Model Construction for Convex-Constrained Derivative-Free Optimization

Joint work with Matthew Hough (Waterloo)

Lindon Roberts, University of Sydney (`lindon.roberts@sydney.edu.au`)
$\longrightarrow$ **University of Melbourne, July 2025**

WOMBAT, University of Sydney
4 December 2024

Further Reading

This talk is based on:

- M. Hough & LR, Model-Based Derivative-Free Methods for Convex-Constrained Optimization, *SIAM J. Optim* 32:4 (2022), pp. 2552–2579.
- LR, Model Construction for Convex-Constrained Derivative-Free Optimization, *arXiv:2403.14960* (2024).

Outline

1. **Convex-constrained derivative-free optimisation (DFO)**
2. Quadratic model construction

Convex-Constrained DFO

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{x} \in C.$$

- Objective $f : \mathbb{R}^n \to \mathbb{R}$ is smooth (C^1 with Lipschitz gradient) and nonconvex
- Constraint set C is closed and convex, with nonempty interior and easy-to-compute Euclidean projection

$$\text{proj}_C(\boldsymbol{x}) := \arg\min_{\boldsymbol{y}\in C} \|\boldsymbol{y} - \boldsymbol{x}\|_2.$$

e.g. bounds, ball, linear inequalities, ...

Convex-Constrained DFO

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \qquad \text{s.t.} \quad \mathbf{x} \in C.$$

- Objective $f:\mathbb{R}^n \to \mathbb{R}$ is smooth (C^1 with Lipschitz gradient) and nonconvex
- Constraint set C is closed and convex, with nonempty interior and easy-to-compute Euclidean projection

$$\text{proj}_C(\mathbf{x}) := \arg\min_{\mathbf{y}\in C} \|\mathbf{y}-\mathbf{x}\|_2.$$

  e.g. bounds, ball, linear inequalities, ...

Focus is on the derivative-free optimisation (DFO) setting: although ∇f exists, we only have access to the zero-th order oracle $\mathbf{x} \mapsto f(\mathbf{x})$.

Convex-Constrained DFO

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{x} \in C.$$

- Objective $f : \mathbb{R}^n \to \mathbb{R}$ is smooth (C^1 with Lipschitz gradient) and nonconvex
- Constraint set C is closed and convex, with nonempty interior and easy-to-compute Euclidean projection

$$\text{proj}_C(\boldsymbol{x}) := \arg\min_{\boldsymbol{y}\in C} \|\boldsymbol{y} - \boldsymbol{x}\|_2.$$

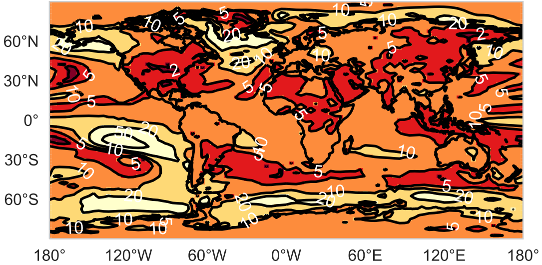
  e.g. bounds, ball, linear inequalities, ...

Focus is on the derivative-free optimisation (DFO) setting: although ∇f exists, we only have access to the zero-th order oracle $\boldsymbol{x} \mapsto f(\boldsymbol{x})$.

Looking for a strictly feasible method, i.e. cannot evaluate f at infeasible points (e.g. $\sqrt{x}$ with $x \geq 0$).

Application 1: Climate Modelling [Tett et al., 2022]

- Parameter calibration for global climate models (least squares minimisation)
- One model run = simulate global climate for 5 years = expensive
- Very complicated, chaotic physics = black-box & noisy
- Box constraints, $\boldsymbol{x} \in [\boldsymbol{x}_L, \boldsymbol{x}_U]$, expected parameter ranges



Applications

Application 2: Adversarial Example Generation [Alzantot et al., 2019]

- Find perturbations of neural network inputs which are misclassified (min. probability of correct label/max. probability of desired incorrect label)
- Neural network structure assumed to be unknown = black-box
- Want to test very few examples ≈ expensive
- Useful for copyright protection of artists' work against generative AI [Shan et al., 2023]
- Box or ball constraints to find small perturbation, $\boldsymbol{x} \approx \boldsymbol{x}_{\text{orig}}$

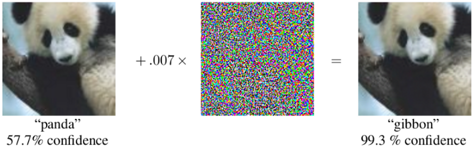


Image from [Goodfellow et al., 2015]

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

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

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

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

- Instead, approximate

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

  and find $\boldsymbol{g}_k$ and H_k without using derivatives

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

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

- Instead, approximate

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

  and find $\boldsymbol{g}_k$ and H_k <u>without</u> using derivatives

- How? Interpolate f over a set of points
- Geometry of points good $\Longrightarrow$ interpolation model Taylor-accurate $\Longrightarrow$ convergence

[Powell, 2003; Conn, Scheinberg & Vicente, 2009]

Model-Based DFO — Basic Ideas

Implement in trust-region method:

1. Build interpolation model $m_k(\boldsymbol{s})$
2. Minimize model inside trust region

$$\boldsymbol{s}_k = \arg\min_{\boldsymbol{s}\in\mathbb{R}^n} m_k(\boldsymbol{s}) \quad \text{s.t.} \quad \|\boldsymbol{s}\|_2 \leq \Delta_k, \; \boldsymbol{x}_k + \boldsymbol{s} \in C.$$

3. Accept/reject step and adjust Δ_k based on quality of new point $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k, & \text{if sufficient decrease,} \qquad \longleftarrow \text{(maybe increase } \Delta_k\text{)} \\ \boldsymbol{x}_k, & \text{otherwise.} \qquad \longleftarrow \text{(decrease } \Delta_k\text{)} \end{cases}$$

4. Update interpolation set: add $\boldsymbol{x}_k + \boldsymbol{s}_k$ to interpolation set
5. If needed, ensure new interpolation set is 'good'

Convergence? Define the stationarity measure (unconstrained case $\pi(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|$)

$$\pi(\boldsymbol{x}) := \left| \min_{\substack{\boldsymbol{x}+\boldsymbol{d}\in C \\ \|\boldsymbol{d}\|\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{d} \right|$$

Note: $\pi(\boldsymbol{x}) \geq 0$, $\pi(\boldsymbol{x}^*) = 0$ if and only if $\boldsymbol{x}^*$ first-order critical, Lipschitz continuous in $\boldsymbol{x}$.

Convergence? Define the stationarity measure (unconstrained case $\pi(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|$)

$$\pi(\boldsymbol{x}) := \left| \min_{\substack{\boldsymbol{x}+\boldsymbol{d}\in C \\ \|\boldsymbol{d}\|\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{d} \right|$$

Note: $\pi(\boldsymbol{x}) \geq 0$, $\pi(\boldsymbol{x}^*) = 0$ if and only if $\boldsymbol{x}^*$ first-order critical, Lipschitz continuous in $\boldsymbol{x}$.

Convergence & worst-case complexity match derivative-based trust-region methods.

Theorem (Hough & LR, 2022)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi(\boldsymbol{x}_k) = 0$. Furthermore, we achieve $\pi(\boldsymbol{x}_k) \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations.

Model-Based DFO — Theory

What is a 'good interpolation set' and 'good model'?

In the unconstrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is fully linear if, for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$,

  $$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k),$$

  (e.g. linear Taylor series)

Model-Based DFO — Theory

What is a 'good interpolation set' and 'good model'?

In the unconstrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is fully linear if, for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$,

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k),$$

  (e.g. linear Taylor series)
- An interpolation set is Λ-poised if

$$\max_t \max_{\|\boldsymbol{s}\|_2 \leq \Delta_k} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda,$$

  where ℓ_t is the t-th Lagrange polynomial for the set (i.e. $\ell_t(\boldsymbol{y}_s) = \delta_{s,t}$).

What is a 'good interpolation set' and 'good model'?

In the unconstrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is fully linear if, for all $\|\boldsymbol{s}\|_2 \le \Delta_k$,

  $$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k),$$

  (e.g. linear Taylor series)
- An interpolation set is Λ-poised if

  $$\max_t \max_{\|\boldsymbol{s}\|_2 \le \Delta_k} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \le \Lambda,$$

  where ℓ_t is the t-th Lagrange polynomial for the set (i.e. $\ell_t(\boldsymbol{y}_s) = \delta_{s,t}$).
- If interpolation set is Λ-poised and all points are $\mathcal{O}(\Delta_k^2)$ from $\boldsymbol{x}_k$, then the corresponding interpolation model is fully linear. [Conn, Scheinberg & Vicente, 2009]

Model-Based DFO — Theory

In the convex-constrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is C-fully linear if, for all $\|\boldsymbol{s}\|_2 \le \Delta_k$ with $\boldsymbol{x}_k + \boldsymbol{s} \in C$,

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k).$$

Model-Based DFO — Theory

In the convex-constrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is C-fully linear if, for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$ with $\boldsymbol{x}_k + \boldsymbol{s} \in C$,

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k).$$

- An interpolation set is Λ-poised in C if

$$\max_t \max_{\substack{\|\boldsymbol{s}\|_2 \leq \Delta_k \\ \boldsymbol{x}_k + \boldsymbol{s} \in C}} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda.$$

Model-Based DFO — Theory

In the convex-constrained case, we have:

- A model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is C-fully linear if, for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$ with $\boldsymbol{x}_k + \boldsymbol{s} \in C$,

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| = \mathcal{O}(\Delta_k^2), \quad \text{and} \quad \|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 = \mathcal{O}(\Delta_k).$$

- An interpolation set is Λ-poised in C if

$$\max_t \max_{\substack{\|\boldsymbol{s}\|_2 \leq \Delta_k \\ \boldsymbol{x}_k + \boldsymbol{s} \in C}} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda.$$

- If interpolation set is Λ-poised in C and all points are $\mathcal{O}(\Delta_k^2)$ from $\boldsymbol{x}_k$, then the corresponding interpolation model is C-fully linear. [Hough & LR, 2022]

Problem: this theory only works for linear interpolation, but practical methods require quadratic interpolation models.

Outline

1. Convex-constrained derivative-free optimisation (DFO)
2. **Quadratic model construction**

Quadratic Interpolation Models

We want to build a quadratic interpolation model

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

by defining an interpolation set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$ and requiring $m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t)$.

Quadratic Interpolation Models

We want to build a quadratic interpolation model

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

by defining an interpolation set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$ and requiring $m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t)$.

Since H_k is symmetric, we have $p = 1 + n + \frac{n(n+1)}{2} = \frac{(n+1)(n+2)}{2} \approx \frac{n^2}{2}$ degrees of freedom.

Quadratic Interpolation Models

We want to build a quadratic interpolation model

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

by defining an interpolation set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$ and requiring $m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t)$.

Since H_k is symmetric, we have $p = 1 + n + \frac{n(n+1)}{2} = \frac{(n+1)(n+2)}{2} \approx \frac{n^2}{2}$ degrees of freedom.

If we have this many interpolation points (usually including $\boldsymbol{x}_k$), then m_k is uniquely defined by solving a linear system for c_k, $\boldsymbol{g}_k$ and upper(H_k) (unless points are chosen badly).

Quadratic Interpolation Models

We want to build a quadratic interpolation model

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

by defining an interpolation set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$ and requiring $m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t)$.

Since H_k is symmetric, we have $p = 1 + n + \frac{n(n+1)}{2} = \frac{(n+1)(n+2)}{2} \approx \frac{n^2}{2}$ degrees of freedom.

If we have this many interpolation points (usually including $\boldsymbol{x}_k$), then m_k is uniquely defined by solving a linear system for c_k, $\boldsymbol{g}_k$ and upper(H_k) (unless points are chosen badly).

If n is even moderately large, then this requires a lot of evaluations. Can we use fewer points?

Underdetermined quadratic models

If we have $n+1 < p < \frac{(n+1)(n+2)}{2}$ interpolation points, there are (usually) infinitely many interpolating quadratics.

Underdetermined quadratic models

If we have $n+1 < p < \frac{(n+1)(n+2)}{2}$ interpolation points, there are (usually) infinitely many interpolating quadratics.

A practically successful choice is the minimum Hessian Frobenius norm model:

$$\min_{c_k, \boldsymbol{g}_k, H_k} \|H_k\|_F^2, \qquad \text{s.t.} \qquad m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t)\,\forall t = 1, \ldots, p, \quad \text{and} \quad H_k = H_k^T.$$

Quadratic Interpolation Models

Underdetermined quadratic models

If we have $n+1 < p < \frac{(n+1)(n+2)}{2}$ interpolation points, there are (usually) infinitely many interpolating quadratics.

A practically successful choice is the minimum Hessian Frobenius norm model:

$$\min_{c_k, \boldsymbol{g}_k, H_k} \|H_k\|_F^2, \qquad \text{s.t.} \qquad m_k(\boldsymbol{y}_t - \boldsymbol{x}_k) = f(\boldsymbol{y}_t) \, \forall t = 1, \ldots, p, \quad \text{and} \quad H_k = H_k^T.$$

Equality-constrained convex QP, reduces to size $p+n+1$ linear system with saddle point structure.

[Powell, 2004]

Underdetermined Quadratic Interpolation Models

Define the corresponding Lagrange polynomials in an analogous way:

$$\min_{c,\boldsymbol{g},H} \|H\|_F^2, \qquad \text{s.t.} \qquad \ell_t(\boldsymbol{y}_s - \boldsymbol{x}_k) = \delta_{s,t} \, \forall s = 1, \ldots, p, \quad \text{and} \quad H = H^T.$$

Define the corresponding Lagrange polynomials in an analogous way:

$$\min_{c,\boldsymbol{g},H} \|H\|_F^2, \qquad \text{s.t.} \qquad \ell_t(\boldsymbol{y}_s - \boldsymbol{x}_k) = \delta_{s,t} \, \forall s = 1, \ldots, p, \quad \text{and} \quad H = H^T.$$

In the unconstrained case, we have:

Theorem (Conn, Scheinberg, Vicente, 2009)

If the interpolation set is Λ-poised (using above defined Lagrange polynomials) and all points are distance $\mathcal{O}(\Delta_k^2)$ from $\boldsymbol{x}_k$, then the interpolation model is fully linear.

Define the corresponding Lagrange polynomials in an analogous way:

$$\min_{c,\boldsymbol{g},H} \|H\|_F^2, \qquad \text{s.t.} \qquad \ell_t(\boldsymbol{y}_s - \boldsymbol{x}_k) = \delta_{s,t} \; \forall s = 1, \ldots, p, \quad \text{and} \quad H = H^T.$$

In the unconstrained case, we have:

Theorem (Conn, Scheinberg, Vicente, 2009)

If the interpolation set is Λ*-poised (using above defined Lagrange polynomials) and all points are distance* $\mathcal{O}(\Delta_k^2)$ *from* $\boldsymbol{x}_k$*, then the interpolation model is fully linear.*

Moreover, the model Hessian $\|H_k\| = \mathcal{O}(\Lambda)$ *is uniformly bounded (standard requirement for trust-region convergence).*

Underdetermined Quadratic Interpolation Models

New result: theory extends to convex-constrained case exactly like linear interpolation models:

Theorem (LR, 2024)

If the interpolation set is Λ-poised in C and all points are distance $\mathcal{O}(\Delta_k^2)$ from $\boldsymbol{x}_k$, then the interpolation model is C-fully linear.

Note: if we require all points to be in C, then regular unconstrained definitions may yield arbitrarily large constants (ruins theoretical complexity bound).

Underdetermined Quadratic Interpolation Models

New result: theory extends to convex-constrained case exactly like linear interpolation models:

Theorem (LR, 2024)

If the interpolation set is Λ-poised in C and all points are distance $\mathcal{O}(\Delta_k^2)$ from $\boldsymbol{x}_k$, then the interpolation model is C-fully linear.

Note: if we require all points to be in C, then regular unconstrained definitions may yield arbitrarily large constants (ruins theoretical complexity bound).

We do not have a uniform bound on $\|H_k\|$, and instead can only bound Rayleigh quotient-type quantities:

$$\max_{s,t=1,\ldots,p} \frac{(\boldsymbol{y}_s - \boldsymbol{x}_k)^T H_k (\boldsymbol{y}_t - \boldsymbol{x}_k)}{\max_u \|\boldsymbol{y}_u - \boldsymbol{x}_k\|^2} \leq \mathcal{O}(\Lambda).$$

How do we make a set Λ-poised?

Algorithm to ensure Λ-poisedness:

- Find t and $\boldsymbol{y} \in B(\boldsymbol{x}_k, \Delta_k) \cap C$ with $|\ell_t(\boldsymbol{y} - \boldsymbol{x}_k)| > \Lambda$.
- If no such t and $\boldsymbol{y}$ exist, set is Λ-poised.
- If t and $\boldsymbol{y}$ found, replace $\boldsymbol{y}_t$ with $\boldsymbol{y}$ and loop.

Underdetermined Quadratic Interpolation Models

How do we make a set Λ-poised?

Algorithm to ensure Λ-poisedness:

- Find t and $\boldsymbol{y} \in B(\boldsymbol{x}_k, \Delta_k) \cap C$ with $|\ell_t(\boldsymbol{y} - \boldsymbol{x}_k)| > \Lambda$.
- If no such t and $\boldsymbol{y}$ exist, set is Λ-poised.
- If t and $\boldsymbol{y}$ found, replace $\boldsymbol{y}_t$ with $\boldsymbol{y}$ and loop.

This the unconstrained and extends to the convex-constrained case without issue. Key theoretical idea:

$$|\det(F_{\text{new}})| \geq \ell_t(\boldsymbol{y} - \boldsymbol{x}_k)^2 \cdot |\det(F_{\text{old}})| \geq \Lambda^2 \cdot |\det(F_{\text{old}})|,$$

where F_{old} and F_{new} are the linear systems for the minimum Frobenius QP before/after point swap. *(harder if F_{old} not invertible)*

Conclusions & Future Work

Conclusions

- Trust-region theory works with convex constraint sets
- Easy to construct feasible linear interpolation models
- New theory for minimum Frobenius norm quadratic interpolation models in feasible sets
  - Justifies steps used in state-of-the-art software (e.g. COBYQA in SciPy), where Lagrange polynomials are maximized subject to bound constraints.

Conclusions & Future Work

Conclusions

- Trust-region theory works with convex constraint sets
- Easy to construct feasible linear interpolation models
- New theory for minimum Frobenius norm quadratic interpolation models in feasible sets
  - Justifies steps used in state-of-the-art software (e.g. COBYQA in SciPy), where Lagrange polynomials are maximized subject to bound constraints.

Future Work

- Extend algorithm theory to second-order optimality
- Fully quadratic interpolation theory (i.e. using full $p \approx n^2/2$ interpolation points)

References i

M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. B. Srivastava, *GenAttack: Practical black-box attacks with gradient-free optimization*, in Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 2019, ACM, pp. 1111–1119.

M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

C. Cartis and L. Roberts, *Scalable subspace methods for derivative-free nonlinear least-squares optimization*, Mathematical Programming, 199 (2023), pp. 461—524.

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

I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, in 3rd International Conference on Learning Representations ICLR, San Diego, 2015.

M. Hough and L. Roberts, *Model-based derivative-free methods for convex-constrained optimization*, SIAM Journal on Optimization, 32 (2022), pp. 2552–2579.

References ii

M. J. D. Powell, *On trust region methods for unconstrained minimization without derivatives*, Mathematical Programming, 97 (2003), pp. 605–623.

———, *Least Frobenius norm updating of quadratic models that satisfy interpolation conditions*, Mathematical Programming, 100 (2004), pp. 183–215.

L. Roberts, *Model construction for convex-constrained derivative-free optimization*, arXiv preprint arXiv:2403.14960, (2024).

S. Shan, W. Ding, J. Passananti, H. Zheng, and B. Y. Zhao, *Prompt-specific poisoning attacks on text-to-image generative models*, arXiv preprint arXiv:2310.13828, (2023).

S. F. B. Tett, J. M. Gregory, N. Freychet, C. Cartis, M. J. Mineter, and L. Roberts, *Does model calibration reduce uncertainty in climate projections?*, Journal of Climate, 35 (2022), pp. 2585–2602.

Updating Invalid Set

If want to update an interpolation set with singular QP matrix

$$F = \begin{bmatrix} Q & M \\ M^T & 0 \end{bmatrix}.$$

- Compute QR factorization with column pivoting for both M and $\begin{bmatrix} Q \\ M^T \end{bmatrix}$
- Select a subset of $p \geq n+1$ points where both submatrices are full column rank (ensures F invertible)
- While need more interpolation points:
  - Find $\boldsymbol{y}$ such that $S(\boldsymbol{y}) := \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}_k\|^4 - \phi(\boldsymbol{y})^T F^{-1} \phi(\boldsymbol{y}) \neq 0$
  - Add $\boldsymbol{y}$ to the interpolation set, recompute F and loop

In the above, $\phi(\boldsymbol{y})$ is a specific vector satisfying $\ell_t(\boldsymbol{y} - \boldsymbol{x}_k) = \boldsymbol{e}_t^T F^{-1} \phi(\boldsymbol{y})$.

Updating Invalid Set

Why does this work?

Initial selection of points (both matrices full column rank) ensures F is initially invertible.

Updating Invalid Set

Why does this work?

Initial selection of points (both matrices full column rank) ensures F is initially invertible.

Adding $\boldsymbol{y}$ to the interpolation set yields, up to permutations,

$$F_{\text{new}} = \begin{bmatrix} F_{\text{old}} & \phi(\boldsymbol{y}) \\ \phi(\boldsymbol{y})^T & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}_k\|^4 \end{bmatrix},$$

and so F_{old} invertible and $S(\boldsymbol{y})$ is invertible (i.e. nonzero) implies F_{new} invertible.

Recall: if A is invertible, then the saddle point system

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

is invertible if and only if the Schur complement $S := C - B^T A^{-1} B$ is invertible.

[Benzi, Golub & Liesen, 2005]