Adaptive Sampling Trust Region Optimization for Derivative-free Stochastic Functions and Deterministic Constraints

Joint work with Nicole Felice & Sara Shashaani (North Carolina State University)

Lindon Roberts, University of Melbourne (lindon.roberts@unimelb.edu.au)
Supported by the Australian Research Council (DE240100006)

AustMS Annual Meeting, University of Western Australia
1 Dec 2026

Key references

This talk is based on:

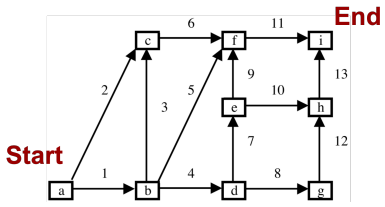
- Nicole Felice, LR & Sara Shashaani, Adaptive Sampling Trust-Region Optimization for Derivative-Free Stochastic Functions and Deterministic Equality Constraints, *Proc. Winter Simulation Conference* (2026). *[arXiv:2608.15894]*

Motivation

Stochastic Activity Networks (SANs) are a model for measuring the performance of a collection of tasks (which have dependencies). We try to allocate a given budget on reducing the time of each task, to minimize the completion time. [Athanassios et al., 1991]

Motivation

Stochastic Activity Networks (SANs) are a model for measuring the performance of a collection of tasks (which have dependencies). We try to allocate a given budget on reducing the time of each task, to minimize the completion time. [Athanassios et al., 1991]



The problem can be written as:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \mathbb{E}_\xi[T(\boldsymbol{x},\xi)] \qquad \text{s.t.} \quad \sum_{i=1}^{d} \frac{c_i}{x_i} = b,$$

where $T(\boldsymbol{x},\cdot)$ is the actual, random completion time of all tasks.

Actual completion times are exponentially distributed with mean x_i.

General Problem

This is an example of a stochastic optimization problem with deterministic, nonlinear constraints:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

General Problem

This is an example of a stochastic optimization problem with deterministic, nonlinear constraints:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

- Objective f is C^1, possibly nonconvex, but not directly observable

General Problem

This is an example of a stochastic optimization problem with deterministic, nonlinear constraints:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

- Objective f is C^1, possibly nonconvex, but not directly observable
- Constraints $\boldsymbol{c}$ are C^1, exact values and first derivatives available
  - Assume Jacobian $J(\boldsymbol{x}) = \nabla \boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^{p\times d}$ is uniformly bounded and singular values uniformly bounded away from zero $\Longrightarrow$ LICQ

General Problem

This is an example of a stochastic optimization problem with deterministic, nonlinear constraints:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

- Objective f is C^1, possibly nonconvex, but not directly observable
- Constraints $\boldsymbol{c}$ are C^1, exact values and first derivatives available
  - Assume Jacobian $J(\boldsymbol{x}) = \nabla \boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^{p\times d}$ is uniformly bounded and singular values uniformly bounded away from zero $\implies$ LICQ
- Sample paths $F(\boldsymbol{x},\cdot)$ are sub-exponentially distributed: e.g. [Wainwright, 2019]

$$\mathbb{P}\left[\left|\frac{1}{N}\sum_{j=1}^{N} F(\boldsymbol{x},\xi_j) - f(\boldsymbol{x})\right| \geq t\right] \leq \mathcal{O}(e^{-tN}) \qquad \text{for some } t > 0$$

General Problem

This is an example of a stochastic optimization problem with deterministic, nonlinear constraints:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

- Objective f is C^1, possibly nonconvex, but not directly observable
- Constraints $\boldsymbol{c}$ are C^1, exact values and first derivatives available
  - Assume Jacobian $J(\boldsymbol{x}) = \nabla \boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^{p\times d}$ is uniformly bounded and singular values uniformly bounded away from zero $\implies$ LICQ
- Sample paths $F(\boldsymbol{x},\cdot)$ are sub-exponentially distributed: e.g. [Wainwright, 2019]

$$\mathbb{P}\left[\left|\frac{1}{N}\sum_{j=1}^{N} F(\boldsymbol{x},\xi_j) - f(\boldsymbol{x})\right| \geq t\right] \leq \mathcal{O}(e^{-tN}) \qquad \text{for some } t > 0$$

- Sample gradients $\nabla_{\boldsymbol{x}} F(\boldsymbol{x},\cdot)$ are not available (e.g. complex simulation)

Basic Framework

There are several complications that we handle by combining existing approaches:

Basic Framework

There are several complications that we handle by combining existing approaches:

- No gradients for the objective
  - This means we have a derivative-free optimization (DFO) problem
  - We follow the successful *model-based* trust-region approach based on building locally accurate interpolation models e.g. [Conn et al., 2009; LR, 2025]

Basic Framework

There are several complications that we handle by combining existing approaches:

- No gradients for the objective
  - This means we have a derivative-free optimization (DFO) problem
  - We follow the successful *model-based* trust-region approach based on building locally accurate interpolation models e.g. [Conn et al., 2009; LR, 2025]
- Nonlinear equality constraints
  - Many approaches, but we use a sequential quadratic programming (SQP) approach
  - Existing convergence theory for gradient-based trust-region methods e.g. [Conn et al., 2000; Byrd et al., 2000]
  - Has been successfully adapted to model-based DFO setting, but without convergence guarantees [Tröltzsch, 2016; Ragonneau, 2022; Hannanu et al., 2025]

Basic Framework

There are several complications that we handle by combining existing approaches:

- No gradients for the objective
  - This means we have a derivative-free optimization (DFO) problem
  - We follow the successful *model-based* trust-region approach based on building locally accurate interpolation models e.g. [Conn et al., 2009; LR, 2025]
- Nonlinear equality constraints
  - Many approaches, but we use a sequential quadratic programming (SQP) approach
  - Existing convergence theory for gradient-based trust-region methods e.g. [Conn et al., 2000; Byrd et al., 2000]
  - Has been successfully adapted to model-based DFO setting, but without convergence guarantees [Tröltzsch, 2016; Ragonneau, 2022; Hannanu et al., 2025]
- (and stochastic objective estimates...)
  - SQP with stochastic gradients (e.g. ML) widely studied e.g. [Berahas et al., 2021; Fang et al., 2024]

SQP Methods

First, suppose we have a deterministic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

A point $\boldsymbol{x}^*$ is a KKT point if:

- $\boldsymbol{c}(\boldsymbol{x}^*) = \boldsymbol{0}$

SQP Methods

First, suppose we have a deterministic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

A point $\boldsymbol{x}^*$ is a KKT point if:

- $\boldsymbol{c}(\boldsymbol{x}^*) = \boldsymbol{0}$
- There exists $\boldsymbol{\lambda} \in \mathbb{R}^p$ such that $\nabla f(\boldsymbol{x}^*) = J(\boldsymbol{x}^*)^T \boldsymbol{\lambda}$

SQP Methods

First, suppose we have a deterministic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

A point $\boldsymbol{x}^*$ is a KKT point if:

- $\boldsymbol{c}(\boldsymbol{x}^*) = \boldsymbol{0}$
- There exists $\boldsymbol{\lambda} \in \mathbb{R}^p$ such that $\nabla f(\boldsymbol{x}^*) = J(\boldsymbol{x}^*)^T \boldsymbol{\lambda}$
  - $\nabla f(\boldsymbol{x}^*) \in \text{col}(J(\boldsymbol{x}^*)^T)$

SQP Methods

First, suppose we have a deterministic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

A point $\boldsymbol{x}^*$ is a KKT point if:

- $\boldsymbol{c}(\boldsymbol{x}^*) = \boldsymbol{0}$
- There exists $\boldsymbol{\lambda} \in \mathbb{R}^p$ such that $\nabla f(\boldsymbol{x}^*) = J(\boldsymbol{x}^*)^T \boldsymbol{\lambda}$
  - $\nabla f(\boldsymbol{x}^*) \in \text{col}(J(\boldsymbol{x}^*)^T)$
  - $N(\boldsymbol{x}^*)^T \nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$, where $N(\boldsymbol{x}) \in \mathbb{R}^{d\times(d-p)}$ is an orthonormal basis for $\text{nul}(J(\boldsymbol{x}))$

SQP Methods

First, suppose we have a deterministic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

A point $\boldsymbol{x}^*$ is a KKT point if:

- $\boldsymbol{c}(\boldsymbol{x}^*) = \boldsymbol{0}$
- There exists $\boldsymbol{\lambda} \in \mathbb{R}^p$ such that $\nabla f(\boldsymbol{x}^*) = J(\boldsymbol{x}^*)^T\boldsymbol{\lambda}$
  - $\nabla f(\boldsymbol{x}^*) \in \text{col}(J(\boldsymbol{x}^*)^T)$
  - $N(\boldsymbol{x}^*)^T\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$, where $N(\boldsymbol{x}) \in \mathbb{R}^{d\times(d-p)}$ is an orthonormal basis for $\text{nul}(J(\boldsymbol{x}))$

So, we can measure distance to first-order optimality via

$$\pi(\boldsymbol{x}) := \|\boldsymbol{c}(\boldsymbol{x})\| + \|N(\boldsymbol{x}^*)^T\nabla f(\boldsymbol{x}^*)\|$$

SQP Methods

Given a current iterate $\boldsymbol{x}_k$, SQP methods calculate a step by solving the (possibly nonconvex) QP

$$\min_{\boldsymbol{s}\in\mathbb{R}^d} m_k(\boldsymbol{s}) \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}_k) + J(\boldsymbol{x}_k)\boldsymbol{s} = \boldsymbol{0}$$

where $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ is a quadratic model for the objective (s.t. linearized constraints).

SQP Methods

Given a current iterate $\boldsymbol{x}_k$, SQP methods calculate a step by solving the (possibly nonconvex) QP

$$\min_{\boldsymbol{s}\in\mathbb{R}^d} m_k(\boldsymbol{s}) \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}_k) + J(\boldsymbol{x}_k)\boldsymbol{s} = \boldsymbol{0}$$

where $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ is a quadratic model for the objective (s.t. linearized constraints).

Problem

Adding a trust-region constraint (to ensure global convergence):

- Makes the step calculation no longer a QP (harder to solve)
- Can make the problem infeasible (if we are far from linearized constraints)

SQP Methods

The Byrd–Omojokun approach is a successful composite step idea:

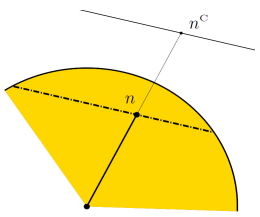
- Compute a normal step $\boldsymbol{n}_k$ to improve linearized feasibility
- Compute a tangential step $\boldsymbol{t}_k$ to decrease (approx) objective, without affecting feasibility
- Final step is $\boldsymbol{s}_k = \boldsymbol{n}_k + \boldsymbol{t}_k$ [Omojokun, 1989; Byrd et al., 2000]

SQP Methods

The Byrd–Omojokun approach is a successful composite step idea:

- Compute a normal step $\boldsymbol{n}_k$ to improve linearized feasibility
- Compute a tangential step $\boldsymbol{t}_k$ to decrease (approx) objective, without affecting feasibility
- Final step is $\boldsymbol{s}_k = \boldsymbol{n}_k + \boldsymbol{t}_k$

[Omojokun, 1989; Byrd et al., 2000]



- Compute $\boldsymbol{n}_k$ by minimizing $\|\boldsymbol{c}_k + J_k\boldsymbol{n}\|^2$ subject to $\|\boldsymbol{n}\| \leq 0.9\Delta_k$
- Compute $\boldsymbol{t}_k$ by minimizing $m_k(\boldsymbol{n}_k + \boldsymbol{t})$ subject to $J_k\boldsymbol{t} = \boldsymbol{0}$ and $\|\boldsymbol{t}\| \leq 0.1\Delta_k$ (eliminate linear constraints using QR factorization)
- Get $\|\boldsymbol{s}_k\| \leq \Delta_k$ since $\boldsymbol{n}_k, \boldsymbol{t}_k$ are orthogonal

With this step calculation, the high-level algorithm is:

1. Given $\boldsymbol{x}_k$ and Δ_k, build objective model $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ and evaluate constraints/derivatives $\boldsymbol{c}(\boldsymbol{x}_k)$ and $J(\boldsymbol{x}_k)$
2. Calculate step $\boldsymbol{s}_k = \boldsymbol{n}_k + \boldsymbol{t}_k$
3. Choose merit parameter $\zeta_k \geq \zeta_{k-1} > 0$ and define the merit function

$$\Phi_k(\boldsymbol{x}) := f(\boldsymbol{x}) + \zeta_k \|\boldsymbol{c}(\boldsymbol{x})\|$$

4. Evaluate $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and $\boldsymbol{c}(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and compute $\Phi_k(\boldsymbol{x}_k + \boldsymbol{s}_k)$
5. If $\boldsymbol{s}_k$ gives sufficient decrease in Φ_k and $\pi(\boldsymbol{x}_k) \geq \mu\Delta_k$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ and choose $\Delta_{k+1} \geq \Delta_k$
   - Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and $\Delta_{k+1} < \Delta_k$

Interpolation Model Construction

Model Construction

How do we build $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ without gradient information?

Interpolation Model Construction

Model Construction

How do we build $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ without gradient information?

Trust-region methods typically use the quadratic model

$$m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k \boldsymbol{s}, \qquad \text{where} \quad H_k \approx \nabla^2_{\boldsymbol{x}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$$

Interpolation Model Construction

Model Construction

How do we build $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ without gradient information?

Trust-region methods typically use the quadratic model

$$m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k \boldsymbol{s}, \qquad \text{where} \quad H_k \approx \nabla^2_{\boldsymbol{x}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$$

Without derivatives for f, we first build an interpolating quadratic model for f:

$$f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k^f(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T + \frac{1}{2}\boldsymbol{s}^T H_k^f \boldsymbol{s},$$

Interpolation Model Construction

Model Construction

How do we build $m_k(\boldsymbol{s}) \approx f(\boldsymbol{x}_k + \boldsymbol{s})$ without gradient information?

Trust-region methods typically use the quadratic model

$$m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k \boldsymbol{s}, \qquad \text{where} \quad H_k \approx \nabla^2_{\boldsymbol{x}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$$

Without derivatives for f, we first build an interpolating quadratic model for f:

$$f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k^f(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T + \frac{1}{2}\boldsymbol{s}^T H_k^f \boldsymbol{s},$$

We then estimate Lagrange multipliers, $J(\boldsymbol{x}_k)^T \boldsymbol{\lambda}_k \overset{\text{l.s.}}{=} \boldsymbol{g}_k$, and build

$$m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k \boldsymbol{s}, \qquad \text{where} \quad H_k = H_k^f - \sum_{i=1}^{p} [\lambda_k]_i \nabla^2 c_i(\boldsymbol{x}_k)$$

Interpolation Model Construction

Question

How accurate are interpolation models? How accurate does m_k need to be?

Interpolation Model Construction

Question

How accurate are interpolation models? How accurate does m_k need to be?

If m_k^f is built by interpolation to a 'good' set of points (i.e. $m_k^f(\boldsymbol{s}_j) = f(\boldsymbol{x}_k + \boldsymbol{s}_j)$ for some $\boldsymbol{s}_j$) and f is sufficiently smooth, then m_k^f is fully linear:

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k^f(\boldsymbol{s})| \leq \mathcal{O}(\Delta_k^2) \qquad \text{and} \qquad \|\nabla f(\boldsymbol{x}_k) - \boldsymbol{g}_k\| \leq \mathcal{O}(\Delta_k),$$

for all $\|\boldsymbol{s}\| \leq \Delta_k$.

Interpolation Model Construction

Question

How accurate are interpolation models? How accurate does m_k need to be?

If m_k^f is built by interpolation to a 'good' set of points (i.e. $m_k^f(\boldsymbol{s}_j) = f(\boldsymbol{x}_k + \boldsymbol{s}_j)$ for some $\boldsymbol{s}_j$) and f is sufficiently smooth, then m_k^f is fully linear:

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k^f(\boldsymbol{s})| \leq \mathcal{O}(\Delta_k^2) \qquad \text{and} \qquad \|\nabla f(\boldsymbol{x}_k) - \boldsymbol{g}_k\| \leq \mathcal{O}(\Delta_k),$$

for all $\|\boldsymbol{s}\| \leq \Delta_k$. This is as accurate as a linear Taylor series (up to constants).

Interpolation Model Construction

Question

How accurate are interpolation models? How accurate does m_k need to be?

If m_k^f is built by interpolation to a 'good' set of points (i.e. $m_k^f(\boldsymbol{s}_j) = f(\boldsymbol{x}_k + \boldsymbol{s}_j)$ for some $\boldsymbol{s}_j$) and f is sufficiently smooth, then m_k^f is fully linear:

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k^f(\boldsymbol{s})| \leq \mathcal{O}(\Delta_k^2) \qquad \text{and} \qquad \|\nabla f(\boldsymbol{x}_k) - \boldsymbol{g}_k\| \leq \mathcal{O}(\Delta_k),$$

for all $\|\boldsymbol{s}\| \leq \Delta_k$. This is as accurate as a linear Taylor series (up to constants).

For example, using $\boldsymbol{x}_k \pm \Delta_k \boldsymbol{e}_j$ for $j = 1, \ldots, d$ we have

$$[\boldsymbol{g}_k]_j = \frac{f(\boldsymbol{x}_k + \Delta_k \boldsymbol{e}_j) - f(\boldsymbol{x}_k - \Delta_k \boldsymbol{e}_j)}{2\Delta_k}, \; [H_k^f]_{j,j} = \frac{f(\boldsymbol{x}_k + \Delta_k \boldsymbol{e}_j) - 2f(\boldsymbol{x}_k) + f(\boldsymbol{x}_k - \Delta_k \boldsymbol{e}_j)}{\Delta_k^2}$$

Interpolation Model Construction

Question

How accurate are interpolation models? How accurate does m_k need to be?

If m_k^f is built by interpolation to a 'good' set of points (i.e. $m_k^f(\boldsymbol{s}_j) = f(\boldsymbol{x}_k + \boldsymbol{s}_j)$ for some $\boldsymbol{s}_j$) and f is sufficiently smooth, then m_k^f is fully linear:

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k^f(\boldsymbol{s})| \leq \mathcal{O}(\Delta_k^2) \qquad \text{and} \qquad \|\nabla f(\boldsymbol{x}_k) - \boldsymbol{g}_k\| \leq \mathcal{O}(\Delta_k),$$

for all $\|\boldsymbol{s}\| \leq \Delta_k$. This is as accurate as a linear Taylor series (up to constants).

For example, using $\boldsymbol{x}_k \pm \Delta_k \boldsymbol{e}_j$ for $j = 1, \ldots, d$ we have

$$[\boldsymbol{g}_k]_j = \frac{f(\boldsymbol{x}_k + \Delta_k \boldsymbol{e}_j) - f(\boldsymbol{x}_k - \Delta_k \boldsymbol{e}_j)}{2\Delta_k}, \; [H_k^f]_{j,j} = \frac{f(\boldsymbol{x}_k + \Delta_k \boldsymbol{e}_j) - 2f(\boldsymbol{x}_k) + f(\boldsymbol{x}_k - \Delta_k \boldsymbol{e}_j)}{\Delta_k^2}$$

More sophisticated alternatives exist which recycle evaluations across iterations.

SQP Convergence

This is enough to ensure convergence of the SQP method.

Theorem [Felice et al., 2026]

If m_k^f is fully linear at each iteration, then $\pi(\boldsymbol{x}_k) \to 0$.

SQP Convergence

This is enough to ensure convergence of the SQP method.

Theorem [Felice et al., 2026]

If m_k^f is fully linear at each iteration, then $\pi(\boldsymbol{x}_k) \to 0$.

This implies $\|\boldsymbol{c}(\boldsymbol{x}_k)\| \to 0$ (feasibility) and $\|N(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k)\| \to 0$ (stationarity of Lagrangian).

SQP Convergence

This is enough to ensure convergence of the SQP method.

Theorem [Felice et al., 2026]

If m_k^f is fully linear at each iteration, then $\pi(\boldsymbol{x}_k) \to 0$.

This implies $\|\boldsymbol{c}(\boldsymbol{x}_k)\| \to 0$ (feasibility) and $\|N(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k)\| \to 0$ (stationarity of Lagrangian).

Remaining difficulty

Now that we have a convergence framework, how to we handle stochastic objective estimates?

Stochastic Objective Estimates

Let's return to the original, stochastic, problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}) := \mathbb{E}_{\xi}[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

Stochastic Objective Estimates

Let's return to the original, stochastic, problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

We cannot evaluate $f(\boldsymbol{x})$ exactly; we can only get random estimates

$$\overline{F}(\boldsymbol{x}, N) := \frac{1}{N}\sum_{j=1}^{N} F(\boldsymbol{x},\xi_j), \qquad \text{for i.i.d. noise } \xi_1,\ldots,\xi_N$$

Stochastic Objective Estimates

Let's return to the original, stochastic, problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := \mathbb{E}_\xi[F(\boldsymbol{x},\xi)], \qquad \text{s.t.} \quad \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \in \mathbb{R}^p$$

We cannot evaluate $f(\boldsymbol{x})$ exactly; we can only get random estimates

$$\overline{F}(\boldsymbol{x}, N) := \frac{1}{N}\sum_{j=1}^{N} F(\boldsymbol{x}, \xi_j), \qquad \text{for i.i.d. noise } \xi_1, \ldots, \xi_N$$

These estimates are used for:

- Interpolation model construction: $m_k^f(\boldsymbol{s}_j) = \overline{F}(\boldsymbol{x}_k + \boldsymbol{s}_j, N)$
- Merit function/sufficient decrease check: $\overline{\Phi}_k(\boldsymbol{x}) := \overline{F}(\boldsymbol{x}, N) + \zeta_k \|\boldsymbol{c}(\boldsymbol{x})\|$

Stochastic Objective Estimates

General Principle

In model-based DFO, if objective estimates $\overline{F}(\boldsymbol{x}, N)$ always have error $\mathcal{O}(\Delta_k^2)$, then everything still works. [Ehrhardt & LR, 2021]

Stochastic Objective Estimates

General Principle

In model-based DFO, if objective estimates $\overline{F}(\boldsymbol{x}, N)$ always have error $\mathcal{O}(\Delta_k^2)$, then everything still works. [Ehrhardt & LR, 2021]

With stochastic estimates, we are not guaranteed this accuracy.

Stochastic Objective Estimates

General Principle

In model-based DFO, if objective estimates $\overline{F}(\boldsymbol{x}, N)$ always have error $\mathcal{O}(\Delta_k^2)$, then everything still works. [Ehrhardt & LR, 2021]

With stochastic estimates, we are not guaranteed this accuracy. Monte Carlo sampling requires $\mathcal{O}(\Delta_k^{-4})$ evaluations to ensure sufficiently accurate with high probability.

Stochastic Objective Estimates

General Principle

In model-based DFO, if objective estimates $\overline{F}(\boldsymbol{x}, N)$ always have error $\mathcal{O}(\Delta_k^2)$, then everything still works. [Ehrhardt & LR, 2021]

With stochastic estimates, we are not guaranteed this accuracy. Monte Carlo sampling requires $\mathcal{O}(\Delta_k^{-4})$ evaluations to ensure sufficiently accurate with high probability.

Adaptive sampling increases the number of samples N until we are confident of $\mathcal{O}(\Delta_k^2)$ accuracy: [Shashaani et al., 2018; Ha et al., 2025]

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

where $\hat{\sigma}(\boldsymbol{x}, n)$ is the sample standard deviation using the same n objective estimates.

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Variance of the estimate mean is $\mathrm{Var}(\overline{F}(\boldsymbol{x}, n)) = \sigma(\boldsymbol{x}, n)/\sqrt{n}$, and we approximate the true standard deviation (of $\overline{F}$) by $\sigma(\boldsymbol{x}, n) \approx \hat{\sigma}(\boldsymbol{x}, n)$, using the standard unbiased estimator.

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Variance of the estimate mean is $\mathrm{Var}(\overline{F}(\boldsymbol{x}, n)) = \sigma(\boldsymbol{x}, n)/\sqrt{n}$, and we approximate the true standard deviation (of $\overline{F}$) by $\sigma(\boldsymbol{x}, n) \approx \hat{\sigma}(\boldsymbol{x}, n)$, using the standard unbiased estimator. With our rule, $\hat{\sigma} \to \sigma$ as $k \to \infty$, almost surely.

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Variance of the estimate mean is $\text{Var}(\overline{F}(\boldsymbol{x}, n)) = \sigma(\boldsymbol{x}, n)/\sqrt{n}$, and we approximate the true standard deviation (of $\overline{F}$) by $\sigma(\boldsymbol{x}, n) \approx \hat{\sigma}(\boldsymbol{x}, n)$, using the standard unbiased estimator. With our rule, $\hat{\sigma} \to \sigma$ as $k \to \infty$, almost surely.

Recall we assume our noise is sub-exponential, and so Bernstein's inequality gives

$$\mathbb{P}\left[\left|\overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})\right| \geq t\right] \leq \mathcal{O}(e^{-tN(\boldsymbol{x})}) \qquad \text{for some } t > 0$$

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Variance of the estimate mean is $\mathrm{Var}(\overline{F}(\boldsymbol{x}, n)) = \sigma(\boldsymbol{x}, n)/\sqrt{n}$, and we approximate the true standard deviation (of $\overline{F}$) by $\sigma(\boldsymbol{x}, n) \approx \hat{\sigma}(\boldsymbol{x}, n)$, using the standard unbiased estimator. With our rule, $\hat{\sigma} \to \sigma$ as $k \to \infty$, almost surely.

Recall we assume our noise is sub-exponential, and so Bernstein's inequality gives

$$\mathbb{P}\left[\left|\overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})\right| \geq t\right] \leq \mathcal{O}(e^{-tN(\boldsymbol{x})}) \qquad \text{for some } t > 0$$

$\log k$ factor? Apply sub-exponential concentration bound with $t \sim (\log k)^{1+\epsilon}\Delta_k^2$ gives failure probabilities $\mathcal{O}(k^{-1-\epsilon})$ which are summable.

Stochastic Objective Estimates

$$N(\boldsymbol{x}) = \min\left\{ n \in \mathbb{N} : \frac{\max(\hat{\sigma}(\boldsymbol{x}, n), \sigma_0)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{(\log k)^{1+\epsilon}} \right\}$$

Where does the adaptive sampling condition come from?

Variance of the estimate mean is $\mathrm{Var}(\overline{F}(\boldsymbol{x}, n)) = \sigma(\boldsymbol{x}, n)/\sqrt{n}$, and we approximate the true standard deviation (of $\overline{F}$) by $\sigma(\boldsymbol{x}, n) \approx \hat{\sigma}(\boldsymbol{x}, n)$, using the standard unbiased estimator. With our rule, $\hat{\sigma} \to \sigma$ as $k \to \infty$, almost surely.

Recall we assume our noise is sub-exponential, and so Bernstein's inequality gives

$$\mathbb{P}\left[\left|\overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})\right| \geq t\right] \leq \mathcal{O}(e^{-tN(\boldsymbol{x})}) \qquad \text{for some } t > 0$$

$\log k$ factor? Apply sub-exponential concentration bound with $t \sim (\log k)^{1+\epsilon}\Delta_k^2$ gives failure probabilities $\mathcal{O}(k^{-1-\epsilon})$ which are summable. $\implies$ Borel–Cantelli

Stochastic Objective Estimates

The key outcome of the adaptive sampling rule is:

Theorem [Ha et al., 2025]

If $\overline{E}(\boldsymbol{x}) := \overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})$ is the sampling error, then

Stochastic Objective Estimates

The key outcome of the adaptive sampling rule is:

Theorem [Ha et al., 2025]

If $\overline{E}(\boldsymbol{x}) := \overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})$ is the sampling error, then

$$\mathbb{P}\left[|\overline{E}(\boldsymbol{x}_k) - \overline{E}(\boldsymbol{x}_k + \boldsymbol{s}_k)| > \kappa\Delta_k^2 \text{ infinitely often}\right] = 0.$$

(and similarly for $\boldsymbol{x}_k$ vs. $\boldsymbol{x}_k + \boldsymbol{s}_j$ for interpolation model construction)

Stochastic Objective Estimates

The key outcome of the adaptive sampling rule is:

Theorem [Ha et al., 2025]

If $\overline{E}(\boldsymbol{x}) := \overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})$ is the sampling error, then

$$\mathbb{P}\left[|\overline{E}(\boldsymbol{x}_k) - \overline{E}(\boldsymbol{x}_k + \boldsymbol{s}_k)| > \kappa\Delta_k^2 \text{ infinitely often}\right] = 0.$$

(and similarly for $\boldsymbol{x}_k$ vs. $\boldsymbol{x}_k + \boldsymbol{s}_j$ for interpolation model construction)

Since we are always *comparing (estimated) objective values*, we only care about *differences in sampling errors*.

Stochastic Objective Estimates

The key outcome of the adaptive sampling rule is:

Theorem [Ha et al., 2025]

If $\overline{E}(\boldsymbol{x}) := \overline{F}(\boldsymbol{x}, N(\boldsymbol{x})) - f(\boldsymbol{x})$ is the sampling error, then

$$\mathbb{P}\left[|\overline{E}(\boldsymbol{x}_k) - \overline{E}(\boldsymbol{x}_k + \boldsymbol{s}_k)| > \kappa\Delta_k^2 \text{ infinitely often}\right] = 0.$$

(and similarly for $\boldsymbol{x}_k$ vs. $\boldsymbol{x}_k + \boldsymbol{s}_j$ for interpolation model construction)

Since we are always *comparing (estimated) objective values*, we only care about *differences in sampling errors*.

So, with probability 1, for all sufficiently large iterations, all objective estimates are 'good enough'.

Convergence Guarantee

As a result, we can prove convergence of our stochastic SQP algorithm:

Theorem [Felice et al., 2026]

With our adaptive sampling rule, $\pi(\boldsymbol{x}_k) \to 0$ almost surely.

Convergence Guarantee

As a result, we can prove convergence of our stochastic SQP algorithm:

Theorem [Felice et al., 2026]

With our adaptive sampling rule, $\pi(\boldsymbol{x}_k) \to 0$ almost surely.

Again, this implies $\|\boldsymbol{c}(\boldsymbol{x}_k)\| \to 0$ (feasibility) and $\|N(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k)\| \to 0$ (stationarity of Lagrangian) almost surely.

Convergence Guarantee

As a result, we can prove convergence of our stochastic SQP algorithm:

Theorem [Felice et al., 2026]

With our adaptive sampling rule, $\pi(\boldsymbol{x}_k) \to 0$ almost surely.

Again, this implies $\|\boldsymbol{c}(\boldsymbol{x}_k)\| \to 0$ (feasibility) and $\|N(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k)\| \to 0$ (stationarity of Lagrangian) almost surely.

Key proof steps (all hold almost surely):

1. Good interpolation set + adaptive sampling: error in estimated merit function is $\mathcal{O}(\Delta_k^2)$ (k sufficiently large)
2. Estimated stationarity $\|\boldsymbol{c}(\boldsymbol{x}_k)\| + \|J(\boldsymbol{x}_k)^T \boldsymbol{g}_k\|$ is $\mathcal{O}(\Delta_k)$ (k sufficiently large)
3. $\Delta_k \to 0$

Numerical Experiments

We implemented this algorithm as an extension of ASTRO-DF, part of the SimOpt library. [Eckman et al., 2023 & 2025]

Numerical Experiments

We implemented this algorithm as an extension of ASTRO-DF, part of the SimOpt library. [Eckman et al., 2023 & 2025]

Compare against COBYQA, a model-based DFO method included in SciPy (handles nonlinear constraints via SQP, advanced interpolation set management, no convergence theory). [Ragonneau, 2022]

Numerical Experiments

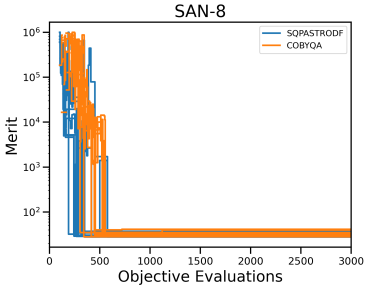
We implemented this algorithm as an extension of ASTRO-DF, part of the SimOpt library. [Eckman et al., 2023 & 2025]

Compare against COBYQA, a model-based DFO method included in SciPy (handles nonlinear constraints via SQP, advanced interpolation set management, no convergence theory). [Ragonneau, 2022]

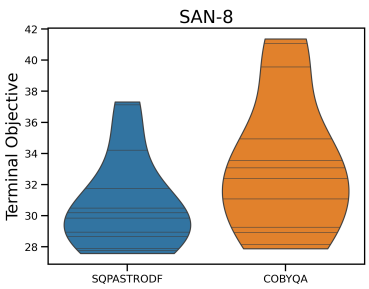
Tested on synthetic SAN problems of varying (low) dimension d; ran both solvers 10 times on each problem.

Numerical Results

Our method reliably converges faster, and to a lower objective value, than COBYQA (both solvers always found a feasible point).



Merit function vs. evaluations

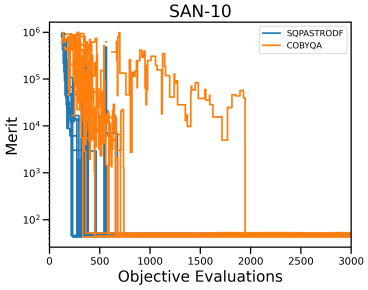


Final objective value distribution

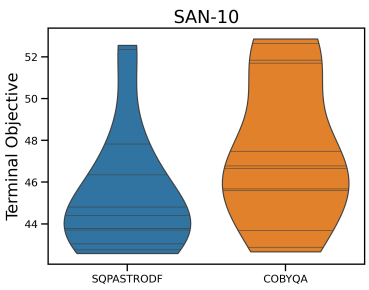
SQP-ASTRO-DF vs. COBYQA on example SAN problem, dimension $d = 8$

Numerical Results

Our method reliably converges faster, and to a lower objective value, than COBYQA (both solvers always found a feasible point).



Merit function vs. evaluations

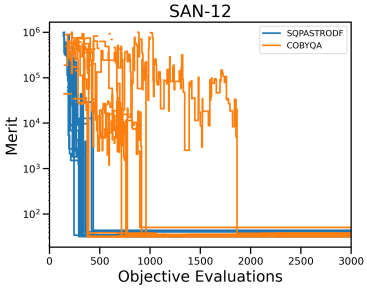


Final objective value distribution

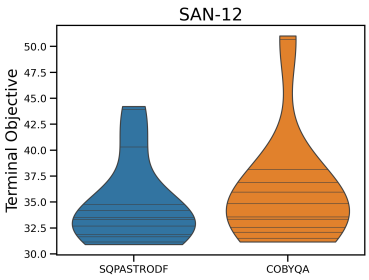
SQP-ASTRO-DF vs. COBYQA on example SAN problem, dimension $d = 10$

Numerical Results

Our method reliably converges faster, and to a lower objective value, than COBYQA (both solvers always found a feasible point).



Merit function vs. evaluations

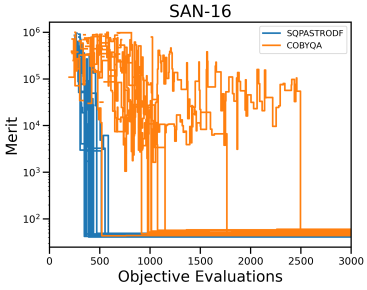


Final objective value distribution

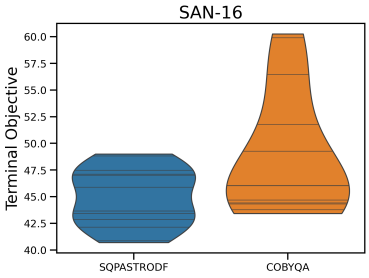
SQP-ASTRO-DF vs. COBYQA on example SAN problem, dimension $d = 12$

Numerical Results

Our method reliably converges faster, and to a lower objective value, than COBYQA (both solvers always found a feasible point).



Merit function vs. evaluations



Final objective value distribution

SQP-ASTRO-DF vs. COBYQA on example SAN problem, dimension $d = 16$

Conclusions & Future Work

Conclusions

- Extended model-based DFO to handle nonlinear constraints (with derivatives)
- Handle stochastic noise with adaptive sampling
- Convergence theory with good numerical performance

Future Work

- Inequality constraints
- Derivative-free and/or stochastic constraints
- Worst-case complexity

References i

N. A. Athanassios, K. W. Bauer, and J. R. Wilson, *Simulation of stochastic activity networks using path control variates*, Naval Research Logistics (NRL), 38 (1991), pp. 183–201.

A. S. Berahas, F. E. Curtis, D. Robinson, and B. Zhou, *Sequential quadratic optimization for nonlinear equality constrained stochastic optimization*, SIAM Journal on Optimization, 31 (2021), pp. 1352–1379.

R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Mathematical Programming, 89 (2000), pp. 149–185.

A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, vol. 1 of MPS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, vol. 8 of MPS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics/Mathematical Programming Society, Philadelphia, 2009.

D. J. Eckman, S. G. Henderson, and S. Shashaani, *Diagnostic tools for evaluating and comparing simulation-optimization algorithms*, INFORMS Journal on Computing, 35 (2023), pp. 350–367.

References ii

D. J. Eckman, S. G. Henderson, S. Shashaani, and R. Pasupathy, *SimOpt*. **https://github.com/simopt-admin/simopt, 2025.**

M. J. Ehrhardt and L. Roberts, *Inexact derivative-free optimization for bilevel learning*, Journal of Mathematical Imaging and Vision, 63 (2021), pp. 580–600.

Y. Fang, S. Na, M. W. Mahoney, and M. Kolar, *Fully stochastic trust-region sequential quadratic programming for equality-constrained optimization problems*, SIAM Journal on Optimization, 34 (2024), pp. 2007–2037.

N. Felice, L. Roberts, and S. Shashaani, *Adaptive sampling trust region optimization for derivative-free stochastic functions and deterministic equality constraints*, in Proceedings of the Winter Simulation Conference Conference, 2026.

Y. Ha, S. Shashaani, and R. Pasupathy, *Complexity of zeroth- and first-order stochastic trust-region algorithms*, SIAM Journal on Optimization, 35 (2025), pp. 2098–2127.

References iii

M. I. Hannanu, E. Camponogara, T. L. Silva, and M. Hovd, *A modified derivative-free SQP-filter trust-region method for uncertainty handling: Application in gas-lift optimization*, Optimization and Engineering, 26 (2025), pp. 401–429.

E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, PhD thesis, University of Colorado at Boulder, 1989.

T. M. Ragonneau, *Model-Based Derivative-Free Optimization Methods and Software*, PhD thesis, Hong Kong Polytechnic University, 2022.

L. Roberts, *Introduction to model-based derivative-free optimization*, arXiv preprint arXiv:2510.04473, (2025).

S. Shashaani, F. S. Hashemi, and R. Pasupathy, *ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization*, SIAM Journal on Optimization, 28 (2018), pp. 3145–3176.

A. Tröltzsch, *A sequential quadratic programming algorithm for equality-constrained optimization without derivatives*, Optimization Letters, 10 (2016), pp. 383–399.

M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, Cambridge University Press, 2019.