Vaswani and W. Lu. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Trans. Signal Process.*, 58:4595–4607, 2010.

[29] S. M. Wild. MNH: A derivative-free optimization algorithm using minimal norm Hessians. In *Tenth Copper Mountain Conference on Iterative Methods*, April 2008.

[30] P. Wojtaszczyk. Stability and instance optimality for gaussian measurements in compressed sensing. *Found. Comput. Math.*, 10:1–13, 2010.

[31] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. *Optim. Methods Softw.*, 10:1–31, 1998.