ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Obtaining Quadratic Models of Noisy Functions[1]

Aswin Kannan and Stefan M. Wild

Mathematics and Computer Science Division

Preprint ANL/MCS-P1975-1111.

September 2012

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. This work was supported by the Advanced Scientific Computing Research Program, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Obtaining Quadratic Models of Noisy Functions[1]

Aswin Kannan and Stefan M. Wild

Abstract

When derivatives of a nonlinear objective function are unavailable, many derivative-free optimization algorithms rely on interpolation-based models of the function. But what if the function values are contaminated by noise, as in most of the simulation-based problems typically encountered in this area? We propose to obtain linear and quadratic models by using knowledge of the level of noise in a function. We develop an efficient algorithm for obtaining the model coefficients, and we analyze the properties of the corresponding quadratic program.

1 Introduction and Motivation

A common theme [3,4,15–17] in derivative-free optimization is the use of a quadratic model

$$q(x) = c + g^T x + \frac{1}{2} x^T H x \tag{1}$$

to approximate a function whose derivatives are unavailable or intractable to approximate directly. The $p_n := \frac{(n+1)(n+2)}{2}$ model coefficients $(c, g, H = H^T)$ are typically obtained by requiring that the quadratic satisfy interpolation conditions of the form

$$q(x_i) = f_i \qquad i = 1, \ldots, m, \tag{2}$$

where the function values $f_1, \ldots, f_m$ are known at the m points in $\mathcal{X} = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$.

In practice, the number of available function values m is often strictly less than p_n. In this case, methods (such as [4,17]) often resolve the remaining degrees of freedom in (2) by minimizing the Frobenius norm of the Hessian,

$$\min_{c,g,H=H^T} \left\{ \frac{1}{2}\|H\|_F^2 = \frac{1}{2} \sum_{i,j=1}^{n} H_{i,j}^2 : q(x_i) = f_i,\, i = 1, \ldots, m \right\}. \tag{3}$$

Previous works (see [3, 17]) provide conditions on $\mathcal{X}$ so that unique solutions to (3) are obtained for arbitrary values of $f \in \mathbb{R}^m$.

In practice, however, objectives with unavailable derivatives are typically based on deterministic simulations that rely on adaptive and/or iterative finite-precision computations,

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. This work was supported by the Advanced Scientific Computing Research Program, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

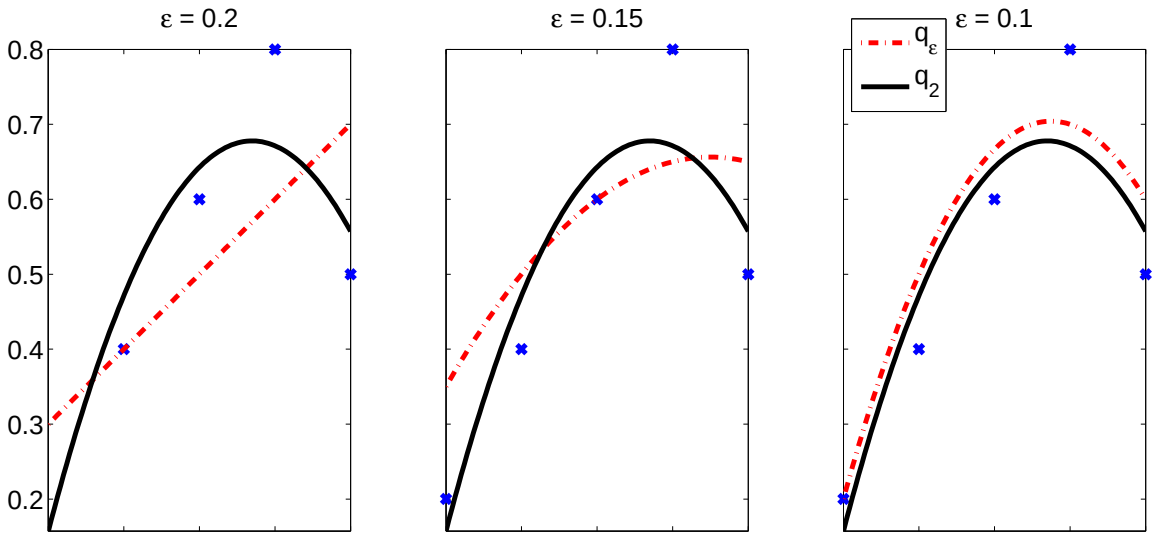

Figure 1: Least-squares quadratic q_2 from (5) and the proposed quadratic q_ϵ from (4) for three values of ϵ.

which result in computational noise contaminating the function values obtained. Noise can be the result of other approximations or errors made in computing the function values. This noise can destroy smoothness in the underlying process being simulated and, in some cases, can hamper the progress of methods employing interpolation-based models.

In this paper we address this noise directly by relaxing the interpolation conditions (2). Our approach is to enlarge the feasible region in (3) to account for absolute errors of size ϵ. For notational ease, we begin by assuming ϵ is a scalar uniform bound, but more general $\epsilon \in \mathbb{R}^m_+$ are straightforward.

We obtain the quadratic coefficients by solving the parametric convex quadratic program

$$\mathcal{P}(\mathcal{X}, f, \epsilon) \equiv \min_{c,g,H=H^T} \left\{ \frac{1}{2}\|H\|_F^2 : |q_\epsilon(x_i) - f_i| \leq \epsilon,\ i = 1, \ldots, m \right\}. \tag{4}$$

A primary benefit of this approach is that it incorporates knowledge of the magnitude of the noise. The value of ϵ can reflect a bound on the absolute errors, a regularization tolerance that depends on the size of a neighborhood of interest, or can be a small factor times the noise level (standard deviations) computed in [12].

Alternatively, provided that $m > p_n$, one can follow the approach of [2] and determine the least-squares solution to (2), which may no longer correspond to a quadratic that interpolates the data. This approach seeks the coefficients that solve the optimization problem

$$\min_{c,g,H=H^T} \left\{ \sum_{i=1}^{m} (q_2(x_i) - f_i)^2 \right\}. \tag{5}$$

The solid line in Figure 1 illustrates the least-squares quadratic, q_2, along with our proposed quadratic, q_ϵ, for three values of ϵ on a univariate example. Here we see that

when the noise is known to be large ($\epsilon = 0.2$), the least-squares quadratic can be viewed as an overfit of the data since a linear function would suffice as an approximation. As the noise tolerance decreases, the quadratic q_ϵ becomes more nonlinear in order to better fit the data. For small enough ϵ, we obtain a certificate that no quadratic passes within ϵ of the given function values.

This simple example motivates the use of an absolute bound ϵ when forming an approximation model for optimization, sensitivity analysis, or other applications. In the remainder of the paper we focus on analysis and techniques for solving the corresponding quadratic program (4). In Section 2 we introduce a reformulation of the problem and provide local approximation bounds on the resulting quadratic. Further analysis of the problem (4) is provided in Section 3, including results on monotonicity of the solution and uniqueness of the solution in special cases. We introduce a specialized active set method in Section 4 and examine the performance of a `Matlab` implementation of the method on a series of test problems in Section 5 to illustrate its robustness in practice. In Section 6 we summarize the main contributions of this paper.

2 Preliminaries

The interpolation conditions (2) can be represented as a linear system of the form

$$\Phi z = [\Phi_l \quad \Phi_h] \begin{bmatrix} z_l \\ z_h \end{bmatrix} = f, \tag{6}$$

where $z \in \mathbb{R}^{p_n}$ corresponds to a vector of the distinct coefficients ($c, g, H = H^T$) and $\Phi \in \mathbb{R}^{m \times p_n}$ is a particular quadratic basis evaluated at the m points in $\mathcal{X}$, which can be suitably partitioned into affine and nonlinear parts, denoted by the subscripts l and h, respectively. One can easily show (see, e.g., [3]) that coefficients satisfying (2) exist if and only if (6) has a solution. Results can be extended to the case when f is arbitrary or when unique solutions are desired, by requiring that $\text{rank}(\Phi) = m$ and $\text{rank}(\Phi) = p_n$, respectively.

For notational convenience we will work with a specific basis defining the ith rows of Φ_l and Φ_h to be given by $[1,\, x_i^T]$ and

$$\left[\frac{1}{2}\mathcal{X}_{1,i}^2,\, \cdots,\, \frac{1}{2}\mathcal{X}_{n,i}^2,\, \frac{1}{\sqrt{2}}\mathcal{X}_{1,i}\mathcal{X}_{2,i},\, \cdots,\, \frac{1}{\sqrt{2}}\mathcal{X}_{n-1,i}\mathcal{X}_{n,i}\right],$$

respectively, where $\mathcal{X}_{j,i}$ denotes the jth component of x_i. This basis means that $\|H\|_F = \|z_h\|_2$, so that (4) is equivalent to

$$\min_z \left\{ \frac{1}{2} z_h^T z_h : -\epsilon \leq \Phi z - f \leq \epsilon \right\}. \tag{7}$$

When ϵ becomes sufficiently large, it is clear that $H = 0$ and there may be several solutions $(c, g, 0)$ to (4). For this reason, given a particular $(\mathcal{X}, f)$, we define

$$\overline{\epsilon} = \overline{\epsilon}(\mathcal{X}, f) = \min_{\epsilon, c, g} \left\{ \epsilon : |c + g^T x_i - f_i| \leq \epsilon,\, i = 1, \ldots, m \right\} \tag{8}$$

to be the first ϵ for which a linear model can be fit to $(\mathcal{X}, f)$.

Likewise, when ϵ becomes sufficiently small, the feasible region of (4) may become empty. Hence, we define

$$\underline{\epsilon} = \underline{\epsilon}(\mathcal{X}, f) = \min_{\epsilon,c,g,H=H^T} \left\{ \epsilon : |c + g^T x_i + \frac{1}{2} x_i^T H x_i - f_i| \leq \epsilon, \, i = 1, \ldots, m \right\} \tag{9}$$

to be the smallest ϵ for which a quadratic interpolating ϵ-perturbed function values exists. One can view the quadratic q_∞ resulting from (9) as an l_∞-norm regression.

An easy consequence of these definitions, Proposition 2.1 shows cases of particular interest: when $\epsilon \in [\underline{\epsilon}, \overline{\epsilon}]$ exist.

Proposition 2.1. *For all finite data $(\mathcal{X}, f)$,*

$$0 \leq \underline{\epsilon} \leq \overline{\epsilon} < \infty. \tag{10}$$

Furthermore, if $\Phi z = f$ has no solution, then $\underline{\epsilon} > 0$.

Proof. The three inequalities in (10) are an immediate result of the absolute value, $H = 0$ being a special case of (9), and by setting $(c, g, H) = 0$ and $\epsilon = \|f\|_\infty$ in (9), respectively. The second result is an immediate consequence of $\epsilon = 0$ being infeasible for the definition in (9). □

Before proceeding, we comment on our particular form of the objective (4). Models that minimize a measure of curvature play a large role in the context of approximation by splines but are also increasingly common in optimization. For example, in model-based derivative-free optimization, the curvature of an interpolating model directly appears in the bounds for the zero- and first-order approximation errors of both underdetermined quadratic [3,17] and radial basis function [18] models. Recent work [1] has also shown that pursuing other metrics of a quadratic model's Hessian can result in second-order approximations of special classes of functions.

Three typical errors of interest are

$$e^f(x) = f(x) - f_s(x), \tag{11}$$

$$e^q(x) = q(x) - f_s(x), \tag{12}$$

$$e^g(x) = \nabla q(x) - \nabla f_s(x) = Hx + g - \nabla f_s(x), \tag{13}$$

where f_s represents the true, underlying smooth function and f is the observed (assumed deterministic) function. When function values are observed with error ($e^f(x) \neq 0$), the typical *fully linear* (*quadratic*) bounds described in [3] no longer apply. The following theorem, proved in the appendix, illustrates how these errors can enter into approximation bounds on some smooth (but unknown) function f_s.

Theorem 2.2. *Suppose that f_s is continuously differentiable, that ∇f_s is γ_f-Lipschitz continuous in $\mathcal{B}_0 = \{x : \|x - x_{i_0}\| \leq \Delta\}$ with $\Delta > 0$, and that $\mathcal{X} \subset \mathcal{B}_0$ contains at least $n + 1$ affinely independent points so that*

$$Y = \frac{1}{\Delta} \begin{bmatrix} x_{i_1} - x_{i_0} & \cdots & x_{i_n} - x_{i_0} \end{bmatrix}$$

is nonsingular. Define $\epsilon^f := \epsilon + \max_{i_0,\ldots,i_n} |e^f(x_i)|$. *Then, if a quadratic* q *satisfies* $|q(x_i) - f(x_i)| \leq \epsilon$ *for all* $x_i \in \mathcal{X}$, *the following inequalities hold for any* $x \in \mathcal{B}_0$*:*

$$|e^q(x)| \leq \Delta^2 (\gamma_f + \|H\|_F) \frac{5\sqrt{n}\|Y^{-1}\| + 1}{2} + \epsilon^f \left(2\sqrt{n}\|Y^{-1}\| + 1\right), \tag{14}$$

$$\|e^g(x)\| \leq \|Y^{-1}\|\sqrt{n} \left(\frac{5}{2}\Delta (\gamma_f + \|H\|_F) + \frac{2}{\Delta}\epsilon^f\right). \tag{15}$$

By themselves (e.g., when $\epsilon^f = 0$), the first terms of the right-hand sides (14) and (15) provide the first-order Taylor-like error bounds pursued in [3], namely,

$$|e^q(x)| \leq \kappa_0 \Delta^2, \qquad \|e^g(x)\| \leq \kappa_1 \Delta,$$

for all $x \in \mathcal{B}_0$, with constants κ_0, κ_1 independent of both x and Δ. If the errors ϵ^f are of order Δ^2, these bounds can again be recovered. In the general setting, however, we note that the presence of errors bounded away from zero results in the divergence of the gradient error bound in (15) as Δ is reduced.

3 Properties

In this section we analyze the parameterized quadratic program (4) (and its equivalent form (7)) and its dependence on the inputs $(\mathcal{X}, f, \epsilon)$. The following proposition discusses a special case when Φ_l is of full rank.

Proposition 3.1. *If* $m > p_n - (n+1)$ *and* Φ_l *is not of full column rank, then* $\Phi = [\Phi_l, \Phi_h]$ *is not of full row rank.*

Proof. Since Φ_l is not of full column rank, let $\rho \in \mathbb{R}^{n+1}$ be a nontrivial dependence relation on the columns of Φ_l:

$$\rho_1 \mathcal{X}_{1,i} + \rho_2 \mathcal{X}_{2,i} + \cdots + \rho_n \mathcal{X}_{n,i} + \rho_{n+1} = 0, \quad i = 1, \ldots, m.$$

Multiplying by $\mathcal{X}_{j,i}$, for $j = 1, \ldots, n$, we obtain

$$2\rho_j \Big(\underbrace{\frac{1}{2}\mathcal{X}_{j,i}^2}_{j^{\text{th}} \text{ col. of } \Phi_h} \Big) = -\sum_{k \neq j} 2^{1/2} \rho_k \Big(\underbrace{2^{-1/2}\mathcal{X}_{j,i}\mathcal{X}_{k,i}}_{\text{other cols. of } \Phi_h} \Big) - \rho_{n+1} \Big(\underbrace{\mathcal{X}_{j,i}}_{j^{\text{th}} \text{ col. of } \Phi_l} \Big), \quad i = 1, \ldots, m. \tag{16}$$

Let $\mathcal{S} = \{j \leq n : \rho_j \neq 0\}$ and $\mathcal{S}^C = \{j \leq n : \rho_j = 0\}$ represent a partitioning of $\{1, \ldots, n\}$. Then, for $j \in \mathcal{S}$, (16) shows that the jth column of Φ_h is linearly dependent on Φ_l and Φ_h (but not the first n columns of Φ_h), leading to a total of $|\mathcal{S}|$ dependent columns. For $j \in \mathcal{S}^C$, the dependence relation can be written as

$$\sum_{k \in \mathcal{S}} \rho_k \mathcal{X}_{j,i} \mathcal{X}_{k,i} = -\rho_{n+1} \mathcal{X}_{j,i}, \quad i = 1, \ldots, m, \tag{17}$$

where the columns in the left-hand side are distinct for different $j \in \mathcal{S}^C$ and do not include the columns on the left-hand side of (16). Therefore there exist at least $|\mathcal{S}| + |\mathcal{S}^C| = n$ columns in Φ_h that are linearly dependent on Φ_l and the other columns of Φ_h. Since $\text{rank}(\Phi_l) < n+1$, there exists at least one column in Φ_l that is linearly dependent on the remaining columns in Φ_l, and hence $\text{rank}(\Phi) \leq p_n - (n+1) < m$. □

Although it is useful primarily when $m \in (p_n - (n+1), p_n]$, Proposition 3.1 shows that, for these values of m, Φ_l is of full rank whenever Φ is. The algorithm presented in the next section requires knowledge of the rank of submatrices of Φ_l at every iteration, and hence Proposition 3.1 can be used to partially eliminate the need for performing an actual rank check. This is especially beneficial when $m \geq p_n$ is large.

The following proposition characterizes the relationship between the objective in the parameterized program (7) and ϵ.

Proposition 3.2. *Let z_h^ϵ be a solution to (7) for $\epsilon \geq 0$. Then,*

$$\|z_h^\epsilon\| \leq (1-\alpha^\epsilon)\|z_h^0\|, \quad \textit{where } \alpha^\epsilon = \min\left(\frac{\epsilon}{\|f\|_\infty}, 1\right). \tag{18}$$

Moreover, if $\epsilon_u > 0$ and $\|z_h^{\epsilon_u}\| \neq 0$, then for $\epsilon > \epsilon_u$, $\|z_h^\epsilon\| < \|z_h^{\epsilon_u}\|$.

Proof. Since z_h^0 solves (7) for $\epsilon = 0$, there exists z_l^0 satisfying $\Phi_h z_h^0 + \Phi_l z_l^0 = f$. Let $z = (1-\alpha)z^0$, where $z^0 = [z_h^0; z_l^0]$ and $\alpha \geq 0$ is a constant. Then

$$\|\Phi z - f\|_\infty = \|(1-\alpha)\Phi z^0 - f\|_\infty = \alpha\|f\|_\infty.$$

Thus, for the $\alpha = \alpha^\epsilon \leq 1$ defined in (18), $\|\Phi z - f\|_\infty \leq \epsilon$, and hence

$$\|z_h^0\| \geq (1-\alpha^\epsilon)\|z_h^0\| \geq \|z_h^\epsilon\|,$$

where the last inequality holds because $(1-\alpha^\epsilon)z_h^0$ and z_h^ϵ are feasible and optimal for (7), respectively.

For the second claim, let $\epsilon_u > 0$, and take $z = (1-\alpha)z^{\epsilon_u}$ for $\alpha > 0$ and $z_h^{\epsilon_u} \neq 0$. Then

$$\|\Phi z - f\|_\infty = \|(1-\alpha)\Phi z^{\epsilon_u} - f\|_\infty \leq \|\Phi z^{\epsilon_u} - f\|_\infty + \alpha\|\Phi z^{\epsilon_u}\|_\infty \leq \epsilon_u + \alpha(\|f\|_\infty + \epsilon_u).$$

For $\epsilon > \epsilon_u$ and $\alpha = \alpha^u = \min\left(\frac{\epsilon-\epsilon_u}{\epsilon_u+\|f\|_\infty}, 1\right)$, it follows that $\|\Phi z - f\|_\infty \leq \epsilon$. Thus when $\epsilon > \epsilon_u$, we have that $\alpha^u > 0$ and hence $\|z_h^\epsilon\| \leq (1-\alpha^u)\|z_h^{\epsilon_u}\| < \|z_h^{\epsilon_u}\|$. □

Proposition 3.2 shows that the optimal objective, provided it is nonzero, is strictly monotone in ϵ and that the optimal $\|z_h\|$ decreases at least linearly in ϵ. The next result identifies the key properties of the active set that we will exploit in our algorithm to obtain the quadratic q_ϵ.

Proposition 3.3. *If*

(A) Φ is such that any subset of $n+1$ rows of Φ_l are full rank,

then the following hold:

1. *Let Φ_l^* denote the rows of Φ_l corresponding to an active set at optimality of* (8)*. If $\overline{\epsilon} > 0$, then rank$(\Phi_l^*) = n+1$, and there are more than $n+1$ rows in Φ_l^*.*

2. *If $\overline{\epsilon} > 0$, then* (8) *has a unique solution.*

3. *Let Φ^* denote the rows of Φ corresponding to an active set at optimality of* (7)*. If Φ is of full row rank and $\epsilon < \overline{\epsilon}$, then rank$(\Phi^*) > n+1$.*

4. *If Φ is of full row rank and $\epsilon < \overline{\epsilon}$, then* (7) *has a unique solution.*

Proof. 1. For an optimal solution $(\overline{\epsilon} > 0, z_l^*)$ to (8) with an active set corresponding to Φ_l^*, we have $\Phi_l^* z_l^* = f^* + \overline{\epsilon} d^*$, where f^* represents the corresponding right-hand side vector and d^* is a vector with entries ± 1.

Suppose that there are $n+1$ or fewer rows in Φ_l^*. Then, by (A) the rows of Φ_l^* are linearly independent, and hence there exists $\hat{z}_l$ such that $\Phi_l^* \hat{z}_l = f^*$. Thus

$$\Phi_l^* \left(z_l^* + \alpha(\hat{z}_l - z_l^*) \right) = f^* + (1-\alpha)\overline{\epsilon} d^*,$$

for any $\alpha \in (0,1)$. Since $\overline{\epsilon} > 0$, the active constraints are now active for a strictly smaller $\epsilon = (1-\alpha)\overline{\epsilon} < \overline{\epsilon}$. For α sufficiently small, the inactive constraints remain inactive with $(1-\alpha)\overline{\epsilon}$, and this situation contradicts the optimality of $(\overline{\epsilon}, z_l^*)$. Hence there are at least $n+2$ rows in Φ_l^* and by (A) $n+1$ are linearly independent.

2. Suppose that there exist two solutions $z_l^0 \neq z_l^1$ to (8). By convexity of (8), $z_l^\alpha = (1-\alpha) z_l^0 + \alpha z_l^1$, $0 \leq \alpha \leq 1$, represents a family of optimal solutions (infinitely many solutions) to (8). Let Φ_l^* refer to one optimal active set. From the fact that rank$(\Phi_l^*) = n+1$, the optimal solution z_l^* corresponding to that active set is unique. From the finiteness of m, the number of different active sets is finite, a contradiction to the family of solutions. The claim follows.

3. Given $\epsilon < \overline{\epsilon}$, for an optimal solution (z_h^*, z_l^*) to (7) with an active set corresponding to Φ^*, we have $\Phi^* z^* = f^* + \epsilon d^*$, where f^* represents the corresponding right-hand side vector and d^* is a vector with entries ± 1. Suppose that there are $n+1$ or fewer rows in Φ^*. Then, by (A) the rows of Φ_l^* are linearly independent, and hence there exists $\bar{z}_l$ such that $\Phi_l^* \bar{z}_l = \Phi^* z^*$. Thus for $\bar{z} = \begin{pmatrix} 0 \\ \bar{z}_l \end{pmatrix}$,

$$\Phi^* \left(z^* + \alpha(\bar{z} - z^*) \right) = \Phi^* z^* = f^* + \epsilon d^*,$$

for any $\alpha \in (0,1)$. This shows that the active constraints are now active for a strictly smaller $\|\bar{z}_h\| = (1-\alpha)\|z_h^*\|$. For α sufficiently small, the inactive constraints remain inactive, and this situation contradicts the optimality of (z_h^*, z_l^*). Hence there are at least $n+2$ rows in Φ^*.

4. Suppose that there exist two solutions $z^0 \neq z^1$ to (7). By convexity of (7), $z^\alpha = (1-\alpha)z^0 + \alpha z^1$, $0 \leq \alpha \leq 1$, represents a family of optimal solutions (infinitely many solutions) to (7). Let Φ^* refer to one optimal active set. From the full rank of Φ and from (A), the optimal solution z^* is unique to a given Φ^* (proved in Lemma 4.1). The claim follows from a similar contradiction with respect to the finiteness of the number of active sets, as stated in part (2) of the proposition.

□

If points in $\mathcal{X}$ are generated uniformly at random, then with probability 1 Φ is of full rank and assumption (A) holds. We are interested in the general case of scattered data $\mathcal{X}$, and hence assumption (A) will not be required to hold for the full matrix Φ. Instead, the algorithm in the next section will operate with submatrices corresponding to a working set.

4 Algorithms: Interpolation and Noise

The previous section focused on theoretical properties. We now turn to algorithms for solving (7). The diagonal structure of the objective, double-sided constraints, possibilities of small values of ϵ, and ill-conditioned basis matrices are notable properties of (7) that we address below.

Use of an interior-point scheme entails the use of slack variables, increasing the overall problem dimension by at least $2m$. Motivated by the desire to solve overdetermined ($m > p_n$) as well as underdetermined cases ($m \leq p_n$) and to potentially not worsen the conditioning of the present system, we employ an active-set scheme [9, 14] specialized to suit our problem's structure. An active set refers to the set of equality constraints and the inequality constraints that are tight. The heart of an active set scheme lies in solving an equality-constrained quadratic problem at every iteration. Throughout this paper, we treat cases where $\epsilon \in \mathbb{R}^m_+$ is strictly positive. We also assume throughout that the problem has a feasible solution, which can be detected by a linear feasibility test corresponding to (9).

Our methodology is summarized as Algorithm 1. From the positivity of ϵ, it is clear that for any double-sided constraint, only one side can be active. Our algorithm uses this feature and treats every double-sided constraint as a "single" constraint rather than as two separate constraints.

For ease of notation, we drop the explicit iteration counter k dependence from the basis matrices Φ. We take Φ_a to be the matrix of active constraints at iteration k:

$$\Phi_a = \begin{pmatrix} \Phi_w \\ \Phi_d \end{pmatrix}, \quad \begin{pmatrix} \Phi_w \\ \Phi_d \end{pmatrix} z^k = \begin{pmatrix} f_w + D_w \epsilon_w \\ f_d + D_d \epsilon_d \end{pmatrix}, \Phi_w = \begin{pmatrix} \Phi_{wh} & \Phi_{wl} \end{pmatrix},$$

where Φ_w and Φ_d represent the linearly independent ("working set") and dependent rows of Φ_a, respectively. We introduce a diagonal matrix $D = \begin{pmatrix} D_w & 0 \\ 0 & D_d \end{pmatrix}$ with entries ± 1 to handle the double sidedness of the constraints. Note that D refers to D^k with the iteration suffixes dropped for notational convenience. Also note that the entry in D corresponding to a constraint could be of different signs for different iteration suffixes k.

Algorithm 1 Modified active-set scheme

Choose tolerance levels $\psi^p > 0$ and $\phi^\lambda > 0$; set $\psi > \psi^p$, $\phi > \phi^\lambda$, and $\mathcal{T} = \{1, \cdots, m\}$; and set $k = 0$.
Find a feasible solution z^0 to (7) and compute an initial active set $\mathcal{A}^0$.
while $\psi > \psi^p$ and $\phi > \phi^\lambda$ **do**
  Partition $\mathcal{A}^k$ into $\mathcal{W}^k$ and $\mathcal{D}^k$ (sets of linearly independent and dependent constraints, respectively).
  Set $\mathcal{I}^k = \mathcal{T}^k - \{\mathcal{A}^k\}$, $n_w = |\mathcal{W}^k|$, and $n_d = |\mathcal{D}^k|$.
  Solve the active-set subproblem (19) to compute the direction p^k and multipliers λ^k.
  Set $\psi = \|p_k\|$.
  if $\psi > \psi^p$ **then**
    Set, $\alpha^k = \min_{i \in \mathcal{I}^k \cup \mathcal{D}^k} \left\{ \max \left(\frac{f_i - \Phi_i z^k - \epsilon_i}{\Phi_i p^k}, \frac{f_i - \Phi_i z^k + \epsilon_i}{\Phi_i p^k} \right) \right\}$ to ensure that $z^k + \alpha^k p^k$ is feasible.
    if $\alpha^k > 1$ **then**
      No blocking constraint is encountered; set $\alpha^k = 1$.
    else
      Add the blocking constraint to $\mathcal{W}^k$, breaking ties lexicographically.
      Modify $\mathcal{A}^k$ and $\mathcal{D}^k$ depending on whether the blocking constraint is from $\mathcal{I}^k$ or $\mathcal{D}^k$.
    end if
  else
    Set $\phi = \min_i(\lambda_i^k)$ and $\alpha_k = 0$.
    if $\phi < -\phi^\lambda$ **then**
      if $n_d > 0$ **then**
        Drop (from $\mathcal{A}^k$ and $\mathcal{W}^k$) the negative multiplier in λ^k with the least index.
      else
        Drop (from $\mathcal{A}^k$ and $\mathcal{W}^k$) the most negative multiplier in λ^k.
      end if
    else
      Optimality within tolerance is attained; exit.
    end if
  end if
  Set $z^{k+1} = z^k + \alpha^k p^k$.
  Update $k \leftarrow k + 1$.
end while

It suffices to solve the active-set subproblem with Φ_w instead of Φ_a,

$$\min_{p_h,p_l}\left\{\frac{1}{2}p_h^T p_h + p_h^T z_h : \Phi_{wh}p_h + \Phi_{wl}p_l = 0\right\}, \tag{19}$$

since $\Phi_w p = 0$ implies that $\Phi_d p = 0$. Letting $\hat{\Phi}_{wl}$ and $\tilde{\Phi}_{wl}$ represent the linearly independent and dependent columns of Φ_{wl}, respectively, we have that

$$\Phi_{wl} = \begin{pmatrix}\hat{\Phi}_{wl} & \tilde{\Phi}_{wl}\end{pmatrix} P, \quad \tilde{\Phi}_{wl} = \hat{\Phi}_{wl}\sigma, \quad p_l = P^T\begin{pmatrix}\hat{p}_l \\ \tilde{p}_l\end{pmatrix}, \tag{20}$$

where P is a permutation matrix. Setting $\tilde{p}_l = 0$, we have that $\Phi_{wl}p_l = \hat{\Phi}_{wl}\hat{p}_l$, and hence (19) can be written as

$$\min_{p_h,\hat{p}_l}\left\{\frac{1}{2}p_h^T p_h + p_h^T z_h : \Phi_{wh}p_h + \hat{\Phi}_{wl}\hat{p}_l = 0\right\}. \tag{21}$$

Let λ represent the vector of multipliers corresponding to the equality constraints in (21). The KKT necessary optimality conditions for the subproblem (21) are sufficient for optimality due to convexity. These conditions can be compactly written as

$$\begin{pmatrix}\Phi_{wh}\Phi_{wh}^T & \hat{\Phi}_{wl} \\ \hat{\Phi}_{wl}^T & 0\end{pmatrix}\begin{pmatrix}\lambda \\ \hat{p}_l\end{pmatrix} = \begin{pmatrix}\Phi_{wh}z_h \\ 0\end{pmatrix}, \tag{22}$$

or, equivalently,

$$\begin{aligned}&(Z^T\Phi_{wh}\Phi_{wh}^T Z)\lambda_z = Z^T\Phi_{wh}z_h \\ &Y^T\hat{\Phi}_{wl}\hat{p}_l = Y^T\Phi_{wh}z_h^k - Y^T\Phi_{wh}\Phi_{wh}^T\lambda \\ &\lambda = Z\lambda_z,\end{aligned}$$

where Z refers to a basis for the null space of $\hat{\Phi}_{wl}^T$ and Y is a basis for the range space of $\hat{\Phi}_{wl}$. Y and Z can be obtained, for example, by a QR factorization of $\hat{\Phi}_{wl}$.

4.1 Solving the active-set subproblem

Based on the previous discussion, solving the subproblem (21) is equivalent to solving the linear system (22). Lemma 4.1 states when this linear system yields a unique solution.

Lemma 4.1. *If the working set matrix Φ_w is of full row rank, then the following hold:*

1. $\begin{pmatrix}\Phi_{wh} & \hat{\Phi}_{wl}\end{pmatrix}$ *is of full row rank;*
2. $Z^T\Phi_{wh}\Phi_{wh}^T Z$ *is positive definite and (21) has a unique solution; and*
3. $M = \begin{pmatrix}\Phi_{wh}\Phi_{wh}^T & \hat{\Phi}_{wl} \\ \hat{\Phi}_{wl}^T & 0\end{pmatrix}$ *is nonsingular.*

Proof. Dropping linearly dependent columns maintains the matrix rank, and hence the first claim follows. The other results are standard (see, for example, [14, Chap. 16] and [17]) and rely on showing that $\Phi_{wh}^T Z$ is of full column rank. □

The matrix denoted by M exploits the sparsity structure of the objective and solves a small linear system. However, when the number of active constraints is close to p_n or when $p_n - m$ is small, we resort to solving the following system instead:

$$Z_\Phi^T Q Z_\Phi t = Z_\Phi^T Q z, \quad p = Z_\Phi t, \quad Q = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix},$$

where Z_Φ denotes the null space matrix of Φ and the size of Z_Φ would at most be $p_n \times (p_n - m)$.

4.2 Adding and deleting constraints

This section focuses on lines 13-14 (addition) and 19-23 (deletion) of Algorithm 1. Once (19) is solved, we backtrack along the obtained direction p^k to maintain feasibility. We let α^k refer to the backtracking parameter (line 9 in Algorithm 1) and let Φ_b refer to the blocking constraint. If Φ_b is linearly independent of Φ_w, it is added to the working set ($\alpha^k > 0$); otherwise it is added to the dependent active set ($\alpha^k = 0$). In case of a tie, the least index is added to the working set.

If $p^k = 0$, then progress cannot be made with the current working set. In this case (as is done in a standard active-set scheme), we drop a constraint with a negative multiplier. If there is no negative multiplier, the current solution and working set are optimal. In cases of negative multipliers, Lemma 4.2 proves that descent is obtained in the subsequent iterations.

Lemma 4.2. *Let Φ_w^k refer to the matrix of the working set at iteration k, and let $(\Phi_w^k) = \begin{pmatrix} \Phi_c \\ \Phi_e \end{pmatrix}$, where Φ_e is the constraint dropped (due to a negative multiplier) at the end of the iteration ($\Phi_w^{k+1} = \Phi_c$). Then p_h^{k+1} is nonzero and is a descent direction for (21).*

Proof. Let λ and μ be the respective multipliers for Φ_c and Φ_e in the kth iteration, and let $\bar{\lambda}$ be the multiplier of Φ_c in the $(k+1)$st iteration. Let D_e be the scalar ± 1 associated with constraint e. Then, from

$$\begin{pmatrix} z_h^k \\ 0 \end{pmatrix} = -\Phi_c^T D_c^T \lambda - \Phi_e^T D_e \mu, \quad \begin{pmatrix} z_h^k + p_h^{k+1} \\ 0 \end{pmatrix} = -\Phi_c^T D_c^T \bar{\lambda},$$

we have that

$$\begin{pmatrix} p_h^{k+1} \\ 0 \end{pmatrix} = -\Phi_c^T D_c^T (\bar{\lambda} - \lambda) + \Phi_e^T D_e \mu. \tag{23}$$

Since $\mu < 0$ and Φ_c and Φ_e are linearly independent, $p_h^{k+1} \neq 0$. Taking the inner product of (23) and p^{k+1} and noting that $\Phi_c p^{k+1} = 0$, we have that $0 < \|p_h^{k+1}\|^2 = \Phi_e p^{k+1}(D_e \mu)$.

If $D_e\mu > 0$, then $\Phi_e p^{k+1} > 0$, while if $D_e\mu < 0$, then $\Phi_e p^{k+1} < 0$; both cases implying that feasibility is maintained with respect to the current working set. From $p_h^{k+1} \neq 0$ and from the uniqueness of p_h^{k+1}, $\|z_h^k + p_h^{k+1}\|^2 < \|z_h^k\|^2$, imply that $\|p_h^{k+1}\|^2 + (z_h^k)^T p_h^{k+1} < 0$. Therefore,

$$\alpha^2\|p_h^{k+1}\|^2 + \alpha(z_h^k)^T p_h^{k+1} \leq \alpha\|p_h^{k+1}\|^2 + \alpha(z_h^k)^T p_h^{k+1} < 0,$$

for all $0 < \alpha \leq 1$, implying that p_h^{k+1} is a descent direction for (21). □

When the active set has no dependent constraints, we drop the constraint with the most negative multiplier. If the active set has dependent constraints, then we employ Bland's lexicographic rule (see, e.g., [10]) to prevent cycling (dropping the least index from the negative multipliers). Moreover, in this case, we also backtrack on the set of active dependent constraints in addition to the inactive indices in the subsequent iterations. At any iteration, if z^k is not optimal, descent is guaranteed after a finite number of iterations. Given a working set, z_h^k is unique; since there are a finite number of working sets, this procedure has finite termination.

4.3 Special case of Φ

An important feature of Algorithm 1 is that it operates for any general case, with no restrictions on $\mathcal{X}$ and f. In contrast, uniqueness of standard linear regression and interpolation models heavily rely on linear independence assumptions (e.g., that Φ_l is of full rank) that restrict the $\mathcal{X}$ that can be considered. Moreover, if the basis Φ is of full rank, it eliminates possibilities of degeneracy and thereby results in considerably easier problems.

This motivates a more restrictive case of Φ given by the following assumption.

Assumption 4.3. *Φ is of full row rank, and Φ_l is of full column rank.*

Notably, in a pure interpolation problem (e.g., (7) with $\epsilon = 0$), Assumption 4.3 guarantees uniqueness of the resulting quadratic model [17]. With purely theoretical interest, we now propose a minor modification to Algorithm 1 when a working set does not satisfy Assumption 4.3.

The following lemma states that Assumption 4.3 holds on the active-set matrix corresponding to one optimal solution.

Lemma 4.4. *Let Assumption 4.3 hold. Let Φ^* be the matrix corresponding to the active set of a solution z^* to (7) and suppose rank$(\Phi_l^*) < n+1$. Then, for fixed z_h^*, there exists a $\tilde{z}_l$ and a larger active set with matrix $\tilde{\Phi} = \begin{pmatrix} \Phi^* \\ \Phi_\nu \end{pmatrix}$ such that rank$(\tilde{\Phi}_l) = n+1$.*

Proof. Let $\mathcal{N}(\tilde{\Phi}_l^*)$ denote the null space of $\tilde{\Phi}_l^*$, and let the inactive constraints be represented by $\mathcal{I}$. For all $j \in \mathcal{I}$, take

$$a_j = \frac{f_j - \epsilon_j - \Phi_j z^*}{\max_{p\in\mathcal{N}(\tilde{\Phi}_l^*)} \Phi_j p}, \quad b_j = \frac{f_j + \epsilon_j - \Phi_j z^*}{\max_{p\in\mathcal{N}(\tilde{\Phi}_l^*)} \Phi_j p}, \qquad t = \min_{j\in\mathcal{I}}(\max(a_j, b_j)). \tag{24}$$

Since $\mathcal{N}(\tilde{\Phi}_l^*) \neq \{0\}$, we have that $t > 0$ is finite. Constraints Φ_ν thus can be iteratively added to the working set until the set of rows $\Phi_{\nu l}$ satisfies $\mathcal{N}(\tilde{\Phi}_l^*) \cap \mathcal{N}(\Phi_{\nu l}) = \{0\}$, or equivalently, $\mathcal{N}(\tilde{\Phi}_l) = \{0\}$. □

With the above motivation, we modify the active-set scheme slightly and start with a working set with $\text{rank}(\Phi_{wl}) = n + 1$. In any subsequent iteration, when a constraint is dropped that reduces the rank of Φ_{wl}, the above procedure of adding a new constraint is employed. Theorem 4.5 establishes finite termination and convergence of this scheme.

Theorem 4.5. *Let Assumption 4.3 hold, and let $\mathcal{I}$ denote the set of indices of the inactive constraints at iteration k. Suppose the working active set represented by $\Phi_w^k = \begin{pmatrix} \Phi_c \\ \Phi_e \end{pmatrix}$ has rank$(\Phi_{wl}^k) = n + 1$, with Φ_e denoting the constraint dropped from the working set (because of a negative multiplier) such that rank$(\Phi_{cl}) < n + 1$. Then, the following hold:*

1. *There exists a $\nu \in \mathcal{I}$ such that $\Phi_{wl}^{k+1} = \begin{pmatrix} \Phi_{cl} \\ \Phi_{\nu l} \end{pmatrix}$ is of full column rank.*
2. *If $\Phi_\nu \neq \Phi_e$, then $p_h^{k+1} \neq 0$ is a descent direction.*
3. *If $\Phi_\nu = \Phi_e$, then descent is seen or optimality is reached after a finite number of iterations.*

Proof. The first claim follows from Lemma 4.4. Let λ and μ be the Lagrange multipliers of Φ_c and Φ_e, respectively, in the kth iteration, and let $\bar{\lambda}$ and $\bar{\mu}$ be the Lagrange multipliers of Φ_c and Φ_ν in the $(k+1)$st iteration. Then,

$$-\Phi_c^T D_c^T(\bar{\lambda} - \lambda) - \Phi_\nu^T D_\nu \bar{\mu} + \Phi_e^T D_e \mu = \begin{pmatrix} p_h^{k+1} \\ 0 \end{pmatrix}. \tag{25}$$

For the second claim, $\Phi_\nu \neq \Phi_e$ implies that Φ_c, Φ_ν, and Φ_e are linearly independent and $D_\nu \neq -D_e$.

Therefore $\bar{\mu} = \mu = 0$, is a contradiction, since $\mu < 0$. Therefore, $\|p_h^{k+1}\| \neq 0$. Taking an inner product with p^{k+1}, we get $0 < \|p_h^{k+1}\|^2 = \Phi_e p^{k+1}(D_e \mu)$. If $D_e \mu > 0$, then $\Phi_e p > 0$; and if $D_e \mu < 0$, then $\Phi_e p < 0$, ensuring feasibility. From the uniqueness of $p_h^{k+1} \neq 0$, and along similar lines to Lemma 4.2, one can easily see that p^{k+1} is a descent direction.

For the third result, $\Phi_\nu = \Phi_e$ implies that $D_\nu = -D_e$. It suffices to analyze the case $\|p_h^{k+1}\| = 0$. Considering (25), $\bar{\mu} = -\mu > 0$ and $\bar{\lambda} = \lambda$. Therefore, the constraint Φ_c cannot be dropped. If $\bar{\lambda} > 0$, then optimality is reached. With negative indices, the same procedure can be repeatedly applied until all multipliers are positive or descent is obtained. The worst-case number of iterations is the number of indices (finite) with negative multipliers at the beginning of iteration k. □

Convergence and finite termination follow. Note that if $\text{rank}(\Phi_l) = s < n + 1$, then a column reduction can be done initially ($n+1$ columns to s columns), and the same algorithm can be applied.

4.4 Bound-constrained formulation

When Assumption 4.3 holds, a bound-constrained version of (7) can be posed, with the bounds parameterized by ϵ. In the next section we show that this formulation is useful in practice for certain cases of (7). By introducing additional variables y, (7) can be written as

$$\min_{-\epsilon \le y \le \epsilon} \left\{ \min_{z} \left\{ \frac{1}{2} z_h^T z_h : \Phi z - f = y \right\} \right\}, \tag{26}$$

for some inputs $(\mathcal{X}, f, \epsilon)$. The inner problem is an equality-constrained quadratic program, with KKT conditions represented by linear equations of the form

$$\begin{pmatrix} \lambda \\ z_l \end{pmatrix} = G \begin{pmatrix} f + y \\ 0 \end{pmatrix}, \; G = \begin{pmatrix} \Phi_h \Phi_h^T & \Phi_l \\ \Phi_l^T & 0 \end{pmatrix}^{-1} = \begin{pmatrix} G_\lambda & G_b \\ G_b^T & G_d \end{pmatrix}, \; z_h = \Phi_h^T \lambda.$$

This inversion is possible only if Assumption 4.3 holds. In this case, (26) becomes

$$\min_{-\epsilon \le y \le \epsilon} \left\{ \frac{1}{2} y^T M y + f^T M y \right\}, \tag{27}$$

where $M = G_\lambda^T \Phi_h \Phi_h^T G_\lambda$. We refer to the above as the *bound-constrained formulation* of (7) and note the reduction in problem dimension ($m \le p_n$).

5 Numerical Experiments

We developed a `Matlab` implementation of Algorithm 1, which does not require Assumption 4.3. A linear feasibility solver was also implemented to compute an initial feasible point. The linear feasibility phase also returns a corresponding initial basis matrix Φ that is scaled and preconditioned. The feasibility solver solves (7) approximately and then warm starts the active-set method from a feasible point close to the approximate solution. The entire package is referred to as `noqs` (NOisy Quadratic Solver). The numerical experiments presented in this section were implemented in `Matlab` 7.11.0584 (R2010b) on a Linux operating system with a quad-core 2.664 GHz processor and 4 GB of memory.

In this section, we compare the performance of `noqs` and three other `Matlab`-based solvers for solving (7): `minq` [13], `quadprog` from the `Matlab` Optimization Toolbox [11], and `qppal` [8] using filters. The solver `minq` makes use of the bound-constrained formulation discussed in the previous section and thus requires Assumption 4.3. The solver `quadprog` can be used without this assumption for two different formulations,

$$\begin{array}{lll} \min\limits_{y,z} & \frac{1}{2} z_h^T z_h & \min\limits_{z} \quad \frac{1}{2} z_h^T z_h \\ \text{subject to} & \Phi z = y + f & \text{subject to} \quad \Phi z \le f + \epsilon \\ & -\epsilon \le y \le \epsilon, & \quad\quad\;\; -\Phi z \le -f + \epsilon. \end{array} \tag{28}$$

The first formulation, which is also used by `qppal`, introduces additional variables y while the second uses one-sided constraints. In our tests, we observed that `quadprog` performed

much better using the second formulation; hence this is the version of quadprog we consider in all results that follow. We note that quadprog handles the inequality constraints explicitly as one-sided constraints.

We also tested warm-start versions of minq, quadprog, and qppal, whereby each solver was given the initial z (and the corresponding preconditioned Φ) returned by the linear feasibility routine of noqs. The suffix "ws" associated with solvers other than noqs denotes this warm start. Since the time required for the linear feasibility phase is charged to these solvers, we observed that minq-ws and qppal-ws were overall slightly slower than their non-warm-started versions. Consequently, these two warm-start versions are not included in the results presented in the sequel.

5.1 Test problems

The test set used for comparing the solvers consisted of two classes of problems: ones that satisfy Assumption 4.3 (and thus can be solved by minq) and ones that do not, denoted by $\mathcal{P}_m$ and $\mathcal{P}_r$, respectively. The problems are summarized in Table 1. For each subset shown in the table, problems were generated for eleven values of n, three noise levels, and three values of m, leading to a subset total of 99.

In practice, we expect the allowable perturbation ϵ to model both absolute noise, $\epsilon = \epsilon_a e$, and relative noise, $\epsilon = \epsilon_r |f|$, where ϵ_a and ϵ_r are positive scalars and e denotes a vector of ones. Our test includes three levels of noise (values of ϵ_a and ϵ_r), $\{10^{-5}, 10^{-3}, 10^{-1}\}$. In the case of relative noise, however, so that the alternative algorithms would be given as favorable inputs as possible, we ensure that the corresponding absolute perturbations are not too small and therefore set $\epsilon = \max\left\{\epsilon_r |f|, 5 \cdot 10^{-8}\right\}$, with the max taken componentwise.

For both ill-conditioned and nicely conditioned $\mathcal{X}$, function values f were obtained by both random number generators and by evaluating noisy quadratics. For underdetermined cases, overdetermined cases, and degenerate cases in $\mathcal{P}_r$, n was varied in $\{10, \cdots, 40\}$, $\{10, \cdots, 30\}$, and $\{10, \cdots, 20\}$, respectively. These ranges are indicative of the typical dimensions seen in derivative-free optimization. For each underdetermined case, three values of $m \in [4n, 6n, p_n]$ were considered. For each overdetermined case, m took three values $(\lceil 1.2p_n \rceil, \lceil 1.5p_n \rceil, 2p_n)$. The size of the resulting quadratic programs ranged from 66 variables and 40 constraints to 861 variables and 992 constraints. We note that the constraint matrices Φ are dense and that all of the problems are feasible by construction. The test problems and the noqs solver are available at `http://www.mcs.anl.gov/~wild/noqs/`.

5.2 Stationarity and feasibility metrics

Computation time ("time" refers to CPU time throughout this section) and measures of stationarity and infeasibility are the metrics we used to compare the performance of all the solvers. Prior to defining these metrics, we restate a measure from [6] for computing the

Table 1: Characterization of the test problem sets $\mathcal{P}_m$ and $\mathcal{P}_r$.

			Nicely cond. $\mathcal{X}$	Ill cond. $\mathcal{X}$
$\mathcal{P}_m$	Underdetermined ($m \leq p_n$)	Absolute noise	198	99
		Relative noise	0	99
$\mathcal{P}_r$	Overdetermined ($m > p_n$)	Absolute noise	99	99
		Relative noise	0	99
	Degenerate Φ^* and $m \leq p_n$	Absolute noise	99	0
	Rank-deficient Φ and $m \leq p_n$	Absolute noise	99	0

relative distance δ, between two scalars:

$$\delta[a, b] = \min\left\{|a - b|, \frac{|a - b|}{|a| + |b|}\right\},$$

with $\delta(0, 0) = 0$ and satisfying $0 \leq \delta \leq 1$. Extending this to vectors $x, y \in \mathbb{R}^p$, we define the distance $d \in \mathbb{R}^p$ by

$$d_i[x, y] = \delta[x_i, y_i], \qquad i = 1, \ldots, p,$$

with corresponding distance metric $\|d[x, y]\|$.

We denote the solution from solver s on problem $p \in \mathcal{P}_m \cup \mathcal{P}_r$ by $z^{s,p}$. The solvers noqs, qppal, and quadprog also return corresponding multipliers. Although the formulations used by these solvers are different, the multipliers $\lambda^{s,p}$ corresponding to the formulation (7) can be obtained from elementary rearrangement operations.

The solver minq does not return the multipliers with its solution $y^{s,p}$. Given $y^{s,p}$, however, one can estimate the corresponding $\lambda^{s,p}$ by solving

$$\min_{\lambda}\left\{\|My^{s,p} + Mf^p - \lambda\|^2 : 0 \leq -y_i^{s,p}\lambda_i \perp (\epsilon_i^p)^2 - (y_i^{s,p})^2 \geq 0, i = 1, \ldots, m\right\}. \tag{29}$$

If minq returns a solution $y^{s,p}$ that is optimal, then one can show that the multipliers obtained by solving (29) are equal to the optimal multipliers obtained by solving (7). This entails proving that $\Phi^T M \Phi = I$, which follows from Assumption 4.3 and basic linear algebra. The solution to (29) can be further simplified as follows:

$$\lambda_i^{s,p} = \begin{cases} \max\left(M_i(y^{s,p} + f^p), 0\right) & \text{if } y_i^{s,p} = -\epsilon_i^p \\ \min\left(M_i(y^{s,p} + f^p), 0\right) & \text{if } y_i^{s,p} = \epsilon_i^p \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

for $i = 1, \ldots, m$.

The problem of ultimate interest (7) is characterized by double-sided constraints with a narrow band of feasibility as result of small ϵ. For instance, consider the case of $z^{s,p}$ being infeasible by an order of ϵ. When ϵ is small, the absolute error is negligible but the relative error is high. To make sure that our feasibility and stationarity metrics are scale-invariant, we normalize the constraints by ϵ and define our measures accordingly. We employ a feasibility metric v^f defined by

$$v_i^{f,s,p} = \begin{cases} 0 & \text{if } -1 \leq \frac{\Phi_i^p z^{s,p} - f_i^p}{\epsilon_i^p} \leq 1 \\ \min\left(\delta\left[\frac{\Phi_i^p z^{s,p} - f_i^p}{\epsilon_i^p}, -1\right], \delta\left[\frac{\Phi_i^p z^{s,p} - f_i^p}{\epsilon_i^p}, 1\right]\right) & \text{otherwise,} \end{cases}$$

where we note that the above normalization can also account for differences in sign. The metric for optimality, v^o, is given by the measure of stationarity

$$v^{o,s,p} = d\left[\begin{pmatrix} z_h^{s,p} \\ 0 \end{pmatrix}, (\Phi^p)^T \lambda^{s,p}\right].$$

For given tolerance levels τ^f and τ^o, solver s is considered to have solved problem p if

$$\left\|v^{f,s,p}\right\| \leq \tau^f \quad \text{and} \quad \|v^{o,s,p}\| \leq \tau^o. \tag{31}$$

5.3 Solver termination and benchmarking

Using a solver's default termination conditions may result in suboptimality or delayed termination with a sufficiently optimal solution. For example, solvers may have different internal termination criteria and preset tight tolerances. A common criterion used by each tested solver is the maximum number of outer iterations allowed. For quadprog and noqs, this refers to the number of active set iterations; for qppal and minq, this refers to the sequences of bound-constrained quadratic subproblems and the subspace minimization problems, respectively. However, running each solver for progressively increasing values of these outer iteration numbers and computing feasibility and optimality metrics at regular intervals would be an extremely time-consuming way to check for consistent termination. On the other hand, running the solvers progressively by using the respective previous solutions can propagate infeasible solutions in an adverse way.

Therefore, for this study we ran each solver for a large number of maximum outer iterations and recorded the history of the solution iterates, multipliers, and computational time at intermediate intervals. We computed the metrics of optimality and feasibility at these intervals. If termination criteria were satisfied at these intermediate intervals, then the process of checking was stopped and the solver time set to the corresponding computational time from the history. The maximum number of active-set iterations was set to 10,000 for quadprog and noqs, and the maximum number of major iterations and subspace steps for qppal and minq were set to be 500 and $5m$, respectively.

The process is summarized by Algorithm 2. Abusing notation, z^k and λ^k refer to the solution iterates and multipliers at the k^{th} history point.

With the computational time $t^{s,p}$, the performance profile [6] for solver s is given by

$$\rho^s(\gamma) = \frac{1}{|\mathcal{P}|}\left|\left\{p \in \mathcal{P} : \frac{t^{s,p}}{\min_s t^{s,p}} \leq \gamma\right\}\right|,$$

for $\gamma \geq 1$. We define the feasibility profile of solver s by

$$\omega^s(\gamma) = \frac{1}{|\mathcal{P}|}\left|\left\{p \in \mathcal{P} : \|v^{f,s,p}\| \leq \gamma\right\}\right|,$$

for $\gamma \geq 0$. Both the performance profile $\rho^s(\gamma)$ and feasibility profile $\omega^s(\gamma)$ can be interpreted as cumulative distribution functions.

Algorithm 2 Solver termination

Run solver s on problem p by fixing the maximum outer iterations to k_{curr}; initialize $t^{s,p} = \infty$
Set feasibility and optimality termination criteria $\tau^f > 0$ and $\tau^o > 0$, respectively
Set termination flag $\theta = 0$, $k = 1$ and k_h to be equal to the total number of recorded history points
while $k \leq k_h$ and $\theta = 0$ **do**
  Compute $v^{f,s,p}$ and $v^{o,s,p}$ from z^k and λ^k
  if $\|v^{f,s,p}\| \leq \tau^f$ and $\|v^{o,s,p}\| \leq \tau^o$ **then**
    Set $\theta = 1$
    Set $t^{s,p}$ to be equal to the computational time taken by the solver at the k^{th} history point
  end if
  Set $k = k + 1$
end while

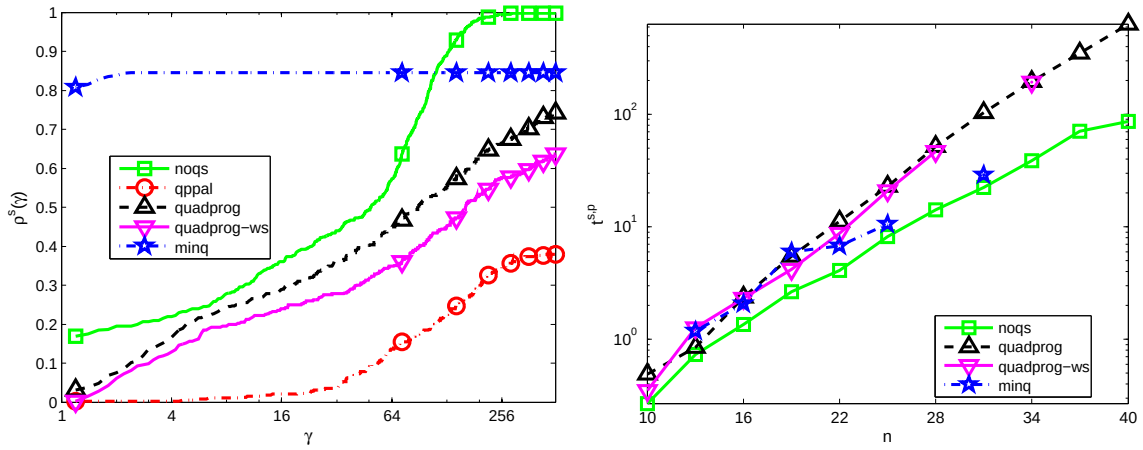

Figure 2: (Left) Performance profiles $\rho_s(\gamma)$ of computation time on problem set $\mathcal{P} = \mathcal{P}_m$. (Right) Computation times (log-scale) for the 11 problems in $\mathcal{P}_m$ with $\epsilon_a = 10^{-1}, m = p_n$, and ill-conditioned $\mathcal{X}$.

5.4 Underdetermined problems

We first consider $\mathcal{P} = \mathcal{P}_m$, corresponding to the subset of test problems that can be formulated as bound-constrained quadratic programs (BCQPs). Figure 2 (left) shows the performance profiles for time using the tolerance levels $\tau^f = \tau^o = 10^{-4}$.

We observed that qppal takes comparatively longer to obtain feasible solutions when the number of constraints (m) increases or when ϵ becomes small. Even when qppal begins with a feasible point (through the noqs warm start), subsequent iterates tend to get infeasible while minimizing the objective. This situation is to be expected because qppal is not specially designed for the feasibility regions that are thin (as a result of double-sided constraints with tight bounds) and defined by relatively large numbers of constraints.

We attribute minq's excellent performance primarily to the use of the BCQP formulation, with all variables y bounded by comparatively smaller values. In this case, the problem is scaled such that the bounds are normalized, and thus the Hessian matrix M is effectively

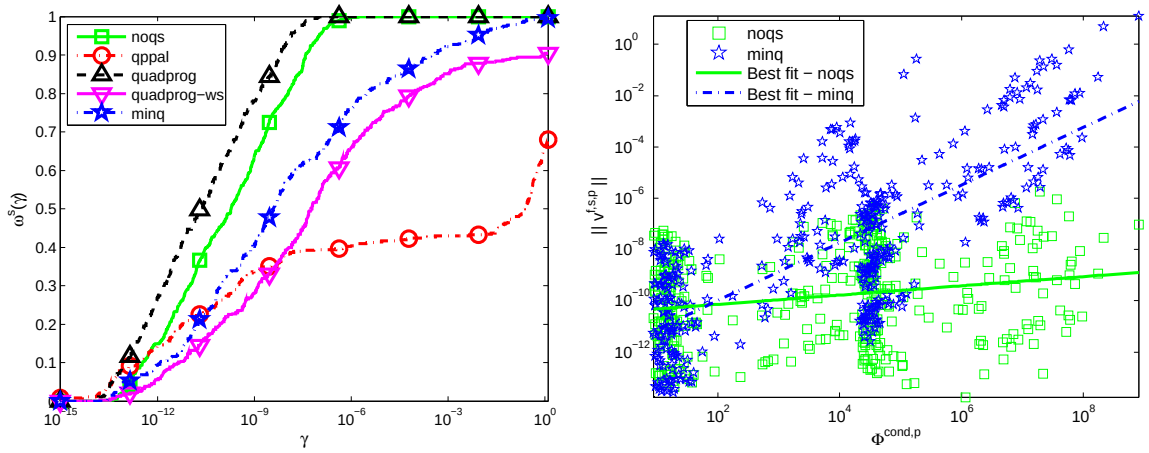

Figure 3: Infeasibility profiles $\omega_s(\gamma)$ on problem set $\mathcal{P} = \mathcal{P}_m$ (left); infeasibility versus condition number of Φ (right).

replaced by $\epsilon_a M$. When ϵ_a is small, the quadratic terms do not dominate, thereby making gradient projections highly effective. For larger values of ϵ_a, however, noqs often performs better than minq. This situation is illustrated in Figure 2 (right), which shows the computation times on the subset of 11 problems in $\mathcal{P}_m$ with absolute noise $\epsilon_a = 10^{-1}, m = p_n$, and ill-conditioned $\mathcal{X}$. One can also see that quadprog is slow in comparison with minq and noqs.

Figure 3 (left) shows the infeasibility profiles, and Figure 3 (right) shows the variation of the measure of infeasibility, v^f, with respect to the condition number of Φ for minq and noqs. The infeasibilities encountered by minq can be attributed to the inversion of the matrix G (required by the BCQP formulation) and the conditioning of M. The inversion of G is a saddle-point problem and can be solved by using preconditioners [7]. We leave the issue of obtaining greater levels of accuracy in the BCQP formulation (e.g., by using extended precision or converting to a better-conditioned quadratic basis) as future work.

5.5 Overdetermined problems

We now turn to the overdetermined cases, degenerate underdetermined cases, and rank-deficient underdetermined cases that cannot be handled by minq. This set $\mathcal{P} = \mathcal{P}_r$ contains the 495 problems shown in Table 1. In a few such cases (when ϵ was small), the linear feasibility solver in noqs had difficulty finding an initial feasible point, marked by larger values of the scale-invariant infeasibility metric

$$\delta^{s,p} = \max_{1 \le i \le m} \left(\frac{|\Phi_i^p z^{s,p} - f_i^p| - \epsilon_i^p}{\epsilon_i^p} \right). \tag{32}$$

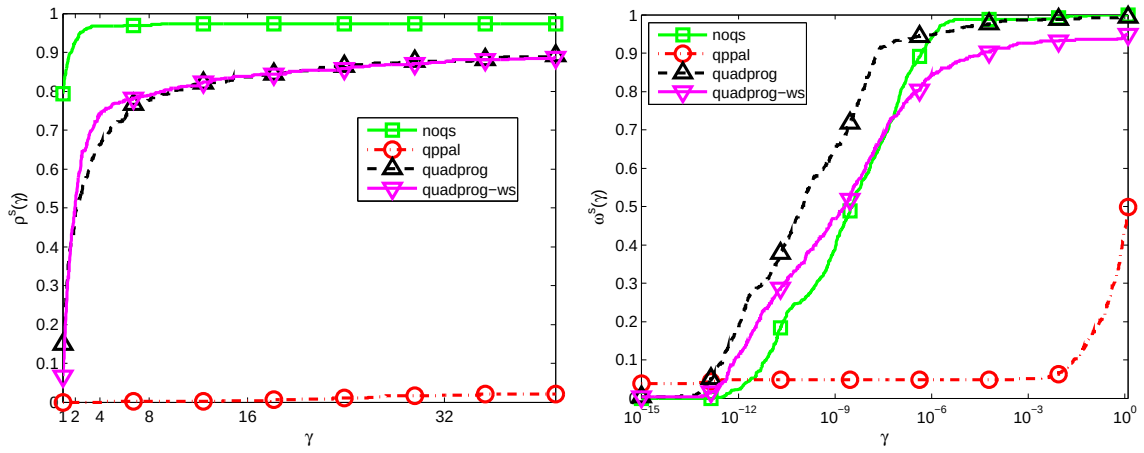

Figure 4: Performance profiles $\rho_s(\gamma)$ of time (left) and infeasibility profiles $\omega_s(\gamma)$ (right) on problem set $\mathcal{P} = \mathcal{P}_r$ (left).

In such cases, we instead ran the active-set method (and all other solvers) on a relaxed problem with $\epsilon^{new} = (1 + r_{\text{tol}})\epsilon$, where

$$r_{\text{tol}} = 10^{-5}, \qquad \text{if} \quad \delta^{s,p} \geq 10^{-5}.$$

This is expected to happen for $p \in \mathcal{P}_r$, when values of ϵ are very small.

Figure 4 (left) and (right) compare the time and infeasibility levels, respectively, of the three applicable solvers for $\mathcal{P}_r$. We observed that most of the cases where quadprog (without the warmstart) outperformed noqs were ones where n was small ($n = 10$ or $n = 12$). This is partly attributed to the fact that the linear feasibility phase consumed a major portion of the time required by noqs for smaller values of n (discussed further below and in Figure 6).

Figure 5 (left) shows the variation of time for noqs and quadprog for n varying from 10 to 50 in steps of 4 (problems generated in addition to $\mathcal{P}_r$) for $m = \lceil 1.5p_n \rceil$ for well-conditioned $\mathcal{X}$ (not degenerate or rank-deficient) and $\epsilon = 10^{-3}e$. Figure 5 (right) shows the variation of time for noqs and quadprog for n varying from 10 to 40 for $m = p_n$ for rank-deficient Φ and $\epsilon = 10^{-3}e$. One can deduce that warm starting quadprog tends to result in a reduction in the total time.

Figure 6 shows the percentage of time spent on the linear feasibility phase and the active-set phase by noqs for the same set of problems shown in Figure 5 (left). One can see that the warm-starting or feasibility phase consumes a significant amount of time for noqs when n is small, while the active-set phase dominates for sufficiently large n.

Overall, noqs is seen to be robust with respect to all problem dimensions (both p_n and m), all values of ϵ, ill-conditioning, degeneracy, and rank deficiency. This robustness is primarily attributed to exploitation of the problem structure in terms of convexity, sparsity, and double-sided constraints, solving the appropriate KKT system, parameterization of the constraints in terms of ϵ (not $f \pm \epsilon$), use of matrix factorizations and updates only when required, and use of tie-breaking rules for degeneracy only when required.

n	NoQS	quadprog	quadprog-ws
10	1.09e+00	4.70e-01	1.21e+00
14	2.19e+00	2.02e+00	2.53e+00
18	5.59e+00	7.17e+00	6.19e+00
22	1.49e+01	1.80e+01	1.48e+01
26	3.64e+01	5.04e+01	4.35e+01
30	8.05e+01	1.31e+02	1.13e+02
34	1.72e+02	3.42e+02	2.65e+02
38	3.48e+02	7.58e+02	5.56e+02
42	8.92e+02	1.54e+03	1.21e+03
46	2.10e+03	2.96e+03	2.21e+03
50	3.05e+03	5.34e+03	3.87e+03

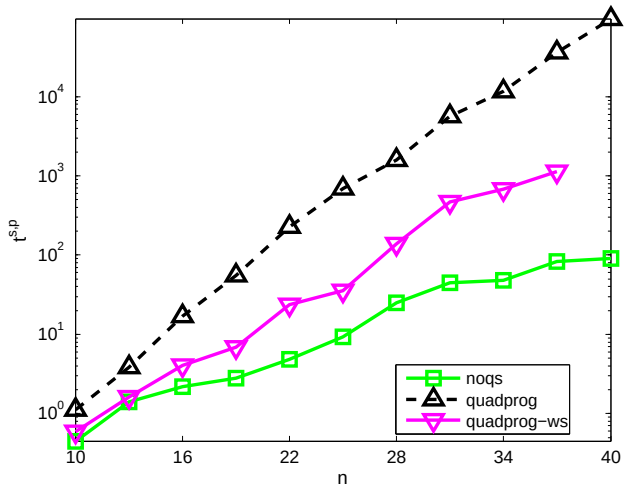

Figure 5: Computational time (in seconds) for overdetermined problems (left) and scalability for rank-deficient underdetermined cases (right).

Since the problem is parameterized with respect to ϵ, our view is that feasibility may be a primary requirement for many applications. With respect to feasibility, noqs is seen to be close to that of quadprog, which requires longer running times. We conclude this section by recalling that minq can serve as an alternative to noqs only in underdetermined cases, under Assumption 4.3, in the presence of nicely conditioned systems, and when values of ϵ are not comparatively large.

6 Concluding Remarks

In this paper we have proposed a framework for obtaining surrogate quadratic models of simulation-based functions that are corrupted by noise, the level of which is known. Our approach requires solving a quadratic program with a diagonal, convex objective and narrow, double-sided linear inequality constraints that are parameterized by the noise parameter. The primary focus of this work is on posing and solving this quadratic program. The contributions are on three fronts:

- **Theoretical properties.** The problem was parameterized in terms of the basis matrix and the noise. Fundamental properties of the basis matrix were studied and the behavior of the objective function in terms of the noise levels was analyzed. Under stronger assumptions, uniqueness was established. Bounds on the error between the surrogate model and the underlying noise-free function were derived.

- **Algorithms.** An active-set scheme was tailored to the quadratic program's specific structure. The algorithm resolves semi-definiteness using basic matrix factorizations, maintains factorizations only when required, makes use of the problem structure efficiently, and tackles degeneracy. Under stronger assumptions, the active-set scheme was further modified to obtain stronger results and an equivalent, bound-constrained

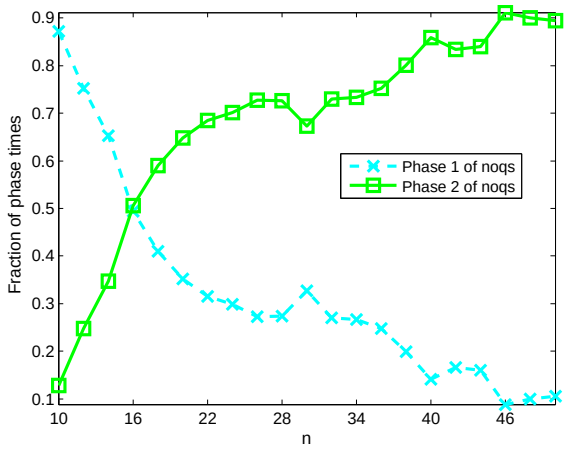

Figure 6: Fraction of noqs solve time devoted to each phase with increasing n for overdetermined cases.

formulation was posed.

- **Implementation.** A Matlab-based implementation, noqs, of the active-set algorithm was developed. A benchmark set consisting of 891 problems of varying sizes and noise levels, and different levels of conditioning was developed and used to compare noqs with other, less-specialized quadratic programming solvers. Overall, noqs was seen to perform faster and more robustly. Additionally, instances were discussed where a bound-constrained formulation can efficiently replace the original formulation.

A Appendix

A.1 Proof of Theorem 2.2

Proof. Since f_s is continuously differentiable, a first-order Taylor expansion (see, for example, [5, Section 4.1]) about $x \in \mathcal{B}_0$ gives

$$\begin{aligned} \langle e^g(x), x_i - x\rangle &= \int_0^1 \langle \nabla f_s\left(x + t(x_i - x)\right) - \nabla f_s(x), x_i - x\rangle dt \\ &\quad -\frac{1}{2}\langle H(x_i - x), x_i - x\rangle + e^q(x_i) - e^q(x), \end{aligned} \tag{33}$$

for any $x_i \in \mathcal{B}_0$, where $\langle \cdot, \cdot \rangle$ is the standard dot product on $\mathbb{R}^n$. Subtracting the equation associated with x_{i_0} shows that

$$\begin{aligned} \langle e^g(x), x_i - x_{i_0}\rangle &= \int_0^1 \langle \nabla f_s\left(x + t(x_i - x)\right) - \nabla f_s(x), x_i - x\rangle dt \\ &\quad - \int_0^1 \langle \nabla f_s\left(x + t(x_{i_0} - x)\right) - \nabla f_s(x), x_{i_0} - x\rangle dt \\ &\quad -\frac{1}{2}\langle H(x_i - x), x_i - x\rangle + \frac{1}{2}\langle H(x_{i_0} - x), x_{i_0} - x\rangle + e^q(x_i) - e^q(x_{i_0}), \end{aligned} \tag{34}$$

for any $x, x_i \in \mathcal{B}_0$. Since ∇f_s is γ_f-Lipschitz, the Cauchy-Schwartz inequality can be used to bound the first two terms in (34):

$$\begin{aligned} & \left| \int_0^1 \langle \nabla f_s\left(x + t(x_i - x)\right) - \nabla f_s(x), x_i - x \rangle dt \right| \\ \leq & \int_0^1 \|\nabla f_s\left(x + t(x_i - x)\right) - \nabla f_s(x)\| \|x_i - x\| dt \leq \frac{\gamma_f}{2} \|x_i - x\|^2. \end{aligned}$$

The third and fourth expressions can be similarly bounded by exploiting the relationship $\|\cdot\| \leq \|\cdot\|_F$ between the spectral and Frobenius matrix norms:

$$|\langle H(x_i - x), x_i - x \rangle| \leq \|H\|_F \|x_i - x\|^2.$$

The final two expressions can be bounded by appealing to the definition of e^q and using the assumption that $|q(x_i) - f(x_i)| \leq \epsilon$ for all $x_i \in \mathcal{X}$,

$$|e^q(x_i)| = |q(x_i) - f(x_i) + e^f(x_i)| \leq \epsilon + |e^f(x_i)|.$$

By assumption, the n columns of the matrix Y collecting n linearly independent, scaled displacements are each bounded by Δ in norm. Hence, the above bounds yield that

$$\|\Delta Y^T e^g(x)\| \leq \sqrt{n}\Delta \|Y^T e^g(x)\|_\infty \leq \sqrt{n} \left(\frac{5}{2}\gamma_f \Delta^2 + \frac{5}{2}\|H\|_F \Delta^2 + 2\epsilon + 2 \max_{i_0, \ldots, i_n} |e^f(x_i)| \right), \tag{35}$$

since $\|\cdot\| \leq \sqrt{n}\|\cdot\|_\infty$. Since Y is invertible, we obtain (15) by noting that

$$\|e^g(x)\| \leq \|Y^{-T}\| \|Y^T e^g(x)\| = \|Y^{-1}\| \|Y^T e^g(x)\|. \tag{36}$$

Returning to $e^q(x)$ in (33) and using $x_i = x_{i_0}$, we obtain (14) by noting that

$$|e^q(x)| \quad \leq \quad \|e^g(x)\| \|x_{i_0} - x\| + \frac{1}{2}\left(\gamma_f \Delta^2 + \|H\|_F \Delta^2\right) + |e^q(x_{i_0})|.$$

□

Acknowledgments

We are grateful to Jorge Moré for many useful discussions in the course of preparing the manuscript.

References

[1] A. Bandeira, K. Scheinberg, and L. Vicente, *Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization*, Mathematical Programming, 134 (2012), pp. 223–257.

[2] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Geometry of sample sets in derivative free optimization: Polynomial regression and underdetermined interpolation*, IMA J. Numer. Analysis, 28 (2008), pp. 721–748.

[3] ——, *Introduction to Derivative-Free Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[4] A. Custódio, H. Rocha, and L. Vicente, *Incorporating minimum Frobenius norm models indirect search*, Computational Optimization and Applications, 46 (2010), pp. 265–278.

[5] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, PA, 1996.

[6] E. D. Dolan, J. J. Moré, and T. S. Munson, *Optimality measures for performance profiles*, SIAM J. Optimization, 16 (2006), pp. 891–909.

[7] H. S. Dollar, N. I. M. Gould, M. Stoll, and A. J. Wathen, *Preconditioning saddle-point systems with applications in optimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 249–270.

[8] M. Friedlander and S. Leyffer, *Global and finite termination of a two-phase augmented Lagrangian filter method for general quadratic programs*, SIAM J. Sci. Comput., 30 (2008), pp. 1706–1729.

[9] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Inertia-controlling methods for general quadratic programming*, SIAM Review, 33 (1991), pp. 1–36.

[10] N. Gould, *An algorithm for large-scale quadratic programming*, IMA J. Numer. Analysis, 11 (1991), pp. 299–324.

[11] Mathworks, *Matlab Optimization Toolbox User's Guide*. Available at `http://www.mathworks.co.uk/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf`.

[12] J. J. Moré and S. M. Wild, *Estimating computational noise*, SIAM J. Sci. Comput., 33 (2011), pp. 1292–1314.

[13] A. Neumaier, *MINQ: general definite and bound constrained indefinite quadratic programming*. `http://www.mat.univie.ac.at/~neum/software/minq`, 1998.

[14] J. Nocedal and S. J. Wright, *Numerical Optimization*, SIAM, Philadephia, PA, 2nd ed., 2006.

[15] M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-Scale Nonlinear Optimization, Springer, 2006, pp. 255–297.

[16] ——, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Tech. Rep. NA2009/06, University of Cambridge, 2009.

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

[18] S. M. Wild and C. A. Shoemaker, *Global convergence of radial basis function trust region derivative-free algorithms*, SIAM Journal on Optimization, 21 (2011), pp. 761–781.