Nonsmooth, Nonconvex Optimization
Algorithms and Examples

Michael L. Overton
Courant Institute of Mathematical Sciences
New York University

Convex and Nonsmooth Optimization Class, Spring 2016, Final Lecture

Mostly based on my research work with Jim Burke and Adrian Lewis

Introduction

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex
- Usually, but not always, locally Lipschitz: for all x there exists L_x such that $|f(x+d) - f(x)| \leq L_x \|d\|$ for small $\|d\|$

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex
- Usually, but not always, locally Lipschitz: for all x there exists L_x such that $|f(x+d) - f(x)| \leq L_x \|d\|$ for small $\|d\|$

Lots of interesting applications

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex
- Usually, but not always, locally Lipschitz: for all x there exists L_x such that $|f(x+d) - f(x)| \leq L_x\|d\|$ for small $\|d\|$

Lots of interesting applications

Any locally Lipschitz function is differentiable almost everywhere on its domain. So, whp, can evaluate gradient at any given point.

Nonsmooth, Nonconvex Optimization

Problem: find x that locally minimizes f, where $f : \mathbb{R}^n \to \mathbb{R}$ is

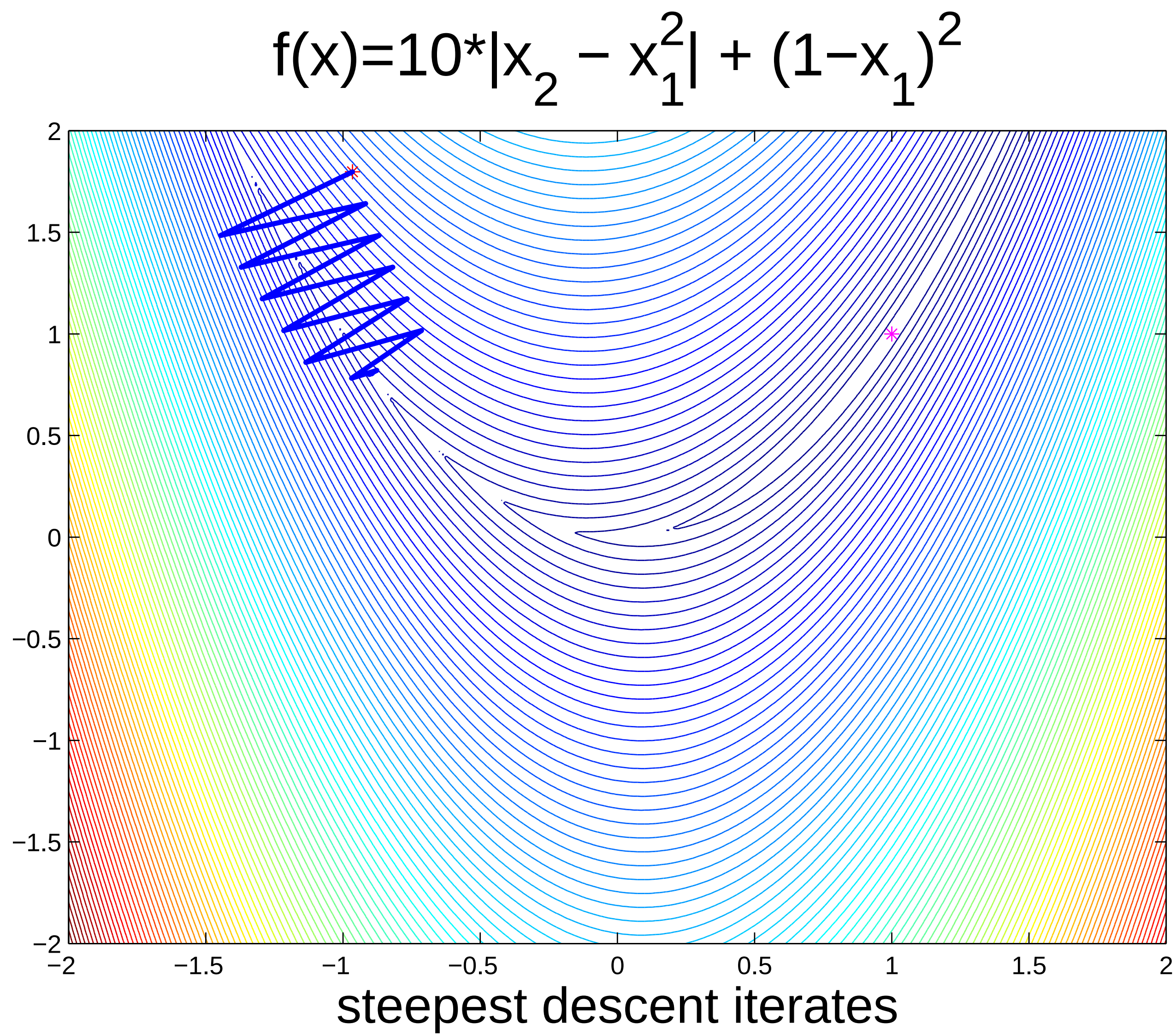
- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex
- Usually, but not always, locally Lipschitz: for all x there exists L_x such that $|f(x+d) - f(x)| \leq L_x \|d\|$ for small $\|d\|$

Lots of interesting applications

Any locally Lipschitz function is differentiable almost everywhere on its domain. So, whp, can evaluate gradient at any given point.

What happens if we simply use steepest descent (gradient descent) with a standard line search?

Example



f(x)=10*|x2 − x1^2| + (1−x1)^2
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
steepest descent iterates

Methods Suitable for Nonsmooth Functions

In fact, it's been known for several decades that at any given iterate, we need to exploit the gradient information obtained at several points, not just at one point. Some such methods:

Methods Suitable for Nonsmooth Functions

In fact, it's been known for several decades that at any given iterate, we need to exploit the gradient information obtained at several points, not just at one point. Some such methods:

- Bundle methods (C. Lemaréchal, K.C. Kiwiel, etc.): extensive practical use and theoretical analysis, but complicated in nonconvex case

Methods Suitable for Nonsmooth Functions

In fact, it's been known for several decades that at any given iterate, we need to exploit the gradient information obtained at several points, not just at one point. Some such methods:

- Bundle methods (C. Lemaréchal, K.C. Kiwiel, etc.): extensive practical use and theoretical analysis, but complicated in nonconvex case
- Gradient sampling: an easily stated method with nice convergence theory (J.V. Burke, A.S. Lewis, M.L.O., 2005; K.C. Kiwiel, 2007), but computationally intensive

Methods Suitable for Nonsmooth Functions

In fact, it's been known for several decades that at any given iterate, we need to exploit the gradient information obtained at several points, not just at one point. Some such methods:

- Bundle methods (C. Lemaréchal, K.C. Kiwiel, etc.): extensive practical use and theoretical analysis, but complicated in nonconvex case
- Gradient sampling: an easily stated method with nice convergence theory (J.V. Burke, A.S. Lewis, M.L.O., 2005; K.C. Kiwiel, 2007), but computationally intensive
- BFGS: traditional workhorse for smooth optimization, works amazingly well for nonsmooth optimization too, but very limited convergence theory

Failure of Steepest Descent: Simpler Example

Let $f(x) = 6|x_1| + 3x_2$. Note that f is polyhedral and convex.

Failure of Steepest Descent: Simpler Example

Let $f(x) = 6|x_1| + 3x_2$. Note that f is polyhedral and convex.

On this function, using a bisection-based backtracking line search with "Armijo" parameter in $[0, \frac{1}{3}]$ and starting at $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$, steepest descent generates the sequence

$$2^{-k} \begin{bmatrix} 2(-1)^k \\ 3 \end{bmatrix}, \quad k = 1, 2, \ldots,$$

converging to $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Failure of Steepest Descent: Simpler Example

Let $f(x) = 6|x_1| + 3x_2$. Note that f is polyhedral and convex.

On this function, using a bisection-based backtracking line search with "Armijo" parameter in $[0, \frac{1}{3}]$ and starting at $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$, steepest descent generates the sequence

$$2^{-k} \begin{bmatrix} 2(-1)^k \\ 3 \end{bmatrix}, \quad k = 1, 2, \ldots,$$

converging to $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

In contrast, BFGS with the same line search rapidly reduces the function value towards $-\infty$ (arbitrarily far, in exact arithmetic) (A.S. Lewis and S. Zhang, 2010).

Gradient Sampling

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x + \epsilon u_1), \ldots, \nabla f(x + \epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x+\epsilon u_1), \ldots, \nabla f(x+\epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball
  - Set $g = \arg\min\{||g|| : g \in \mathrm{conv}(G)\}$

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x + \epsilon u_1), \ldots, \nabla f(x + \epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball
  - Set $g = \arg\min\{||g|| : g \in \text{conv}(G)\}$
  - If $\|g\| > \tau$, do backtracking line search: set $d = -g/\|g\|$ and replace x by $x + td$, with $t \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ and $f(x+td) < f(x) - \beta t \|g\|$

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x + \epsilon u_1), \ldots, \nabla f(x + \epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball
  - Set $g = \arg\min\{||g|| : g \in \mathrm{conv}(G)\}$
  - If $\|g\| > \tau$, do backtracking line search: set $d = -g/\|g\|$ and replace x by $x + td$, with $t \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ and $f(x+td) < f(x) - \beta t\|g\|$
- until $\|g\| \leq \tau$.

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x+\epsilon u_1), \ldots, \nabla f(x+\epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball
  - Set $g = \arg\min\{||g|| : g \in \mathrm{conv}(G)\}$
  - If $\|g\| > \tau$, do backtracking line search: set $d = -g/\|g\|$ and replace x by $x + td$, with $t \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ and $f(x+td) < f(x) - \beta t\|g\|$
- until $\|g\| \leq \tau$.
- New phase: set $\epsilon = \mu\epsilon$ and $\tau = \theta\tau$.

The Gradient Sampling Method

Fix sample size $m \geq n+1$, line search parameter $\beta \in (0,1)$, reduction factors $\mu \in (0,1)$ and $\theta \in (0,1)$.

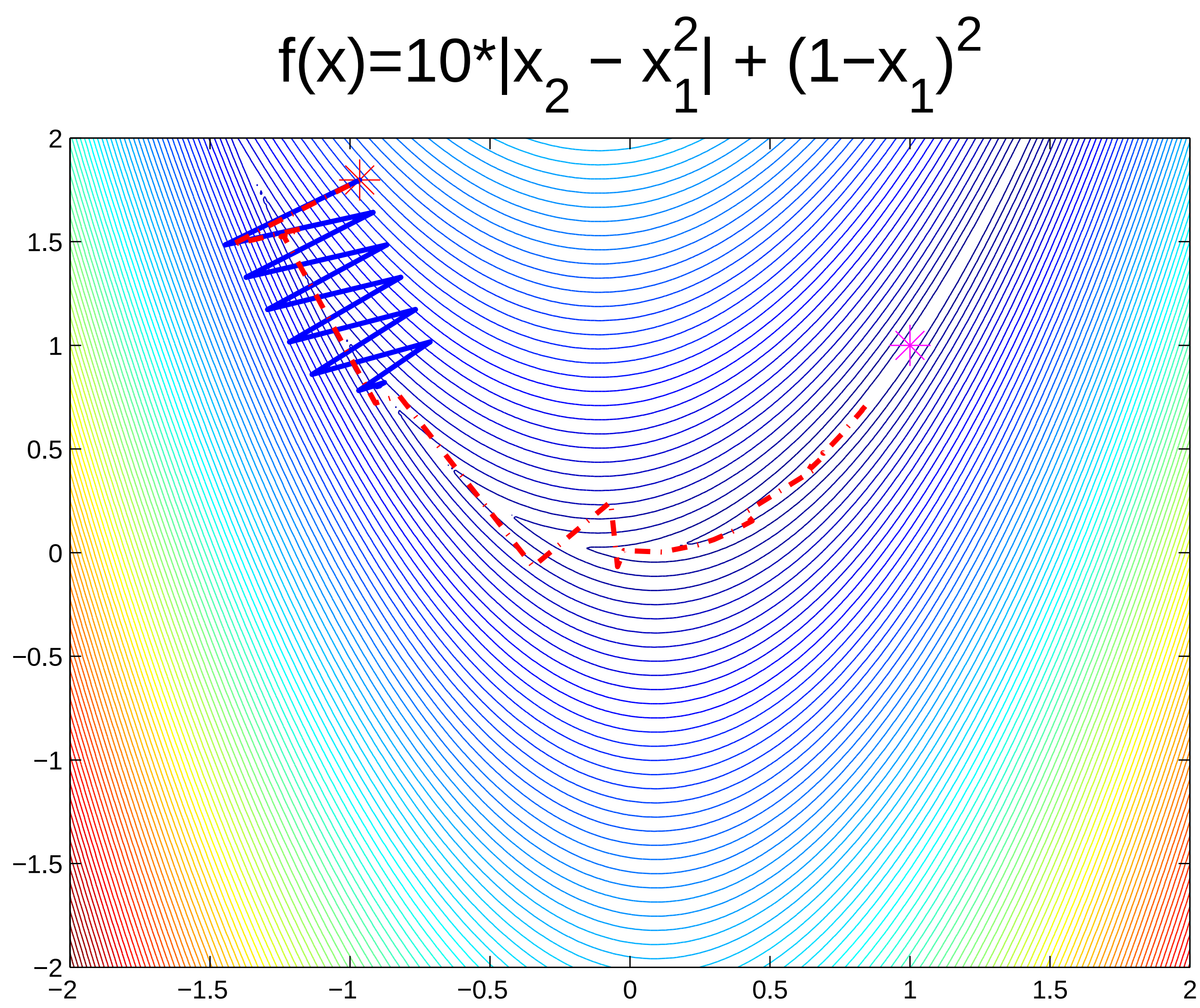
Initialize sampling radius $\epsilon > 0$, tolerance $\tau > 0$, iterate x.

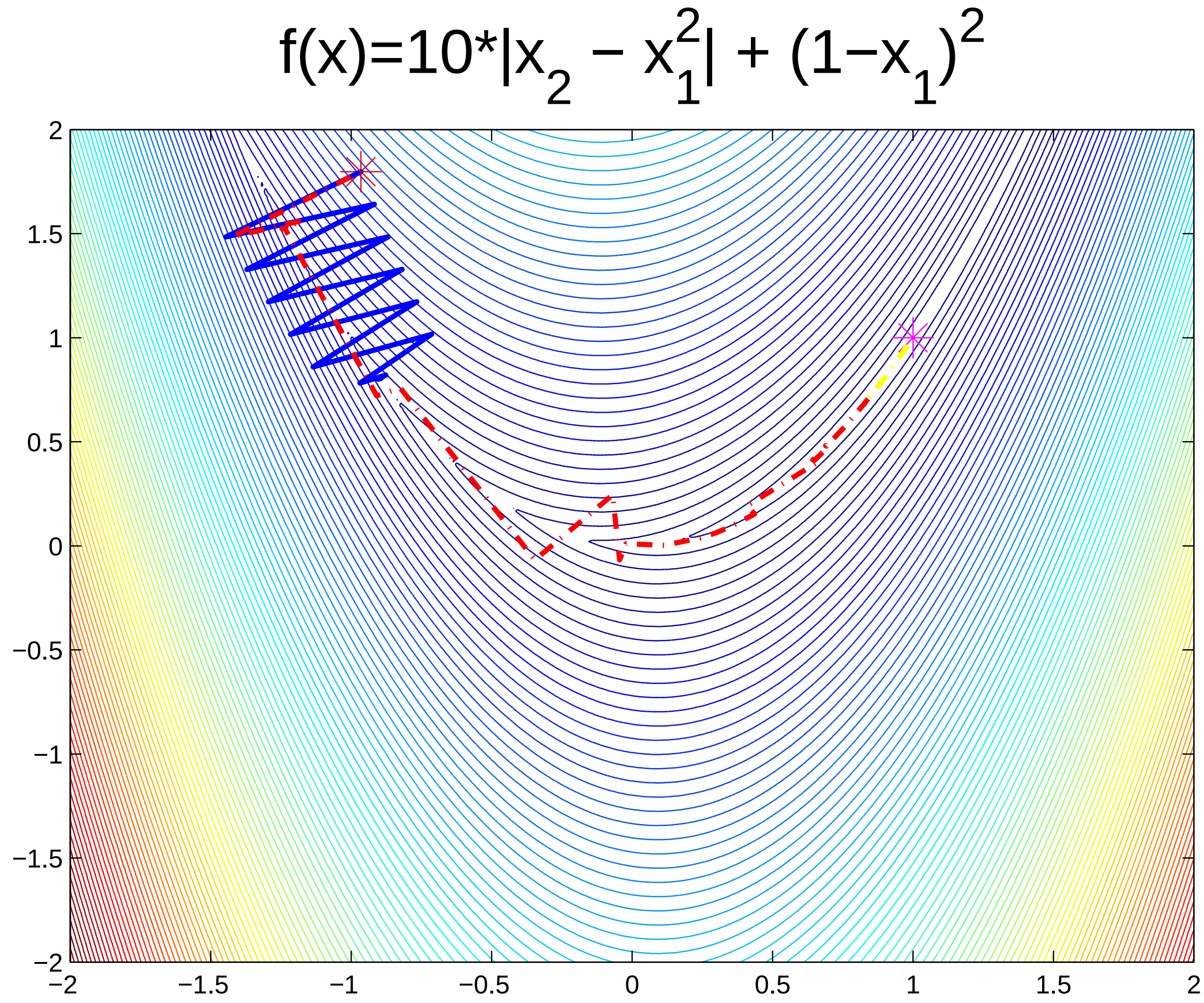
Repeat (outer loop)

- Repeat (inner loop: gradient sampling with fixed ϵ):
  - Set $G = \{\nabla f(x), \nabla f(x + \epsilon u_1), \ldots, \nabla f(x + \epsilon u_m)\}$, sampling $u_1, \cdots, u_m$ from the unit ball
  - Set $g = \arg\min\{||g|| : g \in \mathrm{conv}(G)\}$
  - If $\|g\| > \tau$, do backtracking line search: set $d = -g/\|g\|$ and replace x by $x + td$, with $t \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ and $f(x+td) < f(x) - \beta t \|g\|$
- until $\|g\| \leq \tau$.
- New phase: set $\epsilon = \mu\epsilon$ and $\tau = \theta\tau$.

J.V. Burke, A.S. Lewis and M.L.O., SIOPT, 2005.

With First Phase of Gradient Sampling




f(x)=10*|x_2 − x_1^2| + (1−x_1)^2
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ :\ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

F.H. Clarke, 1973 (he used the name "generalized gradient").

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

F.H. Clarke, 1973 (he used the name “generalized gradient”).

If f is continuously differentiable at $\bar{x}$, then $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and
let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

F.H. Clarke, 1973 (he used the name "generalized gradient").

If f is continuously differentiable at $\bar{x}$, then $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.

If f is convex, ∂f is the subdifferential of convex analysis.

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and
let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

F.H. Clarke, 1973 (he used the name "generalized gradient").

If f is continuously differentiable at $\bar{x}$, then $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.

If f is convex, ∂f is the subdifferential of convex analysis.

We say $\bar{x}$ is Clarke stationary for f if $0 \in \partial f(\bar{x})$.

The Clarke Subdifferential

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, and let $D = \{x \in \mathbb{R}^n \ : \ f \text{ is differentiable at } x\}$.

Rademacher's Theorem: $\mathbb{R}^n \backslash D$ has measure zero.

The Clarke subdifferential of f at $\bar{x}$ is

$$\partial f(\bar{x}) = \operatorname{conv}\left\{ \lim_{x \to \bar{x}, x \in D} \nabla f(x) \right\}.$$

F.H. Clarke, 1973 (he used the name "generalized gradient").

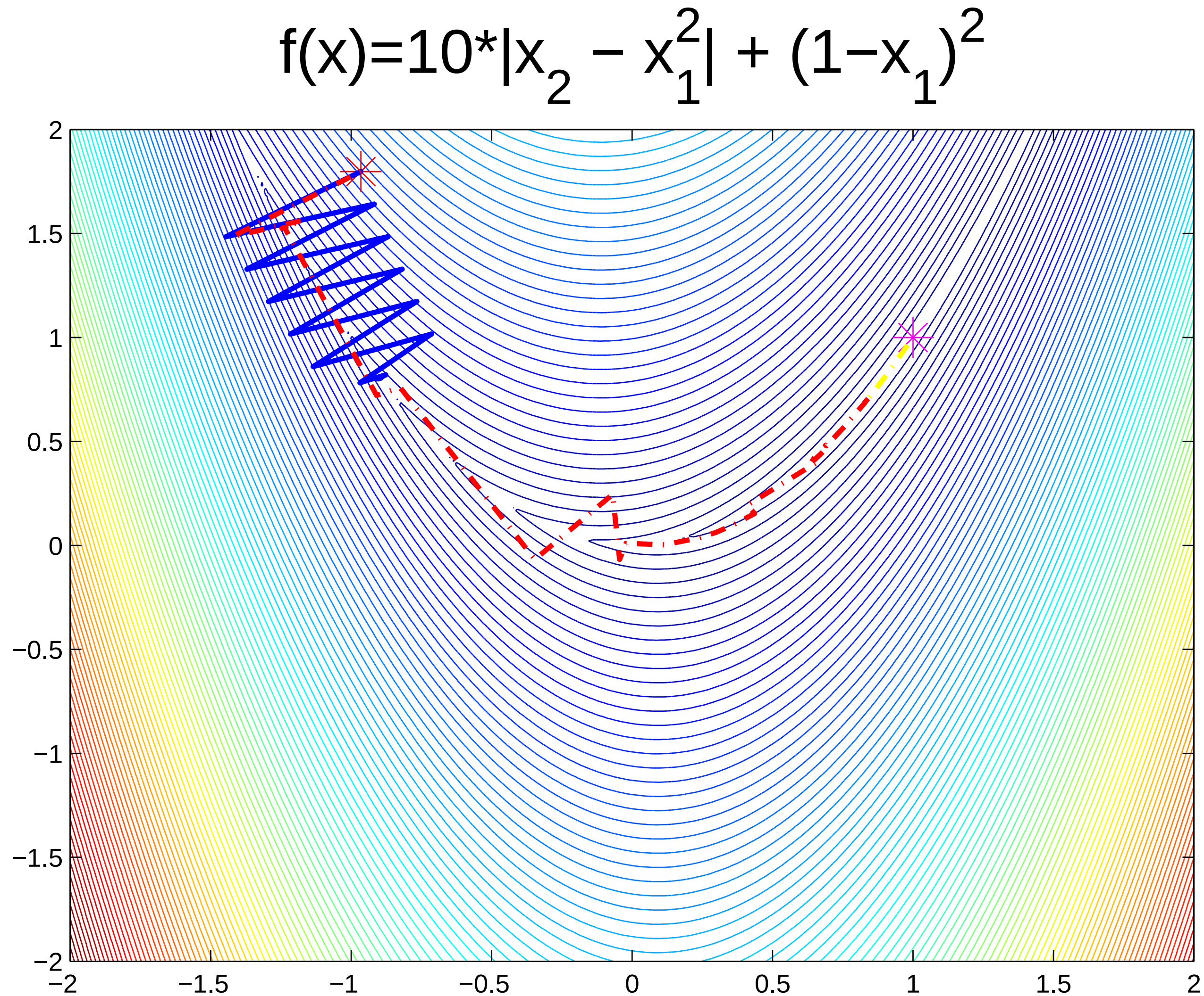
If f is continuously differentiable at $\bar{x}$, then $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.

If f is convex, ∂f is the subdifferential of convex analysis.

We say $\bar{x}$ is Clarke stationary for f if $0 \in \partial f(\bar{x})$.

Key point: the convex hull of the set G generated by Gradient Sampling is a surrogate for ∂f.

Note that $0 \in \partial f(x) = 0$ at $x = [1;\ 1]^T$



Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Grad. Samp.: A Stabilized Steepest Descent Method

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$-\mathrm{dist}(0, G) = -\min_{g \in G} \|g\|$$

Lemma. Let G be a compact convex set. Then

$$-\text{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$-\text{dist}(0, G) = -\min_{g \in G} \|g\|$$

$$= -\min_{g \in G} \max_{\|d\| \le 1} g^T d$$

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0,G) = \min_{\|d\|\le 1} \max_{g\in G} g^T d$$

Proof.

$$-\mathrm{dist}(0,G) = -\min_{g\in G} \|g\|$$

$$= -\min_{g\in G} \max_{\|d\|\le 1} g^T d$$

$$= -\max_{\|d\|\le 1} \min_{g\in G} g^T d$$

Grad. Samp.: A Stabilized Steepest Descent Method

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$-\mathrm{dist}(0, G) = -\min_{g \in G} \|g\|$$

$$= -\min_{g \in G} \max_{\|d\| \le 1} g^T d$$

$$= -\max_{\|d\| \le 1} \min_{g \in G} g^T d$$

$$= -\max_{\|d\| \le 1} \min_{g \in G} g^T (-d)$$

Grad. Samp.: A Stabilized Steepest Descent Method

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$\begin{aligned}
-\mathrm{dist}(0, G) &= -\min_{g \in G} \|g\| \\
&= -\min_{g \in G} \max_{\|d\| \le 1} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T (-d) \\
&= \min_{\|d\| \le 1} \max_{g \in G} g^T d.
\end{aligned}$$

Grad. Samp.: A Stabilized Steepest Descent Method

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0,G) = \min_{\|d\|\le 1} \max_{g\in G} g^T d$$

Proof.

$$\begin{aligned}
-\mathrm{dist}(0,G) &= -\min_{g\in G} \|g\| \\
&= -\min_{g\in G} \max_{\|d\|\le 1} g^T d \\
&= -\max_{\|d\|\le 1} \min_{g\in G} g^T d \\
&= -\max_{\|d\|\le 1} \min_{g\in G} g^T (-d) \\
&= \min_{\|d\|\le 1} \max_{g\in G} g^T d.
\end{aligned}$$

Note: the distance is nonnegative, and zero iff $0 \in G$.

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$\begin{aligned}
-\mathrm{dist}(0, G) &= -\min_{g \in G} \|g\| \\
&= -\min_{g \in G} \max_{\|d\| \le 1} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T(-d) \\
&= \min_{\|d\| \le 1} \max_{g \in G} g^T d.
\end{aligned}$$

Note: the distance is nonnegative, and zero iff $0 \in G$.
Otherwise, equality is attained by $g = \Pi_G(0)$, $d = -g/|g\|$.

Grad. Samp.: A Stabilized Steepest Descent Method

Lemma. Let G be a compact convex set. Then

$$-\mathrm{dist}(0, G) = \min_{\|d\| \le 1} \max_{g \in G} g^T d$$

Proof.

$$\begin{aligned}
-\mathrm{dist}(0, G) &= -\min_{g \in G} \|g\| \\
&= -\min_{g \in G} \max_{\|d\| \le 1} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T d \\
&= -\max_{\|d\| \le 1} \min_{g \in G} g^T(-d) \\
&= \min_{\|d\| \le 1} \max_{g \in G} g^T d.
\end{aligned}$$

Note: the distance is nonnegative, and zero iff $0 \in G$.
Otherwise, equality is attained by $g = \Pi_G(0)$, $d = -g/|g\|$.
Ordinary steepest descent: $G = \{\nabla f(x)\}$.

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

- the inner loop always terminates, so the sequences of sampling radii $\{\epsilon\}$ and tolerances $\{\tau\}$ converge to zero, and

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

- the inner loop always terminates, so the sequences of sampling radii $\{\epsilon\}$ and tolerances $\{\tau\}$ converge to zero, and
- $\bar{x}$ is Clarke stationary for f, i.e., $0 \in \partial f(\bar{x})$.

Convergence of Gradient Sampling Method

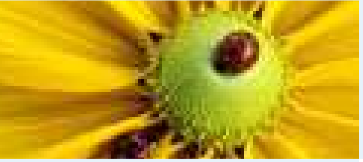
Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

- the inner loop always terminates, so the sequences of sampling radii $\{\epsilon\}$ and tolerances $\{\tau\}$ converge to zero, and
- $\bar{x}$ is Clarke stationary for f, i.e., $0 \in \partial f(\bar{x})$.

J.V. Burke, A.S. Lewis and M.L.O., SIOPT, 2005.



Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

- the inner loop always terminates, so the sequences of sampling radii $\{\epsilon\}$ and tolerances $\{\tau\}$ converge to zero, and
- $\bar{x}$ is Clarke stationary for f, i.e., $0 \in \partial f(\bar{x})$.

J.V. Burke, A.S. Lewis and M.L.O., SIOPT, 2005.

Drop the assumption that f has bounded level sets. Then, wp 1, either the sequence $\{f(x)\} \to -\infty$, or every cluster point of the sequence of iterates $\{x\}$ is Clarke stationary.

Convergence of Gradient Sampling Method

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$

- is locally Lipschitz
- is continuously differentiable on an open dense subset of $\mathbb{R}^n$
- has bounded level sets

Then, with probability one, the line search always terminates, f is differentiable at every iterate x, and if the sequence of iterates $\{x\}$ converges to some point $\bar{x}$, then, with probability one

- the inner loop always terminates, so the sequences of sampling radii $\{\epsilon\}$ and tolerances $\{\tau\}$ converge to zero, and
- $\bar{x}$ is Clarke stationary for f, i.e., $0 \in \partial f(\bar{x})$.

J.V. Burke, A.S. Lewis and M.L.O., SIOPT, 2005.

Drop the assumption that f has bounded level sets. Then, wp 1, either the sequence $\{f(x)\} \to -\infty$, or every cluster point of the sequence of iterates $\{x\}$ is Clarke stationary.

K.C. Kiwiel, SIOPT, 2007.

Extension to Problems with Nonsmooth Constraints

$$\min \; f(x)$$
$$\text{subject to } c_i(x) \le 0, \quad i = 1, \ldots, p$$

Extension to Problems with Nonsmooth Constraints

$$
\begin{aligned}
&\min \ f(x) \\
&\text{subject to } c_i(x) \le 0, \quad i = 1, \ldots, p
\end{aligned}
$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

Extension to Problems with Nonsmooth Constraints

$$\min \; f(x)$$
$$\text{subject to } c_i(x) \le 0, \quad i = 1, \ldots, p$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

A successive quadratic programming gradient sampling method with convergence theory.

Extension to Problems with Nonsmooth Constraints

$$\begin{aligned} &\min \ f(x) \\ &\text{subject to } c_i(x) \le 0, \quad i = 1, \ldots, p \end{aligned}$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

A successive quadratic programming gradient sampling method with convergence theory.

F.E. Curtis and M.L.O., SIOPT, 2012.

Quasi-Newton Methods

Bill Davidon

W. Davidon, a physicist at Argonne, had the breakthrough idea in 1959: since it's too expensive to compute and factor the Hessian $\nabla^2 f(x)$ at every iteration, update an approximation to its inverse using information from gradient differences, and multiply this onto the negative gradient to approximate Newton's method.

Bill Davidon

W. Davidon, a physicist at Argonne, had the breakthrough idea in 1959: since it's too expensive to compute and factor the Hessian $\nabla^2 f(x)$ at every iteration, update an approximation to its inverse using information from gradient differences, and multiply this onto the negative gradient to approximate Newton's method.

Each inverse Hessian approximation differs from the previous one by a rank-two correction.

Bill Davidon

W. Davidon, a physicist at Argonne, had the breakthrough idea in 1959: since it's too expensive to compute and factor the Hessian $\nabla^2 f(x)$ at every iteration, update an approximation to its inverse using information from gradient differences, and multiply this onto the negative gradient to approximate Newton's method.

Each inverse Hessian approximation differs from the previous one by a rank-two correction.

Ahead of its time: the paper was rejected by the physics journals, but published 30 years later in the first issue of SIAM J. Optimization.

Bill Davidon

W. Davidon, a physicist at Argonne, had the breakthrough idea in 1959: since it's too expensive to compute and factor the Hessian $\nabla^2 f(x)$ at every iteration, update an approximation to its inverse using information from gradient differences, and multiply this onto the negative gradient to approximate Newton's method.

Each inverse Hessian approximation differs from the previous one by a rank-two correction.

Ahead of its time: the paper was rejected by the physics journals, but published 30 years later in the first issue of SIAM J. Optimization.

Davidon was a well known active anti-war protester during the Vietnam War. In December 2013, it was revealed that he was the mastermind behind the break-in at the FBI office in Media, PA, on March 8, 1971, during the Muhammad Ali - Joe Frazier world heavyweight boxing championship.

Fletcher and Powell

In 1963, R. Fletcher and M.J.D. Powell improved Davidon's method and established convergence for convex quadratic functions.

Fletcher and Powell

In 1963, R. Fletcher and M.J.D. Powell improved Davidon's method and established convergence for convex quadratic functions.

They applied it to solve problems in 100 variables: a lot at the time.

Fletcher and Powell

In 1963, R. Fletcher and M.J.D. Powell improved Davidon's method and established convergence for convex quadratic functions.

They applied it to solve problems in 100 variables: a lot at the time.

The method became known as the DFP method.

BFGS

In 1970, C.G. Broyden, R. Fletcher, D. Goldfarb and D. Shanno all independently proposed the BFGS method, which is a kind of dual of the DFP method. It was soon recognized that this was a remarkably effective method for smooth optimization.

BFGS

In 1970, C.G. Broyden, R. Fletcher, D. Goldfarb and D. Shanno all independently proposed the BFGS method, which is a kind of dual of the DFP method. It was soon recognized that this was a remarkably effective method for smooth optimization.

In 1973, C.G. Broyden, J.E. Dennis and J.J. Moré proved generic local superlinear convergence of BFGS and DFP and other quasi-Newton methods.

BFGS

In 1970, C.G. Broyden, R. Fletcher, D. Goldfarb and D. Shanno all independently proposed the BFGS method, which is a kind of dual of the DFP method. It was soon recognized that this was a remarkably effective method for smooth optimization.

In 1973, C.G. Broyden, J.E. Dennis and J.J. Moré proved generic local superlinear convergence of BFGS and DFP and other quasi-Newton methods.

In 1975, M.J.D. Powell established convergence of BFGS with an inexact Armijo-Wolfe line search for a general class of smooth convex functions for BFGS. In 1987, this was extended by R.H. Byrd, J. Nocedal and Y.-X. Yuan to include the whole "Broyden" class of methods interpolating BFGS and DFP: *except* for the DFP end point.

BFGS

In 1970, C.G. Broyden, R. Fletcher, D. Goldfarb and D. Shanno all independently proposed the BFGS method, which is a kind of dual of the DFP method. It was soon recognized that this was a remarkably effective method for smooth optimization.

In 1973, C.G. Broyden, J.E. Dennis and J.J. Moré proved generic local superlinear convergence of BFGS and DFP and other quasi-Newton methods.

In 1975, M.J.D. Powell established convergence of BFGS with an inexact Armijo-Wolfe line search for a general class of smooth convex functions for BFGS. In 1987, this was extended by R.H. Byrd, J. Nocedal and Y.-X. Yuan to include the whole "Broyden" class of methods interpolating BFGS and DFP: *except* for the DFP end point.

Pathological counterexamples to convergence in the smooth, nonconvex case are known to exist (Y.-H. Dai, 2002, 2013; W. Mascarenhas 2004), but it is widely accepted that the method works well in practice in the smooth, nonconvex case.

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x+td) < f(x) + \beta t\alpha$ and $\nabla f(x+td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x+td) - \nabla f(x)$

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x + td) - \nabla f(x)$
- Replace x by $x + td$

The BFGS Method (”Full” Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x + td) - \nabla f(x)$
- Replace x by $x + td$
- Replace H by $VHV^T + \frac{1}{s^T y} ss^T$, where $V = I - \frac{1}{s^T y} sy^T$

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x + td) - \nabla f(x)$
- Replace x by $x + td$
- Replace H by $VHV^T + \frac{1}{s^T y} ss^T$, where $V = I - \frac{1}{s^T y} sy^T$

Note that H can be computed in $O(n^2)$ operations since V is a rank one perturbation of the identity

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x + td) - \nabla f(x)$
- Replace x by $x + td$
- Replace H by $VHV^T + \frac{1}{s^Ty}ss^T$, where $V = I - \frac{1}{s^Ty}sy^T$

Note that H can be computed in $O(n^2)$ operations since V is a rank one perturbation of the identity

The Armijo condition ensures "sufficient decrease" in f

The BFGS Method ("Full" Version)

Choose line search parameters $0 < \beta < \gamma < 1$

Initialize iterate x and positive-definite symmetric matrix H (which is supposed to approximate the *inverse* Hessian of f)

Repeat

- Set $d = -H\nabla f(x)$. Let $\alpha = \nabla f(x)^T d < 0$
- Armijo-Wolfe line search: find t so that $f(x + td) < f(x) + \beta t\alpha$ and $\nabla f(x + td)^T d > \gamma\alpha$
- Set $s = td$, $y = \nabla f(x + td) - \nabla f(x)$
- Replace x by $x + td$
- Replace H by $VHV^T + \frac{1}{s^Ty}ss^T$, where $V = I - \frac{1}{s^Ty}sy^T$

Note that H can be computed in $O(n^2)$ operations since V is a rank one perturbation of the identity

The Armijo condition ensures "sufficient decrease" in f

The Wolfe condition ensures that the directional derivative along the line increases algebraically, which guarantees that $s^Ty > 0$ and that the new H is positive definite.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

Key point: use the original Armijo-Wolfe line search. Do not insist on reducing the magnitude of the directional derivative along the line!

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

Key point: use the original Armijo-Wolfe line search. Do not insist on reducing the magnitude of the directional derivative along the line!

In the nonsmooth case, BFGS builds a very ill-conditioned inverse "Hessian" approximation, with some tiny eigenvalues converging to zero, corresponding to "infinitely large" curvature in the directions defined by the associated eigenvectors.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

Key point: use the original Armijo-Wolfe line search. Do not insist on reducing the magnitude of the directional derivative along the line!

In the nonsmooth case, BFGS builds a very ill-conditioned inverse "Hessian" approximation, with some tiny eigenvalues converging to zero, corresponding to "infinitely large" curvature in the directions defined by the associated eigenvectors.

Remarkably, the condition number of the inverse Hessian approximation typically reaches 10^{16} before the method breaks down.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

Key point: use the original Armijo-Wolfe line search. Do not insist on reducing the magnitude of the directional derivative along the line!

In the nonsmooth case, BFGS builds a very ill-conditioned inverse "Hessian" approximation, with some tiny eigenvalues converging to zero, corresponding to "infinitely large" curvature in the directions defined by the associated eigenvectors.

Remarkably, the condition number of the inverse Hessian approximation typically reaches 10^{16} before the method breaks down.

We have never seen convergence to non-stationary points that cannot be explained by numerical difficulties.

BFGS for Nonsmooth Optimization

In 1982, C. Lemaréchal observed that quasi-Newton methods can be effective for nonsmooth optimization, but dismissed them as there was no theory behind them and no good way to terminate them.

Otherwise, there is not much in the literature on the subject until A.S. Lewis and M.L.O. (Math. Prog., 2013): we address both issues in detail, but our convergence results are limited to special cases.

Key point: use the original Armijo-Wolfe line search. Do not insist on reducing the magnitude of the directional derivative along the line!

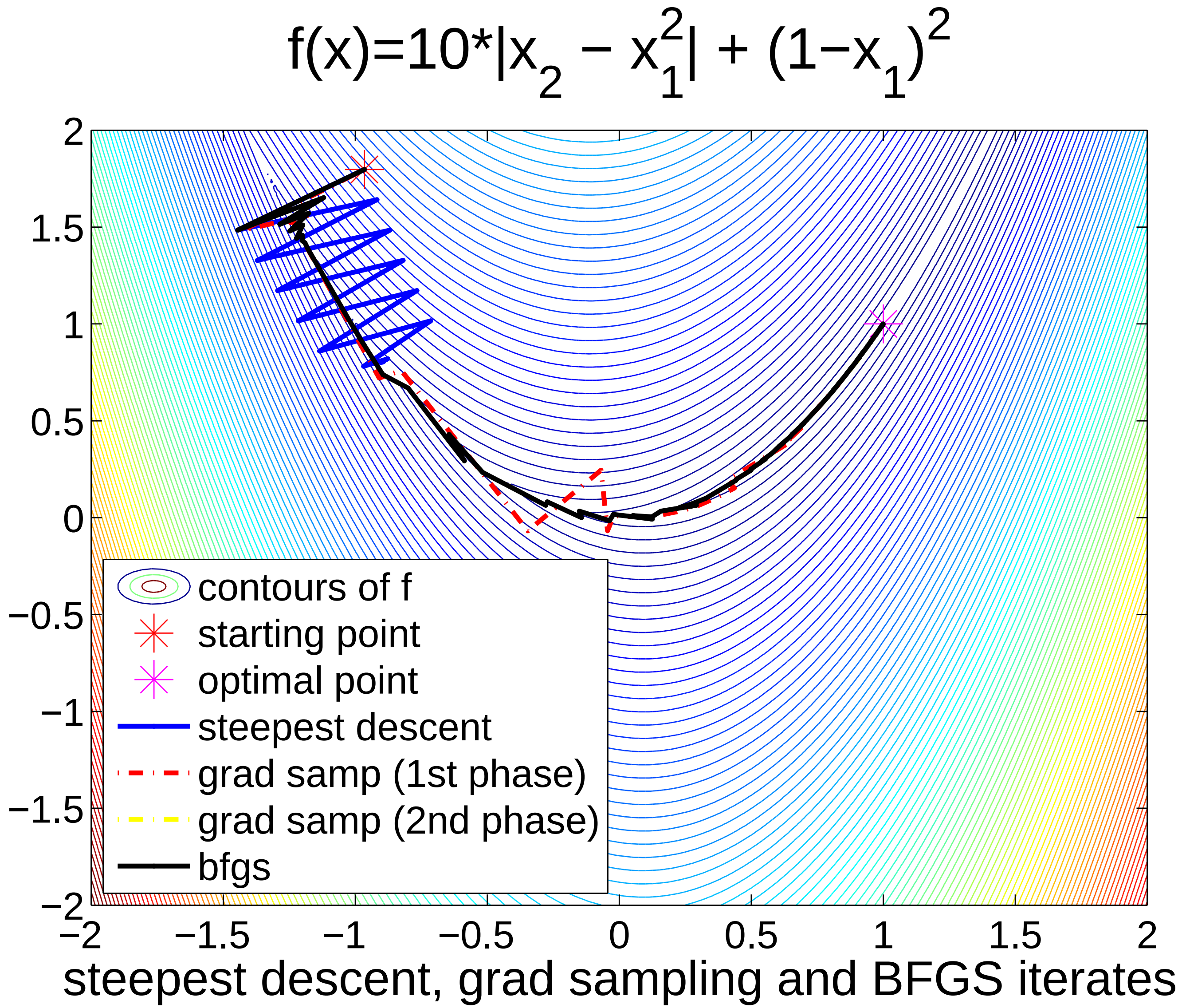
In the nonsmooth case, BFGS builds a very ill-conditioned inverse "Hessian" approximation, with some tiny eigenvalues converging to zero, corresponding to "infinitely large" curvature in the directions defined by the associated eigenvectors.

Remarkably, the condition number of the inverse Hessian approximation typically reaches 10^{16} before the method breaks down.

We have never seen convergence to non-stationary points that cannot be explained by numerical difficulties.

Convergence rate of BFGS is typically linear (not superlinear) in the nonsmooth case.

With BFGS


f(x)=10*|x_2 − x_1^2| + (1−x_1)^2
contours of f
starting point
optimal point
steepest descent
grad samp (1st phase)
grad samp (2nd phase)
bfgs
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
steepest descent, grad sampling and BFGS iterates

Example: Minimizing a Product of Eigenvalues

Let S^N denote the space of real symmetric $N \times N$ matrices, and

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

denote the eigenvalues of $X \in S^N$.

Example: Minimizing a Product of Eigenvalues

Let S^N denote the space of real symmetric $N \times N$ matrices, and

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

denote the eigenvalues of $X \in S^N$. We wish to minimize

$$f(X) = \log \prod_{i=1}^{N/2} \lambda_i(A \circ X)$$

where $A \in S^N$ is fixed and $\circ$ is the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1.

Example: Minimizing a Product of Eigenvalues

Let S^N denote the space of real symmetric $N \times N$ matrices, and

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

denote the eigenvalues of $X \in S^N$. We wish to minimize

$$f(X) = \log \prod_{i=1}^{N/2} \lambda_i(A \circ X)$$

where $A \in S^N$ is fixed and $\circ$ is the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1.

If we replace $\prod$ by $\sum$ we would have a semidefinite program.

Example: Minimizing a Product of Eigenvalues

Let S^N denote the space of real symmetric $N \times N$ matrices, and

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

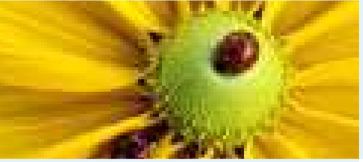
denote the eigenvalues of $X \in S^N$. We wish to minimize

$$f(X) = \log \prod_{i=1}^{N/2} \lambda_i(A \circ X)$$

where $A \in S^N$ is fixed and $\circ$ is the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1.

If we replace $\prod$ by $\sum$ we would have a semidefinite program.

Since f is not convex, may as well replace X by YY^T where $Y \in \mathbb{R}^{N \times N}$: eliminates psd constraint, and then also easy to eliminate diagonal constraint.



Example: Minimizing a Product of Eigenvalues

Let S^N denote the space of real symmetric $N \times N$ matrices, and

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

denote the eigenvalues of $X \in S^N$. We wish to minimize

$$f(X) = \log \prod_{i=1}^{N/2} \lambda_i(A \circ X)$$

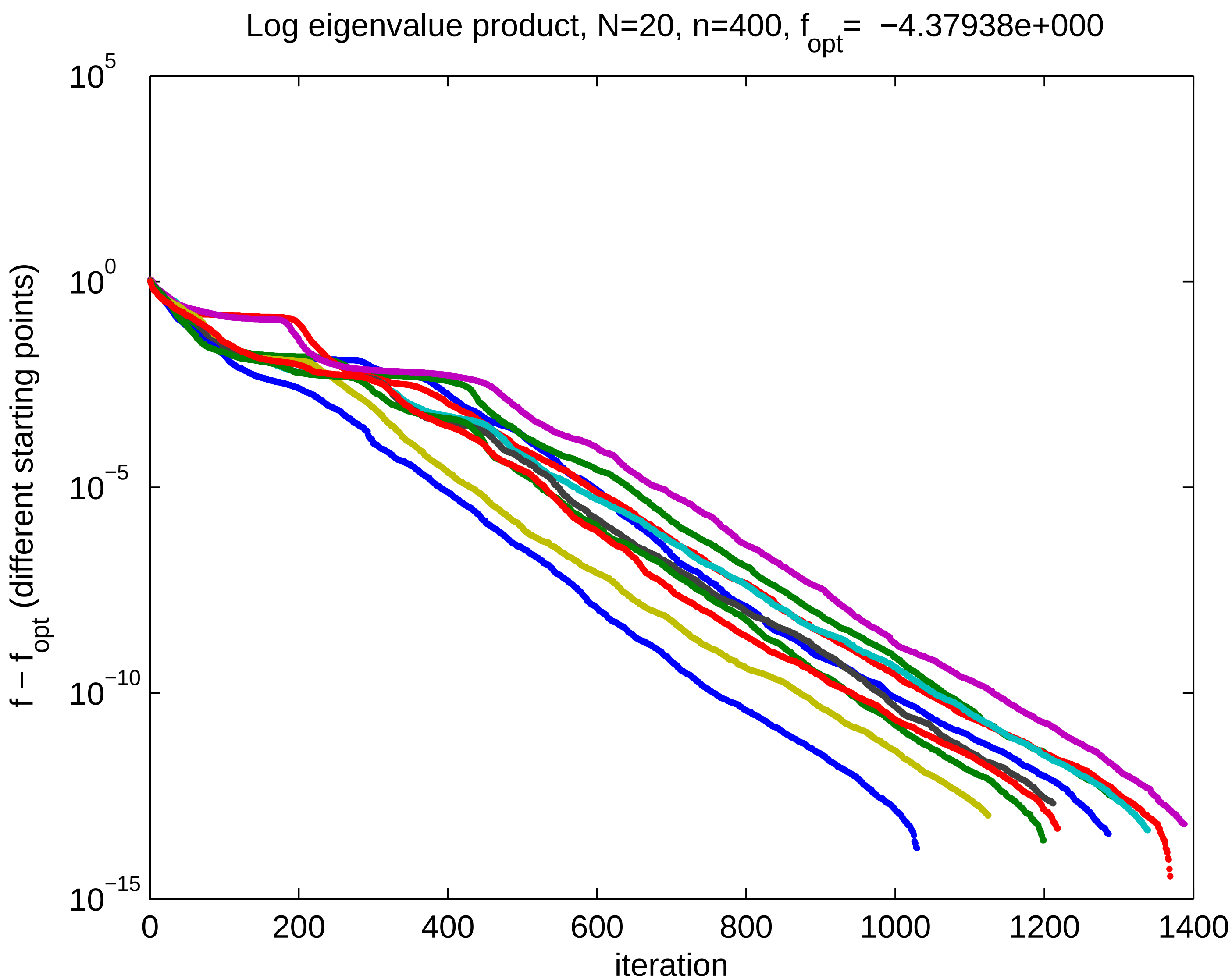
where $A \in S^N$ is fixed and $\circ$ is the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1.

If we replace $\prod$ by $\sum$ we would have a semidefinite program.

Since f is not convex, may as well replace X by YY^T where $Y \in \mathbb{R}^{N \times N}$: eliminates psd constraint, and then also easy to eliminate diagonal constraint.

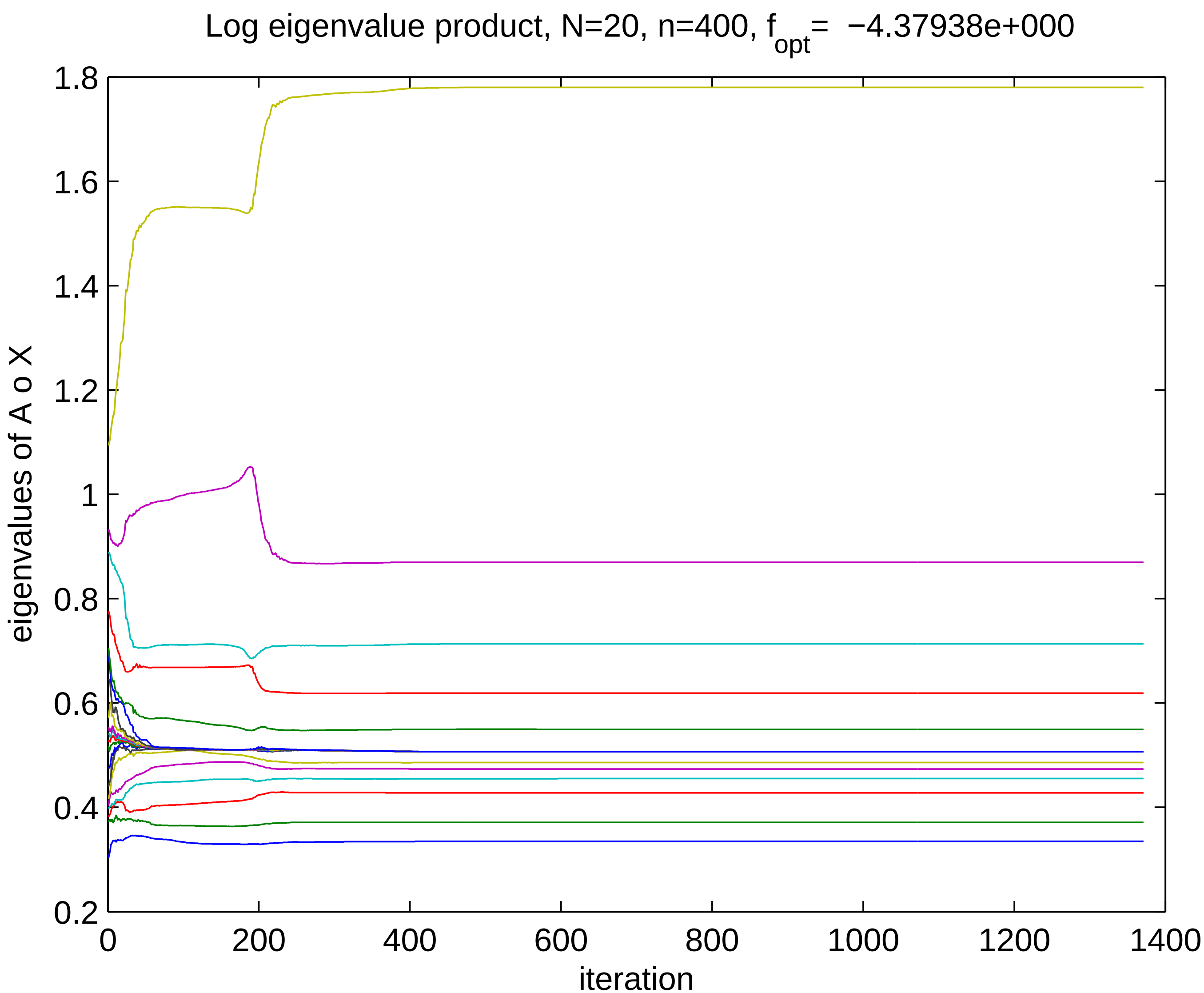
Application: entropy minimization in an environmental application (K.M. Anstreicher and J. Lee, 2004)

BFGS from 10 Randomly Generated Starting Points

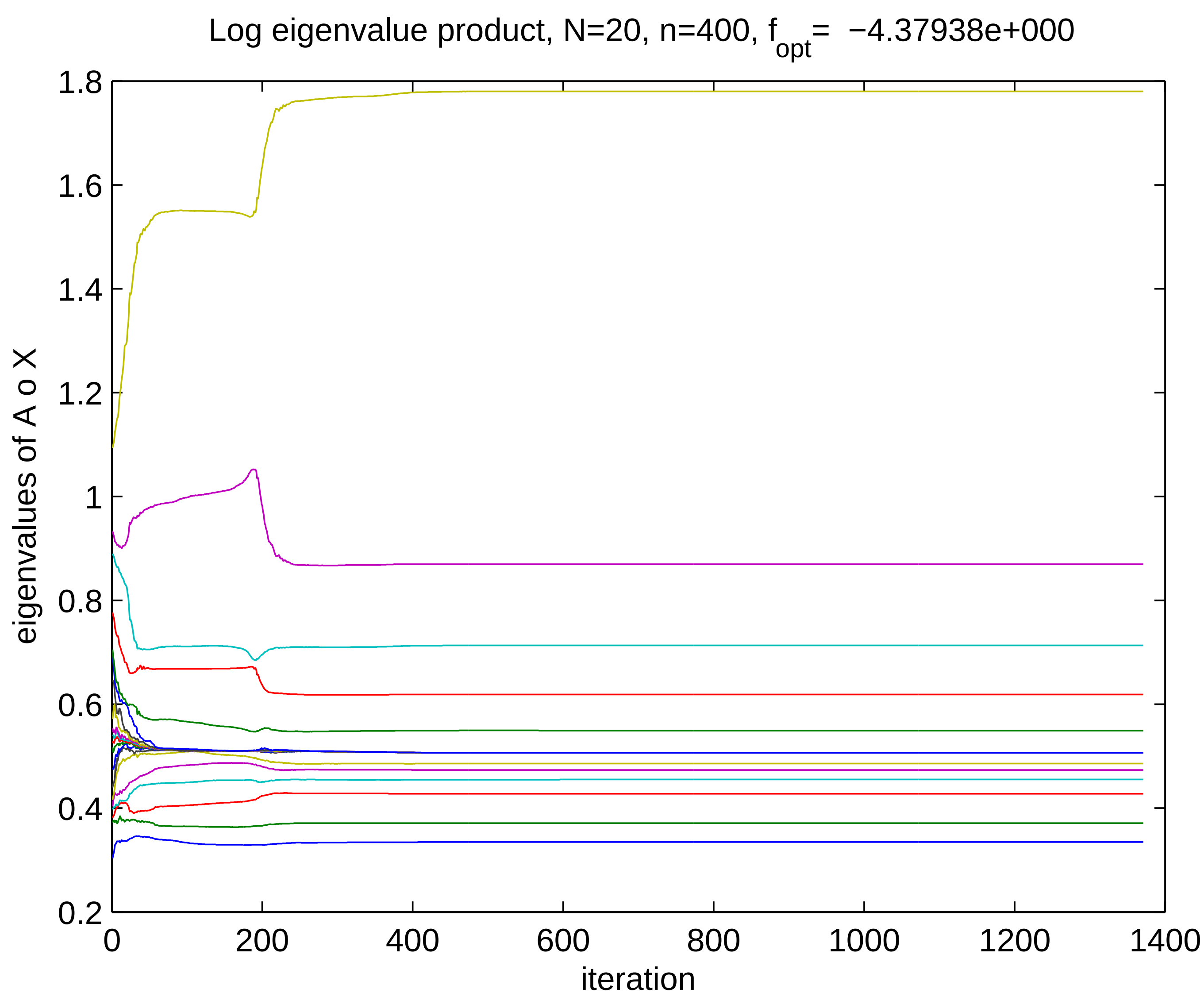


$f - f_{\text{opt}}$, where f_{opt} is least value of f found over all runs

Evolution of Eigenvalues of $A \circ X$

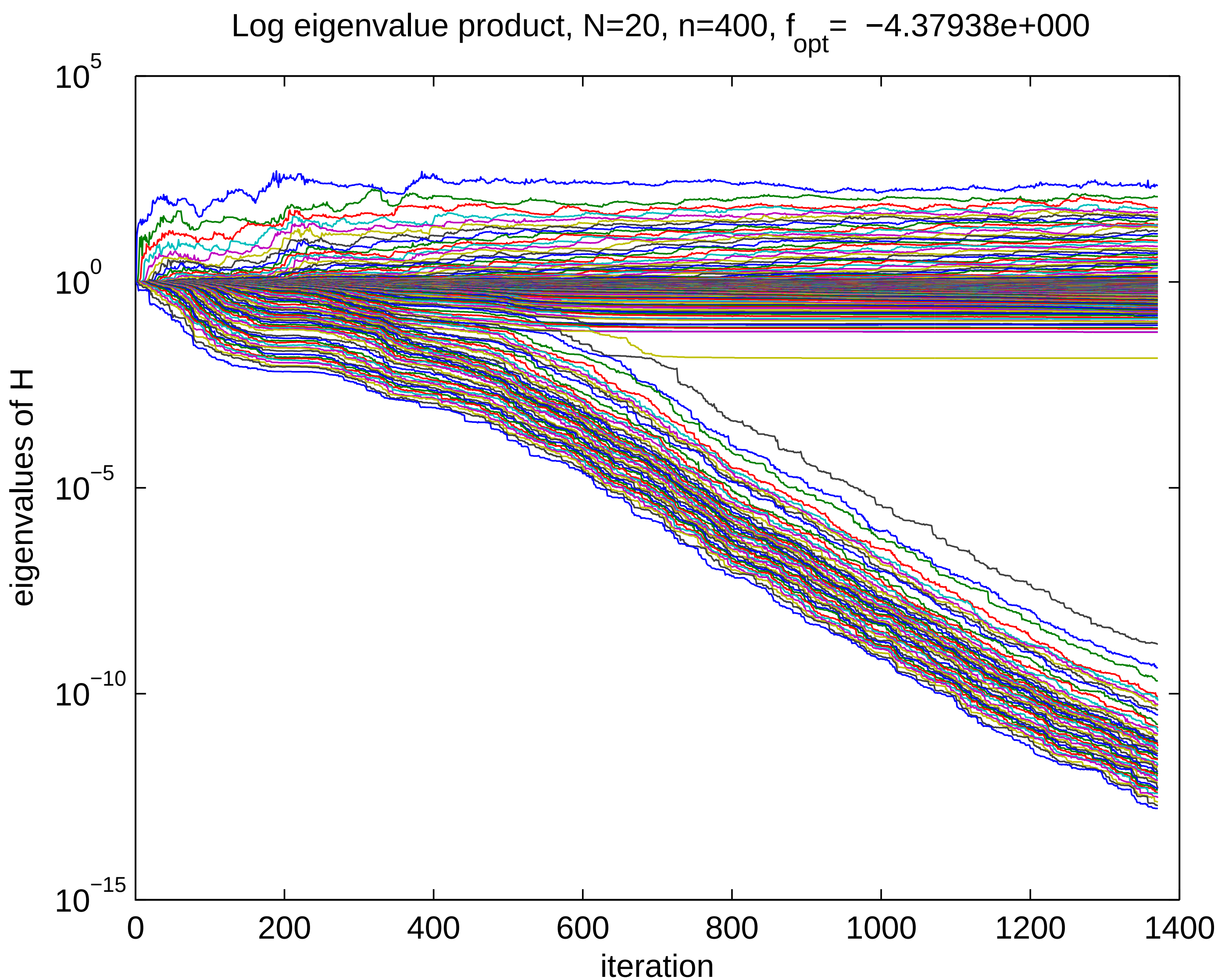


Evolution of Eigenvalues of $A \circ X$

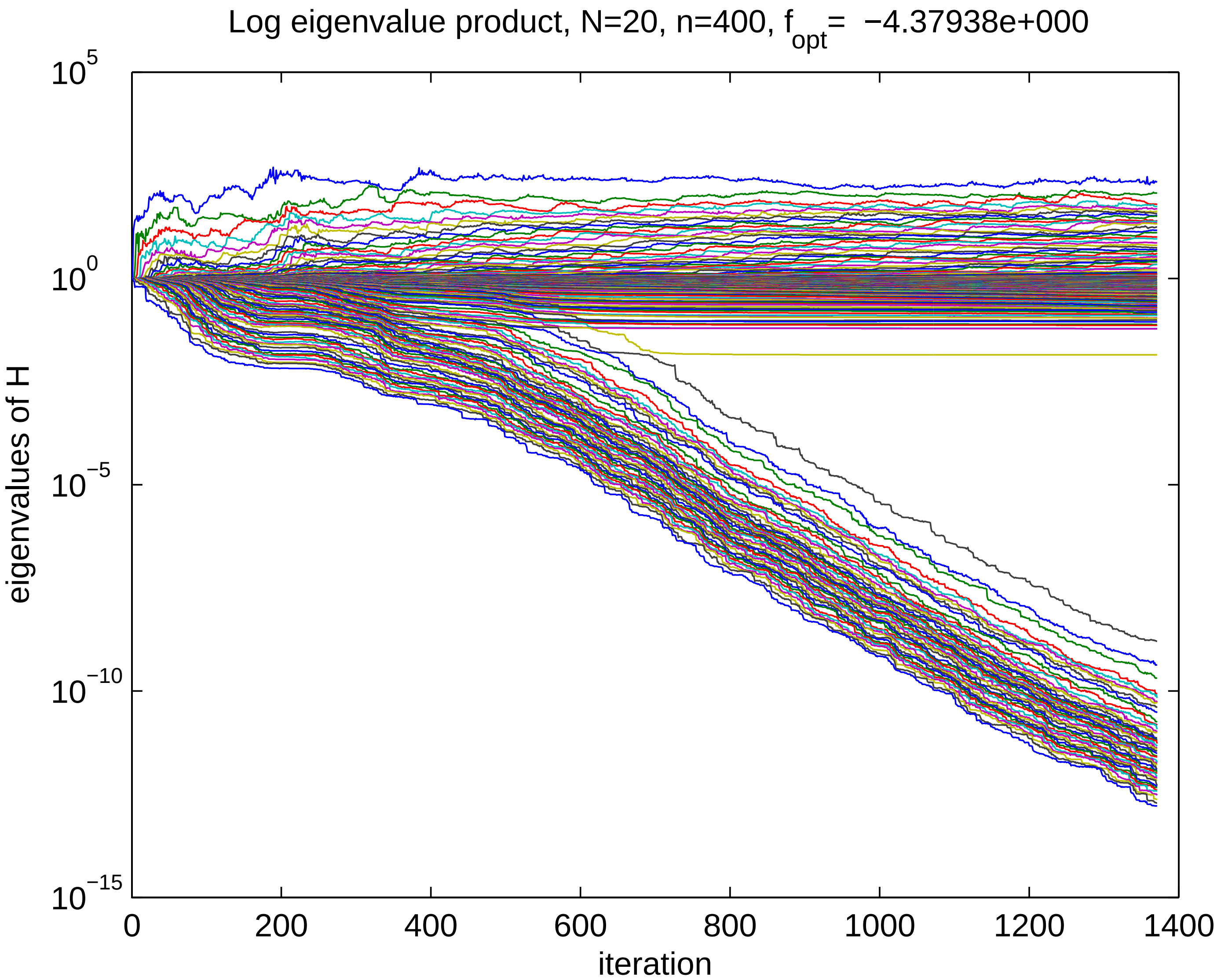


Note that $\lambda_6(X), \ldots, \lambda_{14}(X)$ coalesce

Evolution of Eigenvalues of H



Evolution of Eigenvalues of H



44 eigenvalues of H converge to zero...why???

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d.

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d.

In this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x, d) \geq 0$ for all directions d.

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d.

In this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x, d) \geq 0$ for all directions d.

- All convex functions are regular

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d.

In this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x, d) \geq 0$ for all directions d.

- All convex functions are regular
- All smooth functions are regular

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x; d)$ is upper semicontinuous there for every fixed direction d.

In this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x, d) \geq 0$ for all directions d.

- All convex functions are regular
- All smooth functions are regular
- Nonsmooth concave functions are not regular

  Example: $f(x) = -|x|$

Regularity

A locally Lipschitz, directionally differentiable function f is *regular* (Clarke 1970s) near a point x when its directional derivative $x \mapsto f'(x;d)$ is upper semicontinuous there for every fixed direction d.

In this case $0 \in \partial f(x)$ is equivalent to the first-order optimality condition $f'(x,d) \geq 0$ for all directions d.

- All convex functions are regular
- All smooth functions are regular
- Nonsmooth concave functions are not regular

  Example: $f(x) = -|x|$

Note: this is simpler than the definition of regularity *at a point* given in last week's lecture.

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

- its restriction to $\mathcal{M}$ is twice continuously differentiable near x

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

- its restriction to $\mathcal{M}$ is twice continuously differentiable near x
- the Clarke subdifferential ∂f is continuous on $\mathcal{M}$ near x

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

- its restriction to $\mathcal{M}$ is twice continuously differentiable near x
- the Clarke subdifferential ∂f is continuous on $\mathcal{M}$ near x
- par $\partial f(x)$, the subspace parallel to the affine hull of the subdifferential of f at x, is exactly the subspace normal to $\mathcal{M}$ at x.

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

- its restriction to $\mathcal{M}$ is twice continuously differentiable near x
- the Clarke subdifferential ∂f is continuous on $\mathcal{M}$ near x
- $\operatorname{par} \partial f(x)$, the subspace parallel to the affine hull of the subdifferential of f at x, is exactly the subspace normal to $\mathcal{M}$ at x.

We refer to $\operatorname{par} \partial f(x)$ as the *V-space* for f at x (with respect to $\mathcal{M}$), and to its orthogonal complement, the subspace tangent to $\mathcal{M}$ at x, as the *U-space* for f at x.

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

- its restriction to $\mathcal{M}$ is twice continuously differentiable near x
- the Clarke subdifferential ∂f is continuous on $\mathcal{M}$ near x
- par $\partial f(x)$, the subspace parallel to the affine hull of the subdifferential of f at x, is exactly the subspace normal to $\mathcal{M}$ at x.

We refer to par $\partial f(x)$ as the *V-space* for f at x (with respect to $\mathcal{M}$), and to its orthogonal complement, the subspace tangent to $\mathcal{M}$ at x, as the *U-space* for f at x.

When we refer to the V-space and U-space without reference to a point x, we mean at a minimizer.

Partly Smooth Functions

A regular function f is *partly smooth* at x relative to a manifold $\mathcal{M}$ containing x (A.S. Lewis 2003) if

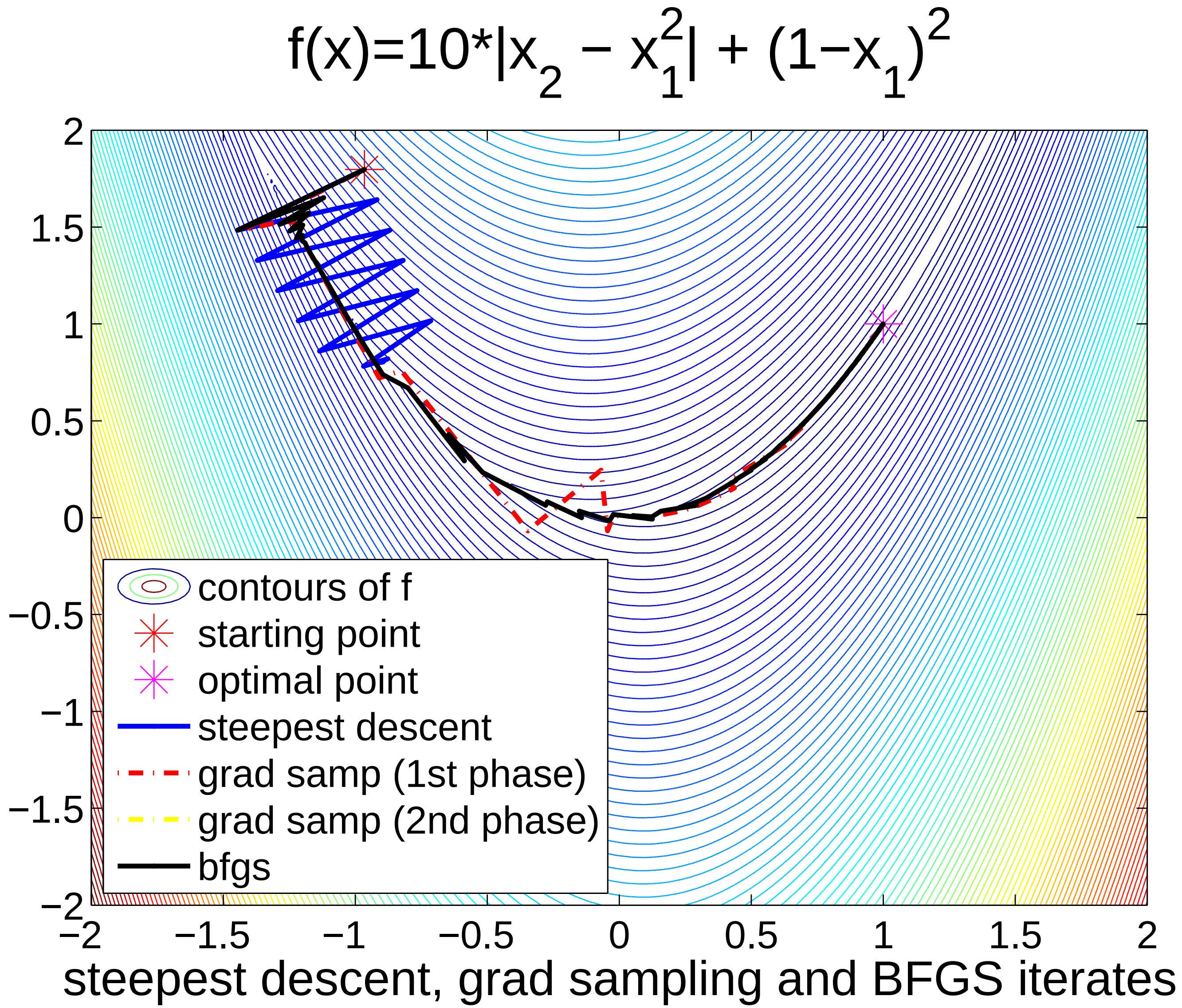
- its restriction to $\mathcal{M}$ is twice continuously differentiable near x
- the Clarke subdifferential ∂f is continuous on $\mathcal{M}$ near x
- $\operatorname{par} \partial f(x)$, the subspace parallel to the affine hull of the subdifferential of f at x, is exactly the subspace normal to $\mathcal{M}$ at x.

We refer to $\operatorname{par} \partial f(x)$ as the *V-space* for f at x (with respect to $\mathcal{M}$), and to its orthogonal complement, the subspace tangent to $\mathcal{M}$ at x, as the *U-space* for f at x.

When we refer to the V-space and U-space without reference to a point x, we mean at a minimizer.

For nonzero y in the V-space, the mapping $t \mapsto f(x + ty)$ is necessarily nonsmooth at $t = 0$, while for nonzero y in the U-space, $t \mapsto f(x + ty)$ is differentiable at $t = 0$ as long as f is locally Lipschitz.

Same Example Again



Relation of Partial Smoothness to Earlier Work

Partial smoothness is closely related to earlier work of J.V. Burke and J.J. Moré (1990,1994) and S.J. Wright (1993) on identification of constraint structure by algorithms.

Relation of Partial Smoothness to Earlier Work

Partial smoothness is closely related to earlier work of J.V. Burke and J.J. Moré (1990,1994) and S.J. Wright (1993) on identification of constraint structure by algorithms.

When f is convex, the partly smooth nomenclature is consistent with the usage of V-space and U-space by C. Lemaréchal, F. Oustry and C. Sagastizábal (2000), but partial smoothness does not imply convexity and convexity does not imply partial smoothness.

Why Did 44 Eigenvalues of H Converge to Zero?

The eigenvalue product is *partly smooth* with respect to the manifold of matrices with an eigenvalue with given multiplicity.

Why Did 44 Eigenvalues of H Converge to Zero?

The eigenvalue product is *partly smooth* with respect to the manifold of matrices with an eigenvalue with given multiplicity.

Recall that at the computed minimizer,

$$\lambda_6(A \circ X) \approx \ldots \approx \lambda_{14}(A \circ X).$$

Matrix theory says that imposing multiplicity m on an eigenvalue a matrix $\in S^N$ is $\frac{m(m+1)}{2} - 1$ conditions, or 44 when $m = 9$, so the dimension of the V-space at this minimizer is 44.

Why Did 44 Eigenvalues of H Converge to Zero?

The eigenvalue product is *partly smooth* with respect to the manifold of matrices with an eigenvalue with given multiplicity.

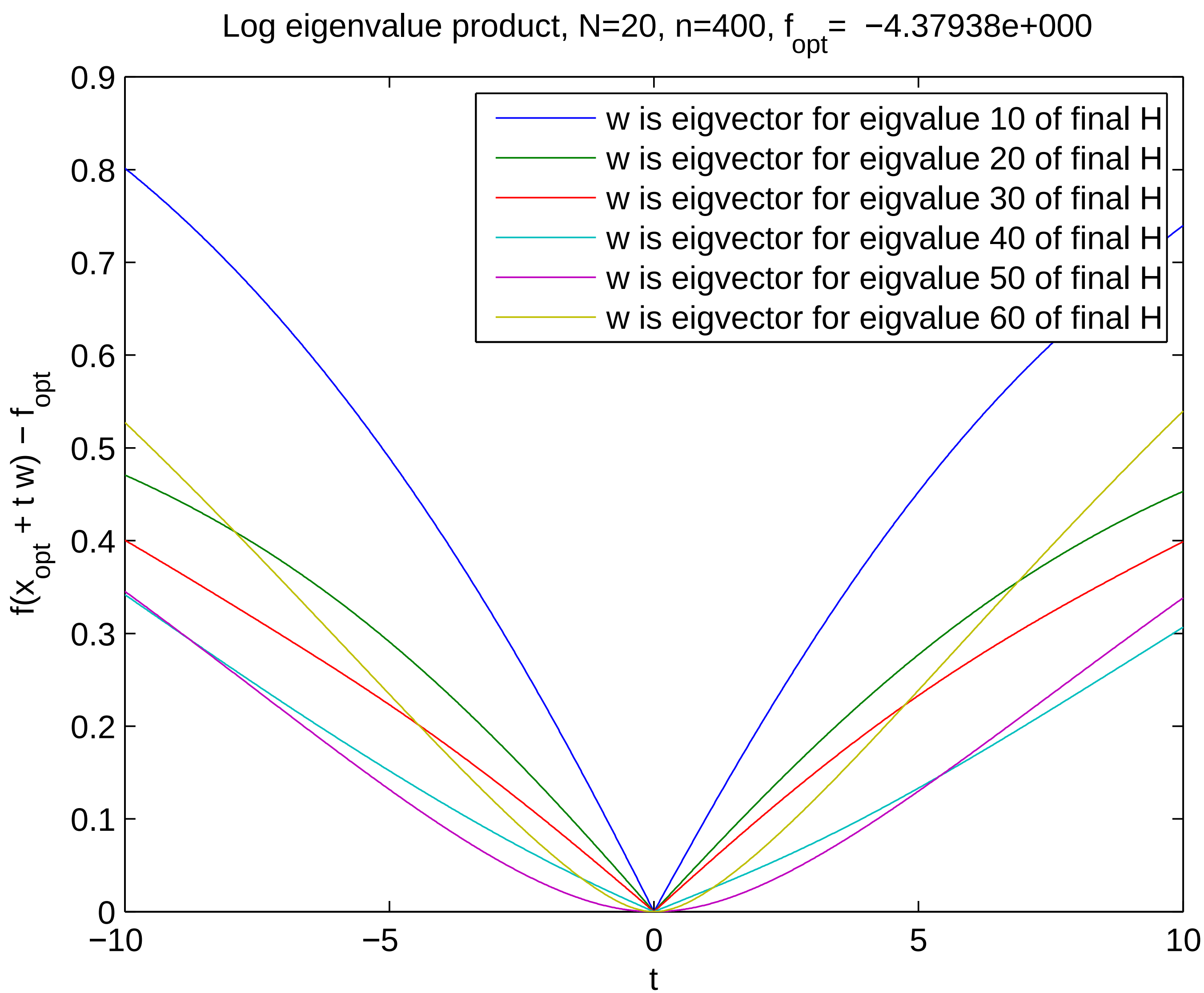
Recall that at the computed minimizer,

$$\lambda_6(A \circ X) \approx \ldots \approx \lambda_{14}(A \circ X).$$

Matrix theory says that imposing multiplicity m on an eigenvalue a matrix $\in S^N$ is $\frac{m(m+1)}{2} - 1$ conditions, or 44 when $m = 9$, so the dimension of the V-space at this minimizer is 44.

Thus BFGS *automatically* detected the U and V space partitioning without knowing anything about the mathematical structure of f!

Variation of f from Minimizer, along EigVecs of H



Eigenvalues of H numbered *smallest to largest*

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Prove or disprove that the following hold with probability one:

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Prove or disprove that the following hold with probability one:

1. BFGS generates an infinite sequence $\{x\}$ with f differentiable at all iterates

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Prove or disprove that the following hold with probability one:

1. BFGS generates an infinite sequence $\{x\}$ with f differentiable at all iterates
2. Any cluster point $\bar{x}$ is Clarke stationary

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Prove or disprove that the following hold with probability one:

1. BFGS generates an infinite sequence $\{x\}$ with f differentiable at all iterates
2. Any cluster point $\bar{x}$ is Clarke stationary
3. The sequence of function values generated (including all of the line search iterates) converges to $f(\bar{x})$ R-linearly

Challenge: Convergence of BFGS in Nonsmooth Case

Assume f is locally Lipschitz with bounded level sets and is semi-algebraic

Assume the initial x and H are generated randomly (e.g. from normal and Wishart distributions)

Prove or disprove that the following hold with probability one:

1. BFGS generates an infinite sequence $\{x\}$ with f differentiable at all iterates
2. Any cluster point $\bar{x}$ is Clarke stationary
3. The sequence of function values generated (including all of the line search iterates) converges to $f(\bar{x})$ R-linearly
4. If $\{x\}$ converges to $\bar{x}$ where f is "partly smooth" w.r.t. a manifold $\mathcal{M}$ then the subspace defined by the eigenvectors corresponding to eigenvalues of H converging to zero converges to the "V-space" of f w.r.t. $\mathcal{M}$ at $\bar{x}$

Extensions of BFGS for Nonsmooth Optimization

A combined BFGS-Gradient Sampling method with convergence theory (F.E. Curtis and X. Que, 2015)

A combined BFGS-Gradient Sampling method with convergence theory (F.E. Curtis and X. Que, 2015)

Constrained Problems

$$\min \; f(x)$$
$$\text{subject to } c_i(x) \leq 0, \quad i = 1, \ldots, p$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

Extensions of BFGS for Nonsmooth Optimization

A combined BFGS-Gradient Sampling method with convergence theory (F.E. Curtis and X. Que, 2015)

Constrained Problems

$$\min \ f(x)$$
$$\text{subject to } c_i(x) \le 0, \quad i = 1, \ldots, p$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

A successive quadratic programming (SQP) BFGS method applied to challenging problems in static-output-feedback control design (F.E. Curtis, T. Mitchell and M.L.O., 2015).

Extensions of BFGS for Nonsmooth Optimization

A combined BFGS-Gradient Sampling method with convergence theory (F.E. Curtis and X. Que, 2015)

Constrained Problems

$$\begin{aligned} \min \; & f(x) \\ \text{subject to } & c_i(x) \leq 0, \quad i = 1, \ldots, p \end{aligned}$$

where f and $c_1, \ldots, c_p$ are locally Lipschitz but may not be differentiable at local minimizers.

A successive quadratic programming (SQP) BFGS method applied to challenging problems in static-output-feedback control design (F.E. Curtis, T. Mitchell and M.L.O., 2015).

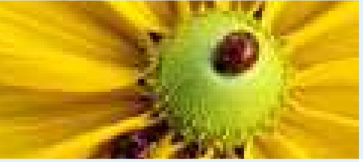
Although there are no theoretical results, it is much more efficient and effective than the SQP Gradient Sampling method which does have convergence results.

Some Difficult Examples

A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$



A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $R_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_R = \{x : x_{i+1} = x_i^2, \quad i = 1, \ldots, n-1\}$$

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $R_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_R = \{x : x_{i+1} = x_i^2, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_R$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of R_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_R$ to reach x^* (unless it “gets lucky”).

A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

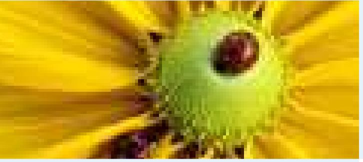
The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $R_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_R = \{x : x_{i+1} = x_i^2, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_R$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of R_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_R$ to reach x^* (unless it "gets lucky").

When $p = 2$: R_2 is smooth but not convex. Starting at $\hat{x}$:



A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $R_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_R = \{x : x_{i+1} = x_i^2, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_R$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of R_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_R$ to reach x^* (unless it "gets lucky").

When $p = 2$: R_2 is smooth but not convex. Starting at $\hat{x}$:

- $n = 5$: BFGS needs 43 iterations to reduce R_2 below 10^{-15}

A Rosenbrock Function

Consider a generalization of the Rosenbrock (1960) function:

$$R_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|^p, \quad \text{where } p > 0.$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $R_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $R_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_R = \{x : x_{i+1} = x_i^2, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_R$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of R_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_R$ to reach x^* (unless it "gets lucky").

When $p = 2$: R_2 is smooth but not convex. Starting at $\hat{x}$:

- $n = 5$: BFGS needs 43 iterations to reduce R_2 below 10^{-15}
- $n = 10$, BFGS needs 276 iterations to reduce R_2 below 10^{-15}.

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

However, R_1 is regular at $x \in \mathcal{M}_R$ and partly smooth at x w.r.t. $\mathcal{M}_R$.

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

However, R_1 is regular at $x \in \mathcal{M}_R$ and partly smooth at x w.r.t. $\mathcal{M}_R$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

However, R_1 is regular at $x \in \mathcal{M}_R$ and partly smooth at x w.r.t. $\mathcal{M}_R$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces R_1 only to about 1×10^{-3} in 1000 iterations

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

However, R_1 is regular at $x \in \mathcal{M}_R$ and partly smooth at x w.r.t. $\mathcal{M}_R$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces R_1 only to about 1×10^{-3} in 1000 iterations
- $n = 10$: BFGS reduces R_1 only to about 7×10^{-4} in 1000 iterations

The Nonsmooth Variant of the Rosenbrock Function

$$R_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - x_i^2|$$

R_1 is nonsmooth (but locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_R$, but R_1 is not differentiable on $\mathcal{M}_R$.

However, R_1 is regular at $x \in \mathcal{M}_R$ and partly smooth at x w.r.t. $\mathcal{M}_R$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces R_1 only to about 1×10^{-3} in 1000 iterations
- $n = 10$: BFGS reduces R_1 only to about 7×10^{-4} in 1000 iterations

Again the method appears to be converging, very slowly, but may be having numerical difficulties.

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

So $T_2(x) = 2x^2 - 1$,

An Aside: Chebyshev Polynomials

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

So $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3$, etc.

An Aside: Chebyshev Polynomials

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

So $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3$, etc.

Important properties that can be proved easily include

An Aside: Chebyshev Polynomials

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

So $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3$, etc.

Important properties that can be proved easily include

- $T_n(x) = \cos(n \cos^{-1}(x))$

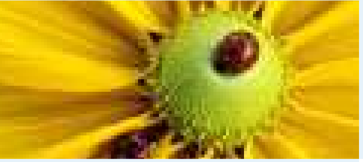
A sequence of orthogonal polynomials defined on $[-1, 1]$ by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

So $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3$, etc.

Important properties that can be proved easily include

- $T_n(x) = \cos(n \cos^{-1}(x))$
- $T_m(T_n(x)) = T_{mn}(x)$



An Aside: Chebyshev Polynomials

A sequence of orthogonal polynomials defined on $[-1, 1]$ by

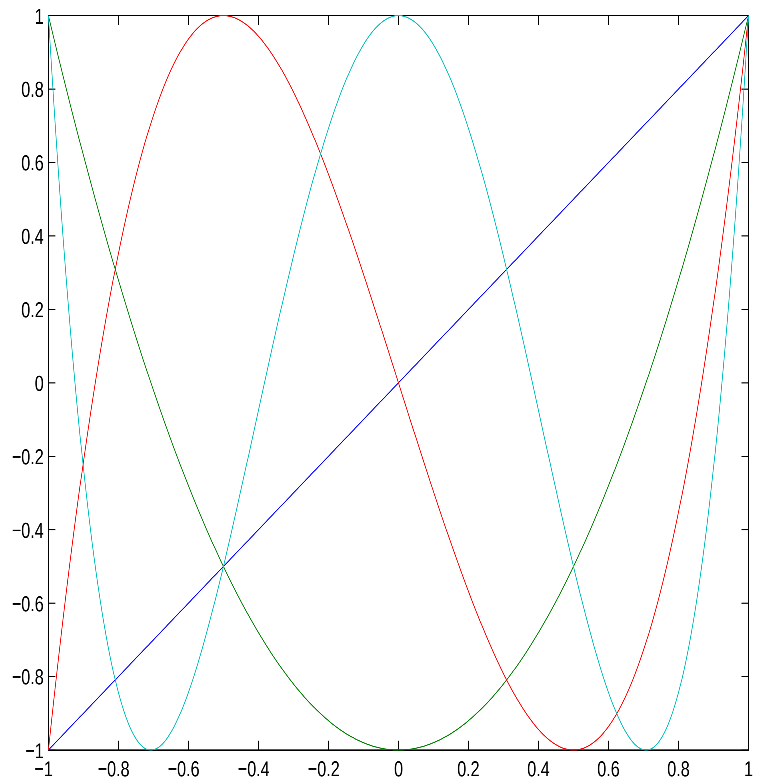
$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

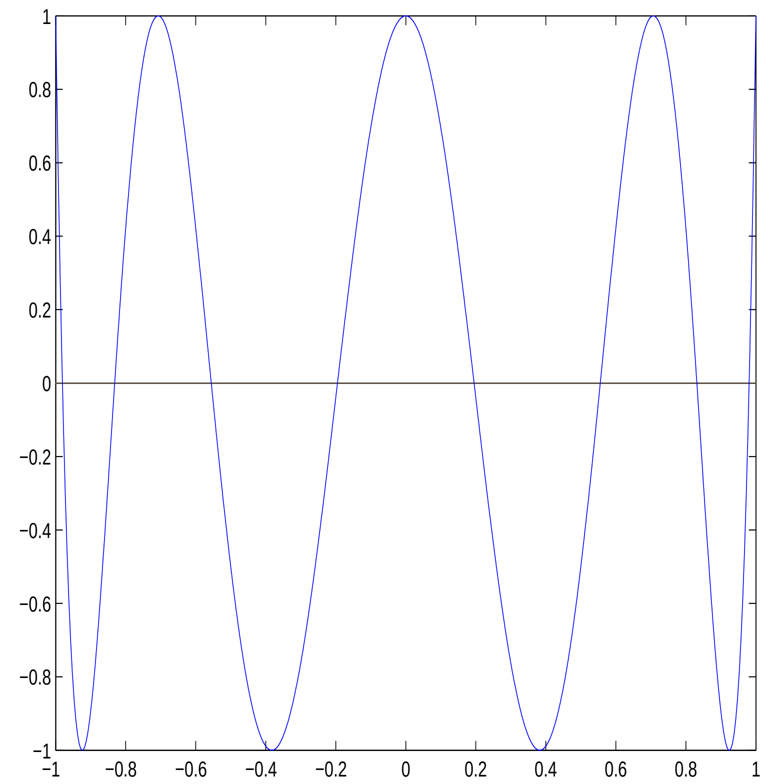
So $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3$, etc.

Important properties that can be proved easily include

- $T_n(x) = \cos(n \cos^{-1}(x))$
- $T_m(T_n(x)) = T_{mn}(x)$
- $\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_i(x) T_j(x) dx = 0$ if $i \neq j$

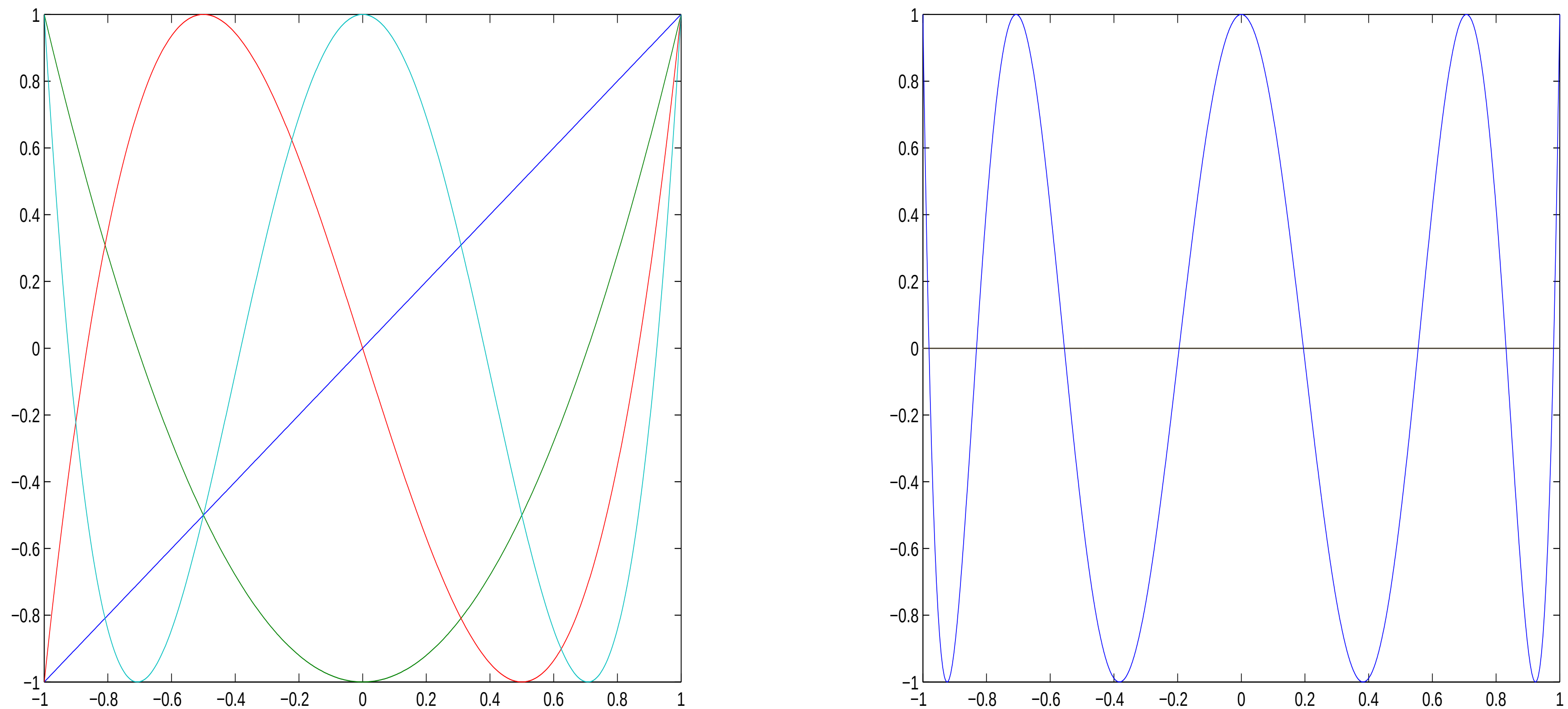
Plots of Chebyshev Polynomials





Left: Plots of $T_0(x), \ldots, T_4(x)$ Right: Plot of $T_8(x)$.

Plots of Chebyshev Polynomials



Left: Plots of $T_0(x), \ldots, T_4(x)$ Right: Plot of $T_8(x)$.

Question: How many extrema does $T_n(x)$ have in $[-1, 1]$?

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $N_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_N = \{x : x_{i+1} = 2x_i^2 - 1, \quad i = 1, \ldots, n-1\}$$

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $N_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_N = \{x : x_{i+1} = 2x_i^2 - 1, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_N$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of N_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_N$ to reach x^* (unless it "gets lucky").

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $N_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_N = \{x : x_{i+1} = 2x_i^2 - 1, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_N$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of N_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_N$ to reach x^* (unless it "gets lucky").

When $p = 2$: N_2 is smooth but not convex. Starting at $\hat{x}$:

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $N_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_N = \{x : x_{i+1} = 2x_i^2 - 1, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_N$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of N_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_N$ to reach x^* (unless it "gets lucky").

When $p = 2$: N_2 is smooth but not convex. Starting at $\hat{x}$:

- $n = 5$: BFGS needs 370 iterations to reduce N_2 below 10^{-15}

Nesterov's Chebyshev-Rosenbrock Functions

Consider the function

$$N_p(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|^p, \quad \text{where } p > 0$$

The unique minimizer is $x^* = [1, 1, \ldots, 1]^T$ with $N_p(x^*) = 0$.

Define $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ with $N_p(\hat{x}) = 1$ and the manifold

$$\mathcal{M}_N = \{x : x_{i+1} = 2x_i^2 - 1, \quad i = 1, \ldots, n-1\}$$

For $x \in \mathcal{M}_N$, e.g. $x = x^*$ or $x = \hat{x}$, the 2nd term of N_p is zero. Starting at $\hat{x}$, BFGS needs to approximately follow $\mathcal{M}_N$ to reach x^* (unless it "gets lucky").

When $p = 2$: N_2 is smooth but not convex. Starting at $\hat{x}$:

- $n = 5$: BFGS needs 370 iterations to reduce N_2 below 10^{-15}
- $n = 10$: needs $\sim$ 50,000 iterations to reduce N_2 below 10^{-15}

even though N_2 is *smooth*!

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$x_{i+1} = 2x_i^2 - 1 = T_2(x_i) = T_2(T_2(x_{i-1}))$$

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} &= 2x_i^2 - 1 = T_2(x_i) = T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} &= 2x_i^2 - 1 = T_2(x_i) = T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} = 2x_i^2 - 1 = T_2(x_i) &= T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} &= 2x_i^2 - 1 = T_2(x_i) = T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1
- $x_2 = 2x_1^2 - 1$ to trace the graph of $T_2(x_1)$ on $[-1, 1]$

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} = 2x_i^2 - 1 = T_2(x_i) &= T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1
- $x_2 = 2x_1^2 - 1$ to trace the graph of $T_2(x_1)$ on $[-1, 1]$
- $x_3 = T_2(T_2(x))$ to trace the graph of $T_4(x_1)$ on $[-1, 1]$

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} = 2x_i^2 - 1 &= T_2(x_i) = T_2(T_2(x_{i-1})) \\ &= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1
- $x_2 = 2x_1^2 - 1$ to trace the graph of $T_2(x_1)$ on $[-1, 1]$
- $x_3 = T_2(T_2(x))$ to trace the graph of $T_4(x_1)$ on $[-1, 1]$
- $x_n = T_{2^{n-1}}(x)$ to trace the graph of $T_{2^{n-1}}(x_1)$ on $[-1, 1]$

which has $2^{n-1} - 1$ extrema in $(-1, 1)$.

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$
\begin{aligned}
x_{i+1} = 2x_i^2 - 1 = T_2(x_i) &= T_2(T_2(x_{i-1})) \\
&= T_2(T_2(\ldots T_2(x_1)\ldots)) = T_{2^i}(x_1).
\end{aligned}
$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1
- $x_2 = 2x_1^2 - 1$ to trace the graph of $T_2(x_1)$ on $[-1, 1]$
- $x_3 = T_2(T_2(x))$ to trace the graph of $T_4(x_1)$ on $[-1, 1]$
- $x_n = T_{2^{n-1}}(x)$ to trace the graph of $T_{2^{n-1}}(x_1)$ on $[-1, 1]$

which has $2^{n-1} - 1$ extrema in $(-1, 1)$.
Even though BFGS will *not* track the manifold $\mathcal{M}_N$ exactly, it will follow it approximately. So, since the manifold is highly oscillatory, BFGS must take relatively short steps to obtain reduction in N_2 in the line search, and hence it takes *many* iterations!

Why BFGS Takes So Many Iterations to Minimize N_2

Let $T_i(x)$ denote the ith Chebyshev polynomial. For $x \in \mathcal{M}_N$,

$$\begin{aligned} x_{i+1} = 2x_i^2 - 1 = T_2(x_i) &= T_2(T_2(x_{i-1})) \\ = T_2(T_2(\ldots T_2(x_1)\ldots)) &= T_{2^i}(x_1). \end{aligned}$$

To move from $\hat{x}$ to x^* along the manifold $\mathcal{M}_N$ *exactly* requires

- x_1 to change from -1 to 1
- $x_2 = 2x_1^2 - 1$ to trace the graph of $T_2(x_1)$ on $[-1, 1]$
- $x_3 = T_2(T_2(x))$ to trace the graph of $T_4(x_1)$ on $[-1, 1]$
- $x_n = T_{2^{n-1}}(x)$ to trace the graph of $T_{2^{n-1}}(x_1)$ on $[-1, 1]$

which has $2^{n-1} - 1$ extrema in $(-1, 1)$.

Even though BFGS will *not* track the manifold $\mathcal{M}_N$ exactly, it will follow it approximately. So, since the manifold is highly oscillatory, BFGS must take relatively short steps to obtain reduction in N_2 in the line search, and hence it takes *many* iterations! At the very end, since N_2 is smooth, BFGS is superlinearly convergent!

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

However, N_1 is regular at $x \in \mathcal{M}_N$ and partly smooth at x w.r.t. $\mathcal{M}_N$.

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

However, N_1 is regular at $x \in \mathcal{M}_N$ and partly smooth at x w.r.t. $\mathcal{M}_N$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

However, N_1 is regular at $x \in \mathcal{M}_N$ and partly smooth at x w.r.t. $\mathcal{M}_N$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces N_1 only to about 5×10^{-3} in 1000 iterations

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

However, N_1 is regular at $x \in \mathcal{M}_N$ and partly smooth at x w.r.t. $\mathcal{M}_N$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces N_1 only to about 5×10^{-3} in 1000 iterations
- $n = 10$: BFGS reduces N_1 only to about 2×10^{-2} in 1000 iterations

First Nonsmooth Variant of Nesterov's Function

$$N_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|$$

N_1 is nonsmooth (though locally Lipschitz) as well as nonconvex. The second term is still zero on the manifold $\mathcal{M}_N$, but N_1 is not differentiable on $\mathcal{M}_N$.

However, N_1 is regular at $x \in \mathcal{M}_N$ and partly smooth at x w.r.t. $\mathcal{M}_N$.

We cannot initialize BFGS at $\hat{x}$, so starting at normally distributed random points:

- $n = 5$: BFGS reduces N_1 only to about 5×10^{-3} in 1000 iterations
- $n = 10$: BFGS reduces N_1 only to about 2×10^{-2} in 1000 iterations

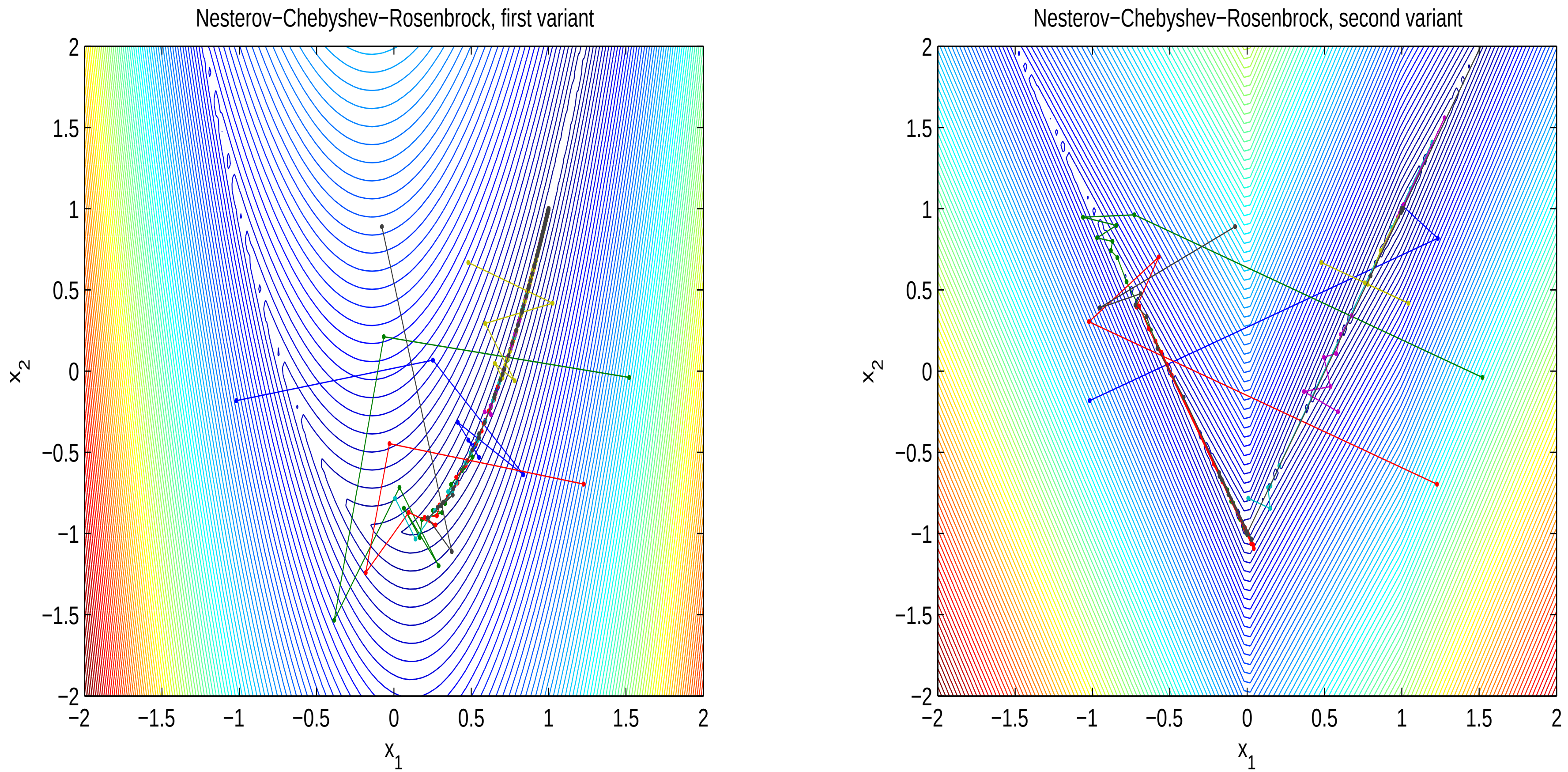
The method appears to be converging, very slowly, but may be having numerical difficulties.

$$\widehat{N}_1(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|.$$

Again, the unique global minimizer is x^*. The second term is zero on the set

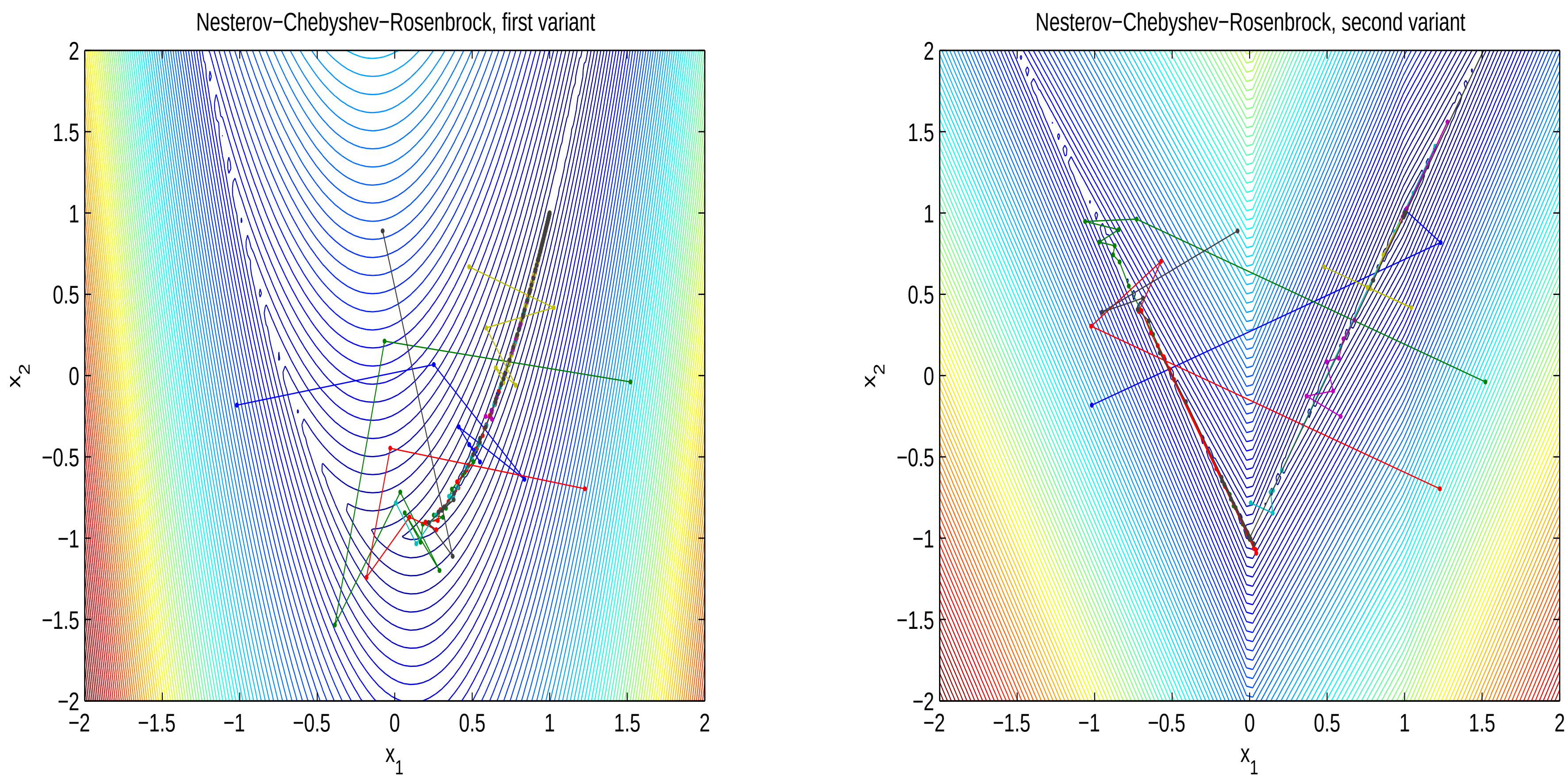
$$S = \{x : x_{i+1} = 2|x_i| - 1, \quad i = 1, \ldots, n-1\}$$

but S is not a manifold: it has "corners".



Contour plots of nonsmooth Chebyshev-Rosenbrock functions N_1 (left) and $\widehat{N}_1$ (right), with $n = 2$, with iterates generated by BFGS initialized at 7 different randomly generated points.

Contour Plots of the Nonsmooth Variants for $n = 2$



Contour plots of nonsmooth Chebyshev-Rosenbrock functions N_1 (left) and $\widehat{N}_1$ (right), with $n = 2$, with iterates generated by BFGS initialized at 7 different randomly generated points.

On the left, always get convergence to $x^* = [1, 1]^T$. On the right, most runs converge to $[1, 1]$ but some go to $x = [0, -1]^T$.

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

In fact, for $n \geq 2$:

- $\widehat{N}_1$ has 2^{n-1} Clarke stationary points

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

In fact, for $n \geq 2$:

- $\widehat{N}_1$ has 2^{n-1} Clarke stationary points
- the only local minimizer is the global minimizer x^*

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

In fact, for $n \geq 2$:

- $\widehat{N}_1$ has 2^{n-1} Clarke stationary points
- the only local minimizer is the global minimizer x^*
- x^* is the only stationary point in the sense of Mordukhovich (i.e., with $0 \in \partial \widehat{N}_1(x)$ where ∂ is defined in Rockafellar and Wets, *Variational Analysis*, 1998).

(M. Gürbüzbalaban and M.L.O., 2012).

Properties of the Second Nonsmooth Variant $\widehat{N}_1$

When $n = 2$, the point $x = [0, -1]^T$ is Clarke stationary for the second nonsmooth variant $\widehat{N}_1$. We can see this because zero is in the convex hull of the gradient limits for $\widehat{N}_1$ at the point x.

However, $x = [0, -1]^T$ is not a local minimizer, because $d = [1, 2]^T$ is a direction of linear descent: $\widehat{N}_1'(x, d) < 0$.

These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

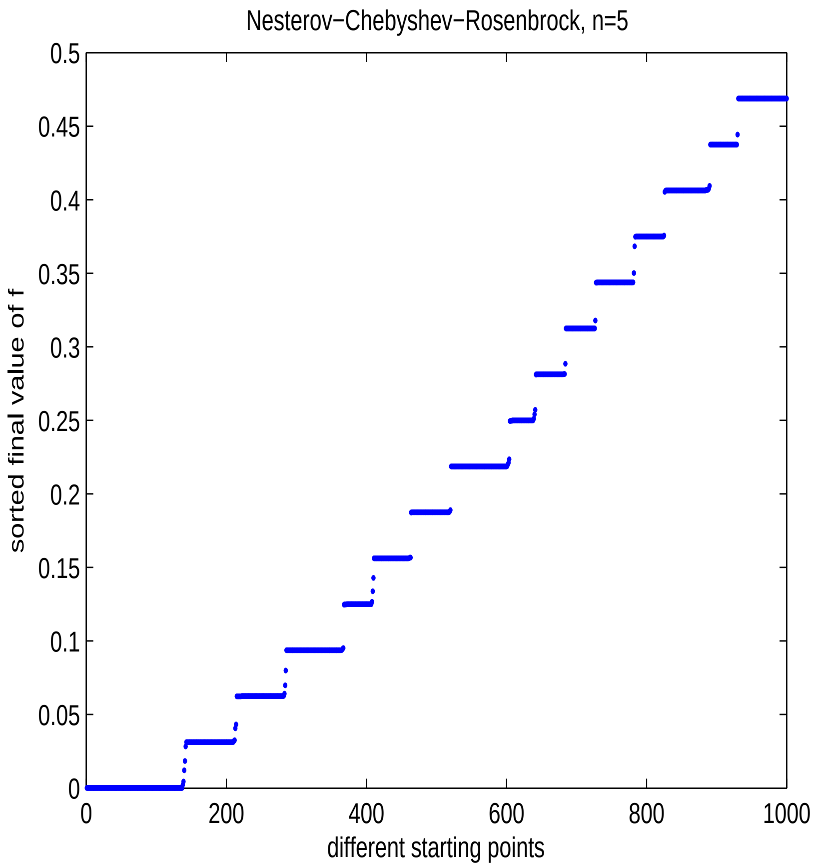
In fact, for $n \geq 2$:

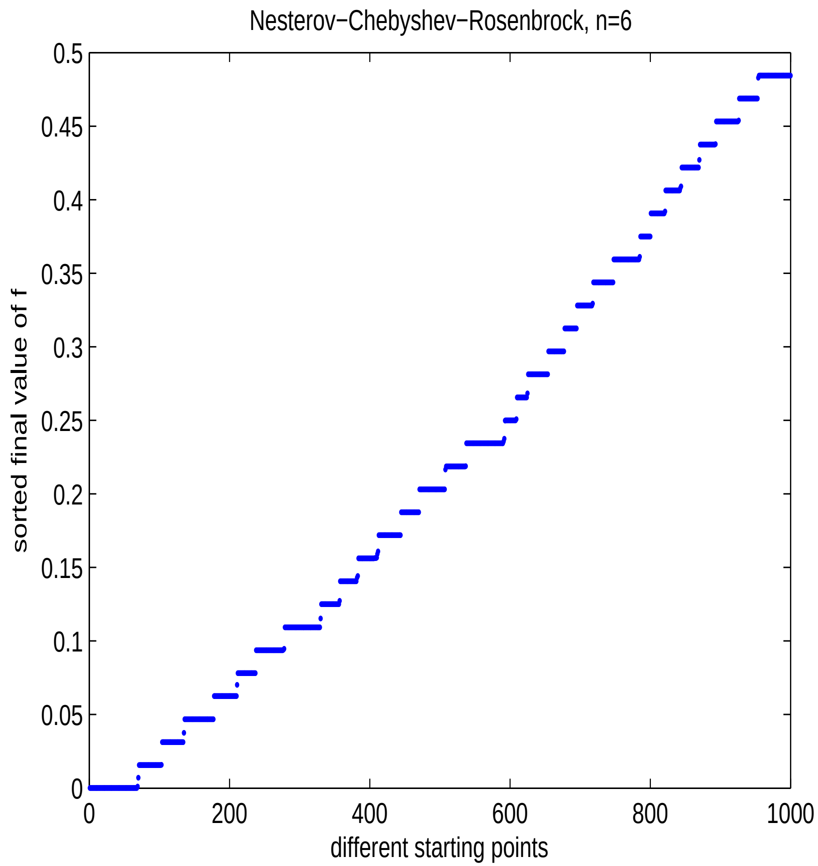
- $\widehat{N}_1$ has 2^{n-1} Clarke stationary points
- the only local minimizer is the global minimizer x^*
- x^* is the only stationary point in the sense of Mordukhovich (i.e., with $0 \in \partial \widehat{N}_1(x)$ where ∂ is defined in Rockafellar and Wets, *Variational Analysis*, 1998).

(M. Gürbüzbalaban and M.L.O., 2012).

Furthermore, starting from enough randomly generated starting points, BFGS finds all 2^{n-1} Clarke stationary points!

Behavior of BFGS on the Second Nonsmooth Variant





Left: *sorted* final values of $\widehat{N}_1$ for 1000 randomly generated starting points, when $n = 5$: BFGS finds all 16 Clarke stationary points. Right: same with $n = 6$: BFGS finds all 32 Clarke stationary points.

Convergence to Non-Locally-Minimizing Points

When f is *smooth*, convergence of methods such as BFGS to non-locally-minimizing stationary points or local maxima is *possible* but not likely, because of the line search, and such convergence will not be stable under perturbation.

Convergence to Non-Locally-Minimizing Points

When f is *smooth*, convergence of methods such as BFGS to non-locally-minimizing stationary points or local maxima is *possible* but not likely, because of the line search, and such convergence will not be stable under perturbation.

However, this kind of convergence is what we are seeing for the non-regular, non-smooth Nesterov Chebyshev-Rosenbrock example, and it *is* stable under perturbation. The same behavior occurs for gradient sampling or bundle methods.

Convergence to Non-Locally-Minimizing Points

When f is *smooth*, convergence of methods such as BFGS to non-locally-minimizing stationary points or local maxima is *possible* but not likely, because of the line search, and such convergence will not be stable under perturbation.

However, this kind of convergence is what we are seeing for the non-regular, non-smooth Nesterov Chebyshev-Rosenbrock example, and it *is* stable under perturbation. The same behavior occurs for gradient sampling or bundle methods.

Kiwiel (private communication): the Nesterov example is the first he had seen which causes his bundle code to have this behavior.

Convergence to Non-Locally-Minimizing Points

When f is *smooth*, convergence of methods such as BFGS to non-locally-minimizing stationary points or local maxima is *possible* but not likely, because of the line search, and such convergence will not be stable under perturbation.

However, this kind of convergence is what we are seeing for the non-regular, non-smooth Nesterov Chebyshev-Rosenbrock example, and it *is* stable under perturbation. The same behavior occurs for gradient sampling or bundle methods.

Kiwiel (private communication): the Nesterov example is the first he had seen which causes his bundle code to have this behavior.

Nonetheless, we don't know whether, in exact arithmetic, the methods would actually generate sequences converging to the nonminimizing Clarke stationary points. Experiments by Kaku (2011) suggest that the higher the precision used, the more likely BFGS is to eventually move away from such a point.

Limited Memory Methods

Limited Memory BFGS

"Full" BFGS requires storing an $n \times n$ matrix and doing matrix-vector multiplies, which is not possible when n is large.

Limited Memory BFGS

"Full" BFGS requires storing an $n \times n$ matrix and doing matrix-vector multiplies, which is not possible when n is large.

In the 1980s, J. Nocedal and others developed a "limited memory" version of BFGS, with $O(n)$ space and time requirements, which is very widely used for minimizing smooth functions in many variables. It works by saving only the most recent k rank two updates to an initial inverse Hessian approximation.

Limited Memory BFGS

"Full" BFGS requires storing an $n \times n$ matrix and doing matrix-vector multiplies, which is not possible when n is large.

In the 1980s, J. Nocedal and others developed a "limited memory" version of BFGS, with $O(n)$ space and time requirements, which is very widely used for minimizing smooth functions in many variables. It works by saving only the most recent k rank two updates to an initial inverse Hessian approximation.

There are two variants: with and without "scaling" (usually scaling is preferred).

Limited Memory BFGS

"Full" BFGS requires storing an $n \times n$ matrix and doing matrix-vector multiplies, which is not possible when n is large.

In the 1980s, J. Nocedal and others developed a "limited memory" version of BFGS, with $O(n)$ space and time requirements, which is very widely used for minimizing smooth functions in many variables. It works by saving only the most recent k rank two updates to an initial inverse Hessian approximation.

There are two variants: with and without "scaling" (usually scaling is preferred).

The convergence rate of limited memory BFGS is linear, not superlinear, on smooth problems.

Limited Memory BFGS

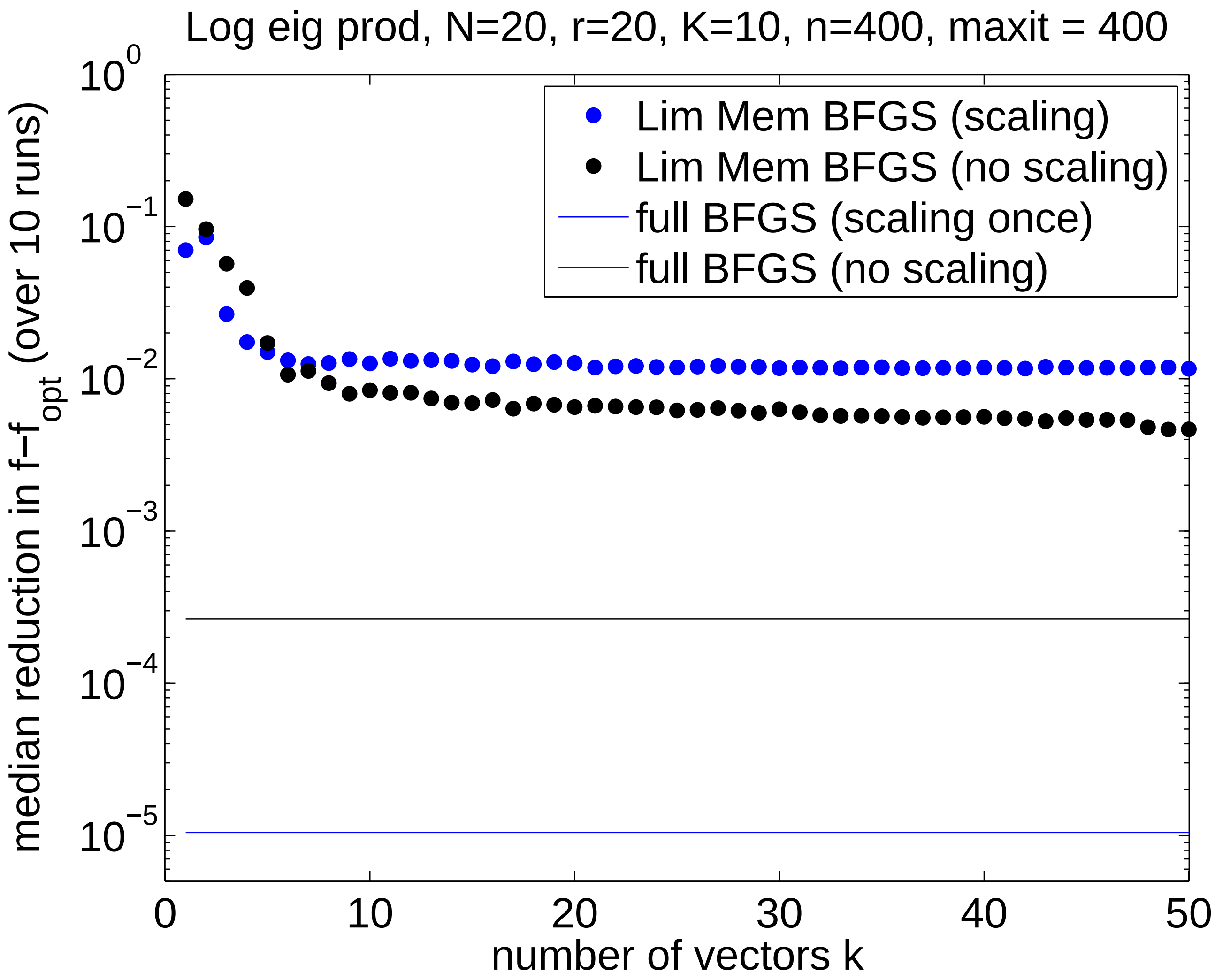
"Full" BFGS requires storing an $n \times n$ matrix and doing matrix-vector multiplies, which is not possible when n is large.

In the 1980s, J. Nocedal and others developed a "limited memory" version of BFGS, with $O(n)$ space and time requirements, which is very widely used for minimizing smooth functions in many variables. It works by saving only the most recent k rank two updates to an initial inverse Hessian approximation.

There are two variants: with and without "scaling" (usually scaling is preferred).

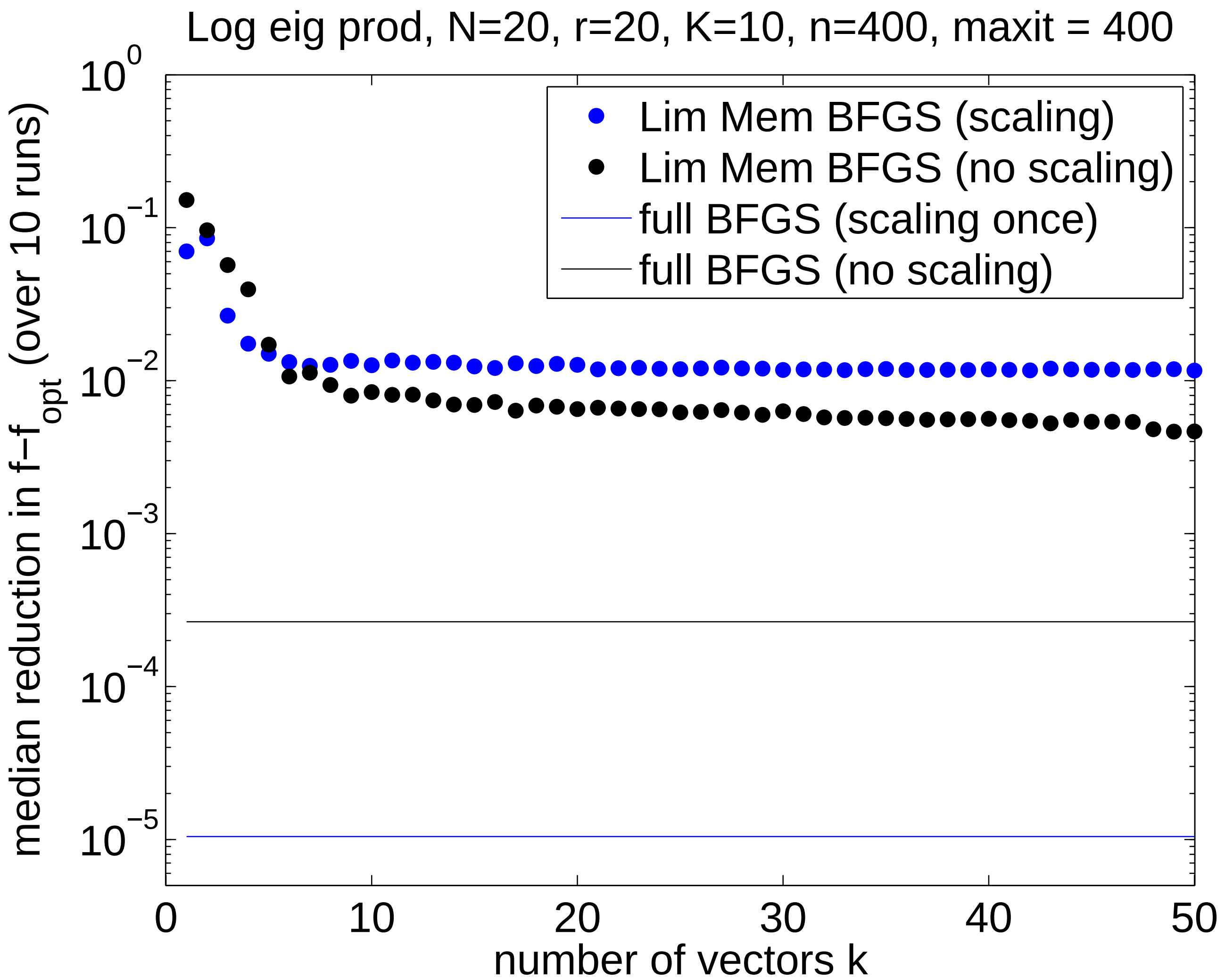
The convergence rate of limited memory BFGS is linear, not superlinear, on smooth problems.

Question: how effective is it on nonsmooth problems?



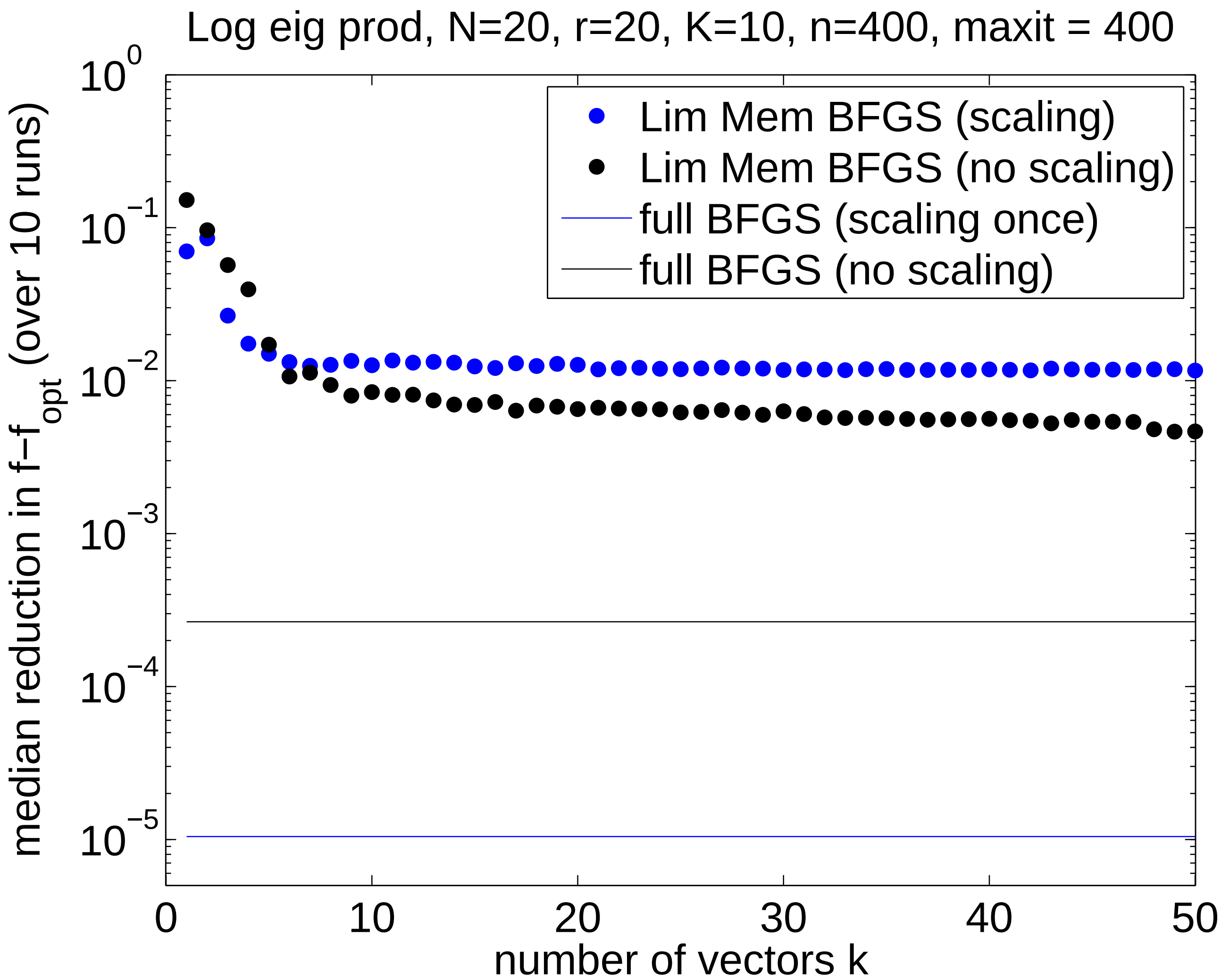
Log eig prod, N=20, r=20, K=10, n=400, maxit = 400
median reduction in f−f_opt (over 10 runs)
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
0
10
20
30
40
50
number of vectors k
Lim Mem BFGS (scaling)
Lim Mem BFGS (no scaling)
full BFGS (scaling once)
full BFGS (no scaling)

Limited Memory BFGS on the Eigenvalue Product



Limited Memory is not nearly as good as full BFGS

Limited Memory BFGS on the Eigenvalue Product



Limited Memory is not nearly as good as full BFGS

No significant improvement when k reaches 44

A More Basic Example

Let $x = [y; z; w] \in \mathbb{R}^{n_A+n_B+n_R}$ and consider the test function

$$f(x) = (y-e)^T A(y-e) + \{(z-e)^T B(z-e)\}^{1/2} + R_1(w)$$

where $A = A^T \succ 0,\ B = B^T \succ 0,\ e = [1; 1; \ldots; 1]$.

A More Basic Example

Let $x = [y; z; w] \in \mathbb{R}^{n_A+n_B+n_R}$ and consider the test function

$$f(x) = (y-e)^T A(y-e) + \{(z-e)^T B(z-e)\}^{1/2} + R_1(w)$$

where $A = A^T \succ 0,\ B = B^T \succ 0,\ e = [1; 1; \ldots; 1]$.

The first term is quadratic, the second is nonsmooth but convex, and the third is the nonsmooth, nonconvex Rosenbrock function.

A More Basic Example

Let $x = [y; z; w] \in \mathbb{R}^{n_A+n_B+n_R}$ and consider the test function

$$f(x) = (y-e)^T A(y-e) + \{(z-e)^T B(z-e)\}^{1/2} + R_1(w)$$

where $A = A^T \succ 0,\ B = B^T \succ 0,\ e = [1; 1; \ldots; 1]$.

The first term is quadratic, the second is nonsmooth but convex, and the third is the nonsmooth, nonconvex Rosenbrock function.

The optimal value is 0, with $x = e$. The function f is partly smooth and the dimension of the V-space is $n_B + n_R - 1$.

A More Basic Example

Let $x = [y; z; w] \in \mathbb{R}^{n_A+n_B+n_R}$ and consider the test function

$$f(x) = (y-e)^T A(y-e) + \{(z-e)^T B(z-e)\}^{1/2} + R_1(w)$$

where $A = A^T \succ 0,\ B = B^T \succ 0,\ e = [1; 1; \ldots; 1]$.

The first term is quadratic, the second is nonsmooth but convex, and the third is the nonsmooth, nonconvex Rosenbrock function.

The optimal value is 0, with $x = e$. The function f is partly smooth and the dimension of the V-space is $n_B + n_R - 1$.

Set $A = XX^T$ where x_{ij} are normally distributed, with condition number about 10^6 when $n_A = 200$. Similarly B with $n_B < n_A$.

A More Basic Example

Let $x = [y; z; w] \in \mathbb{R}^{n_A+n_B+n_R}$ and consider the test function

$$f(x) = (y-e)^T A(y-e) + \{(z-e)^T B(z-e)\}^{1/2} + R_1(w)$$

where $A = A^T \succ 0,\ B = B^T \succ 0,\ e = [1; 1; \ldots; 1]$.

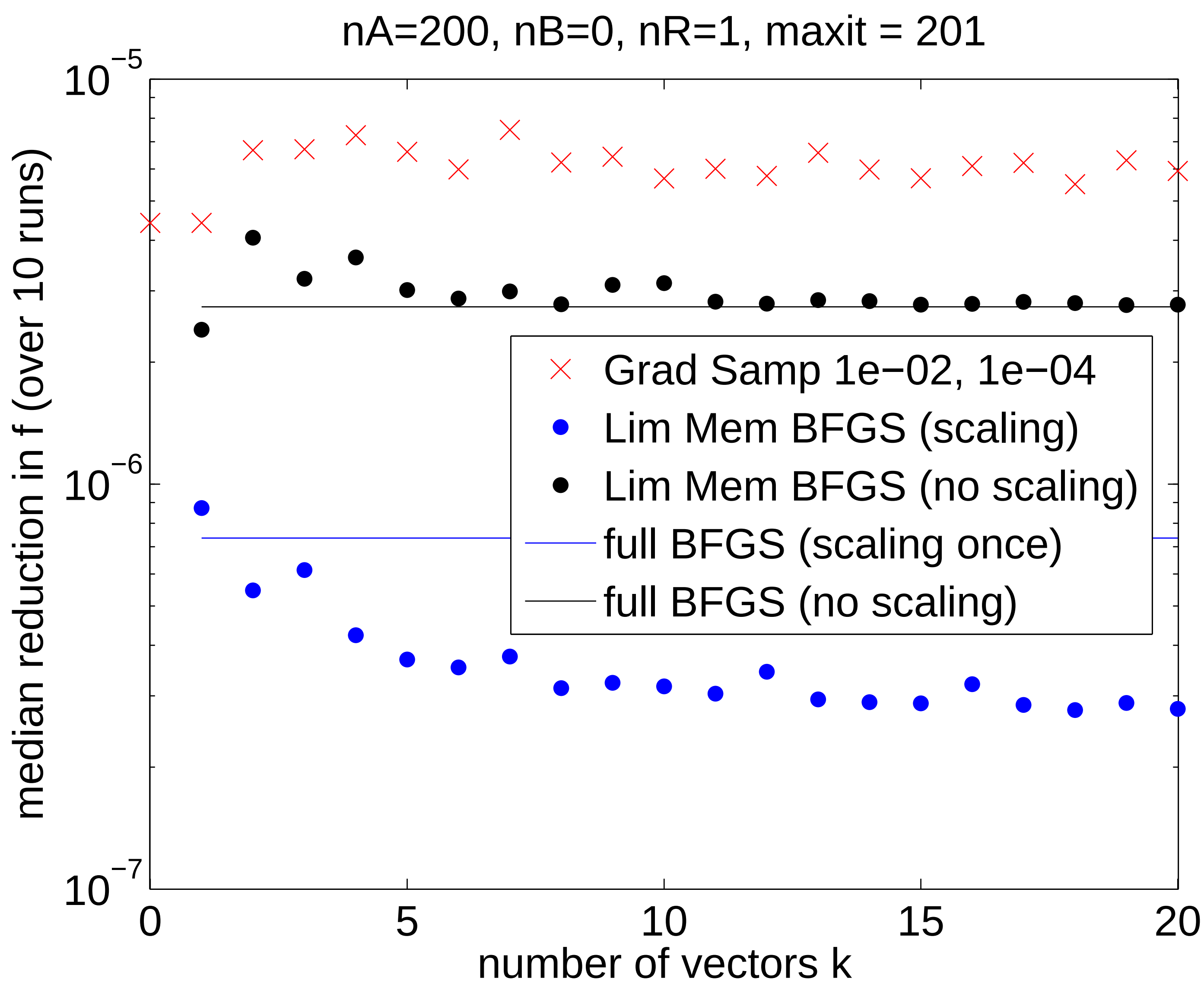
The first term is quadratic, the second is nonsmooth but convex, and the third is the nonsmooth, nonconvex Rosenbrock function.

The optimal value is 0, with $x = e$. The function f is partly smooth and the dimension of the V-space is $n_B + n_R - 1$.

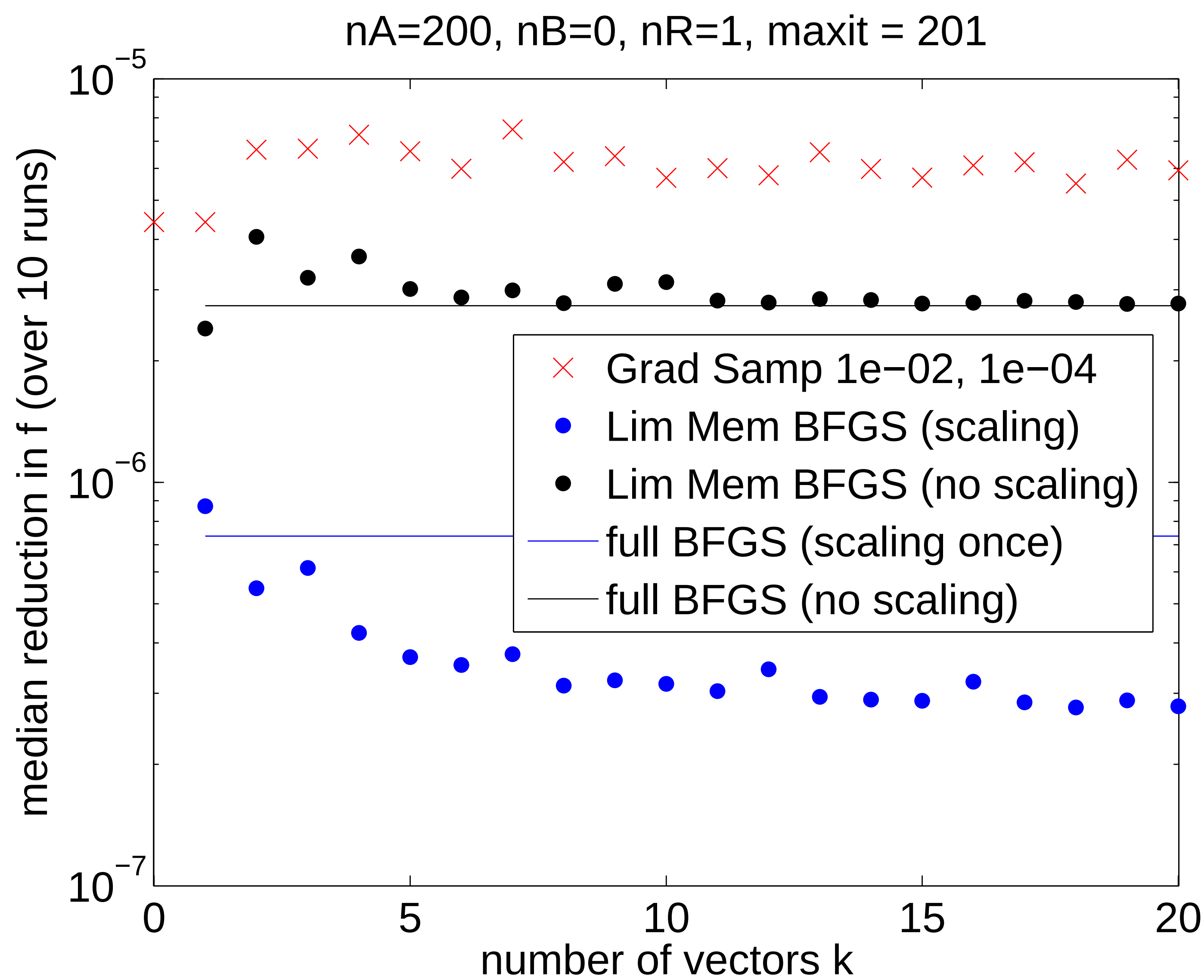
Set $A = XX^T$ where x_{ij} are normally distributed, with condition number about 10^6 when $n_A = 200$. Similarly B with $n_B < n_A$.

Besides limited memory BFGS and full BFGS, we also compare limited memory Gradient Sampling, where we sample $k \ll n$ gradients per iteration.

Smooth, Convex: $n_A = 200, n_B = 0, n_R = 1$

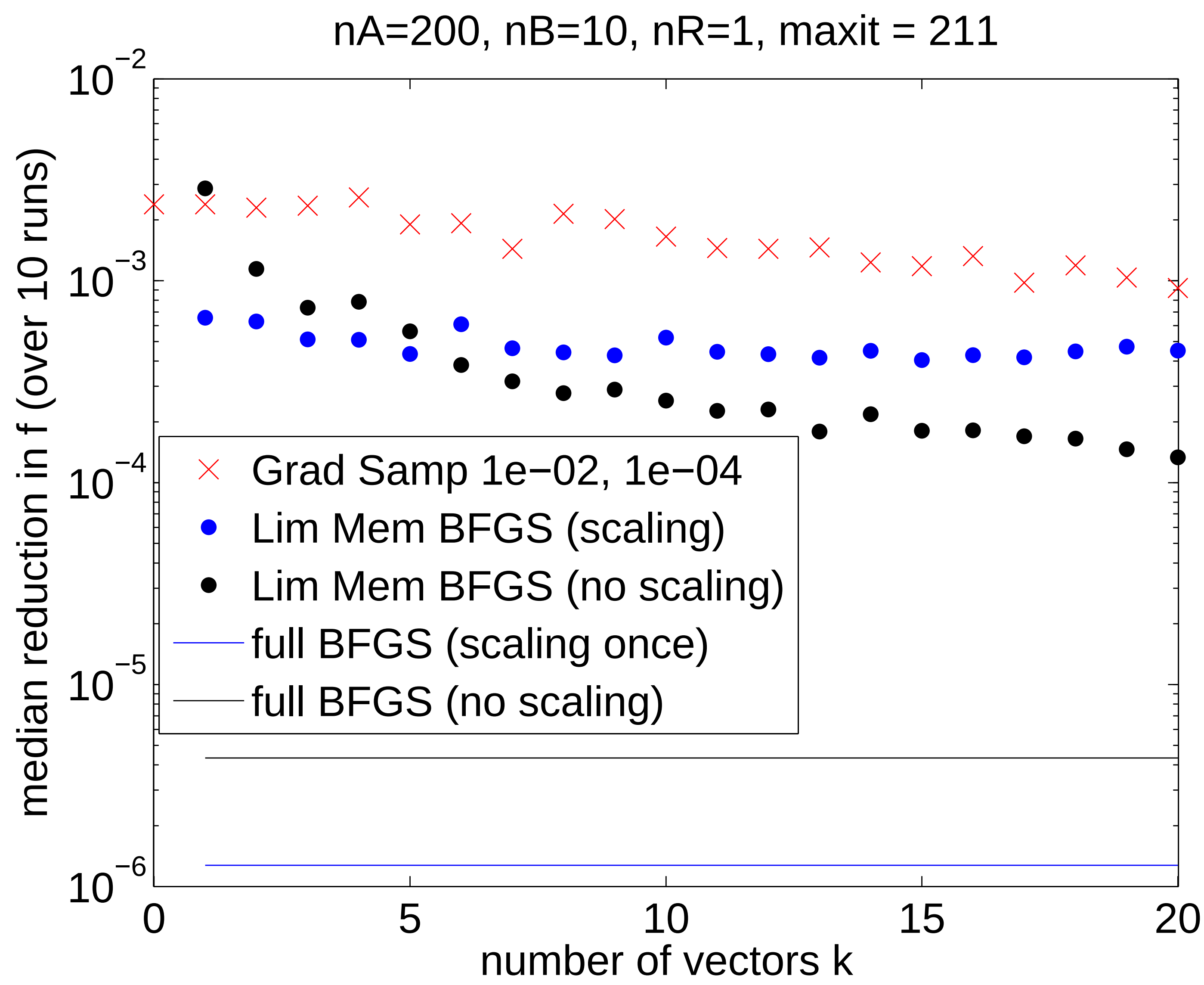


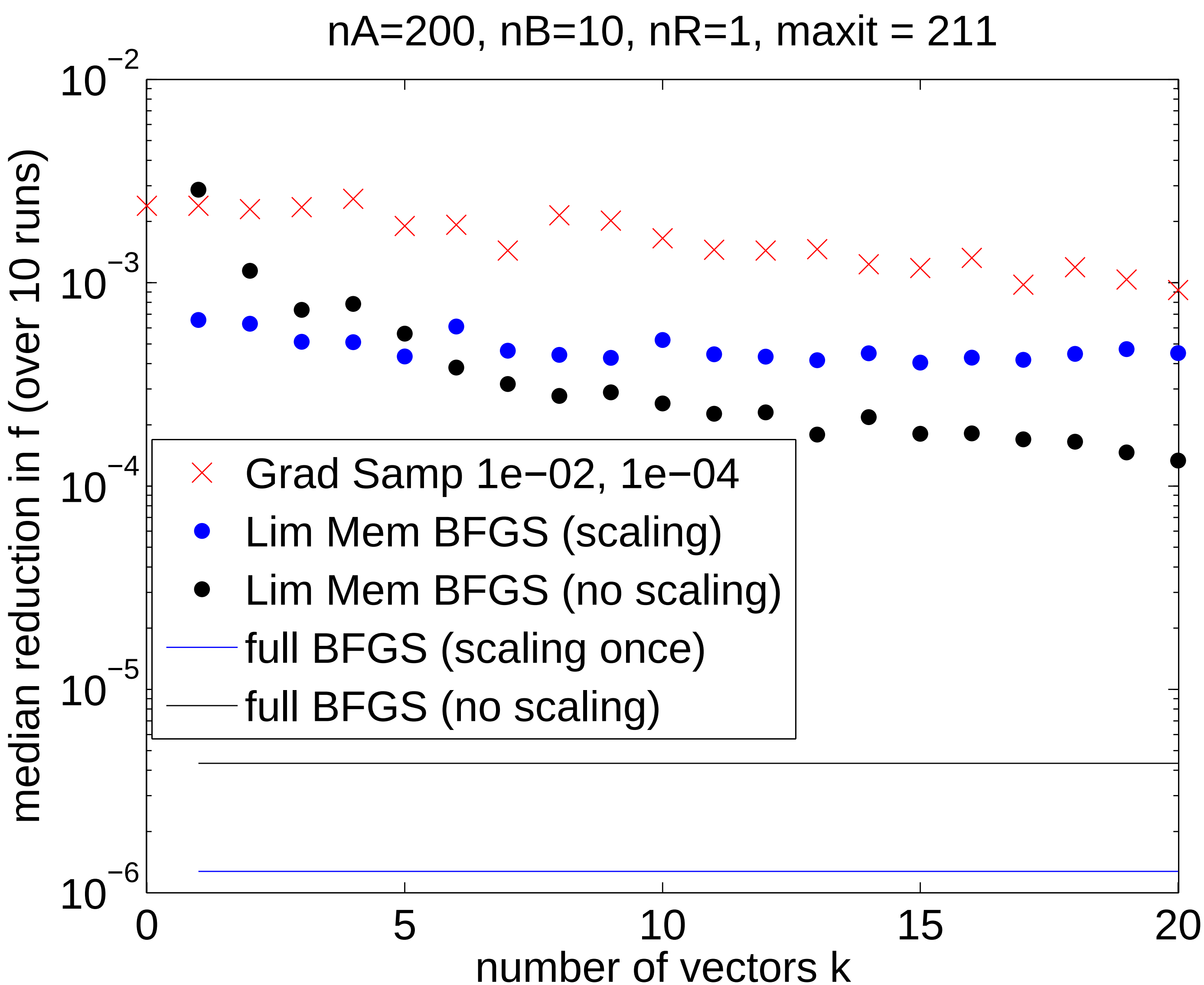
Smooth, Convex: $n_A = 200, n_B = 0, n_R = 1$



LM-BFGS with scaling even better than full BFGS

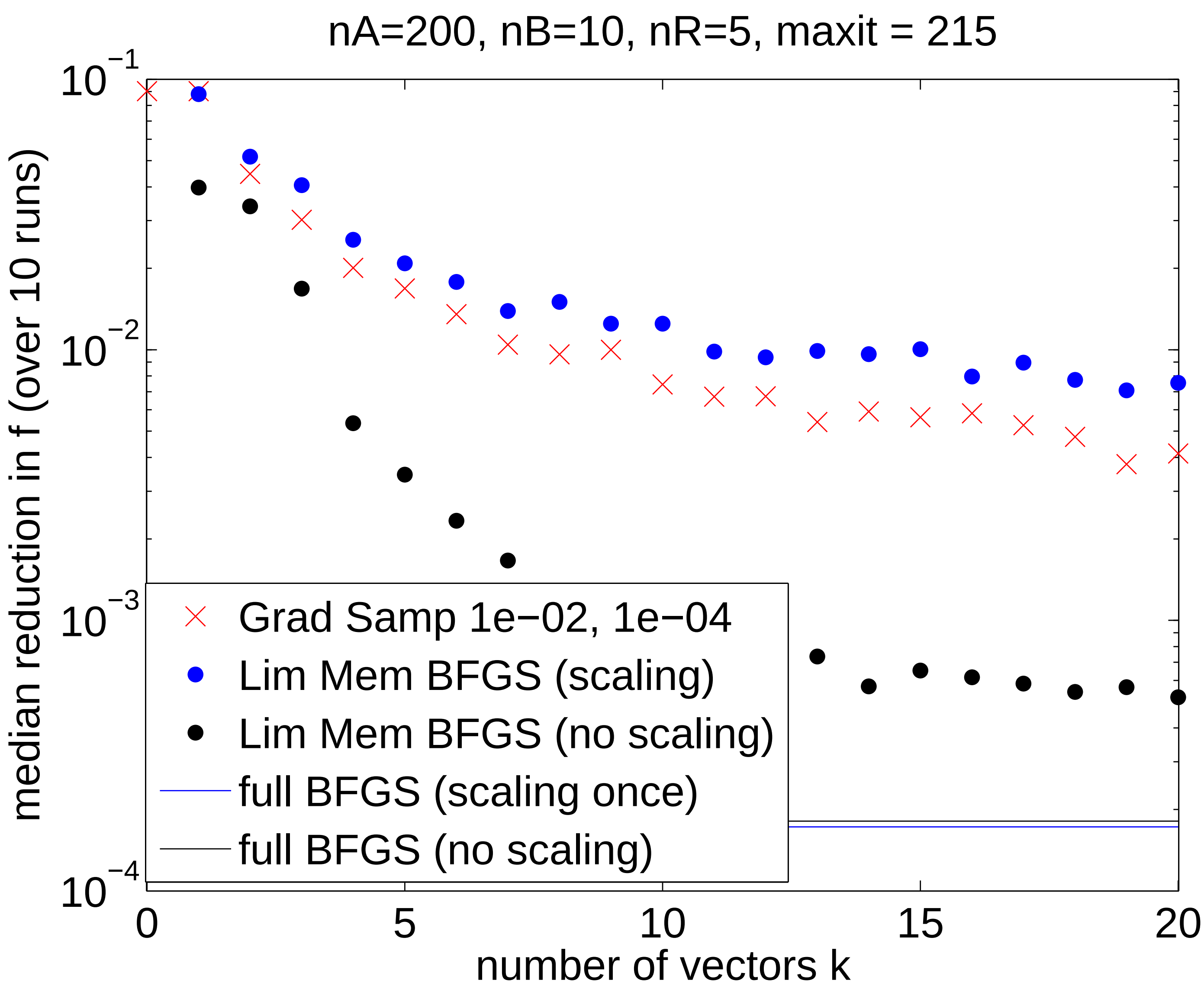
Nonsmooth, Convex: $n_A = 200, n_B = 10, n_R = 1$

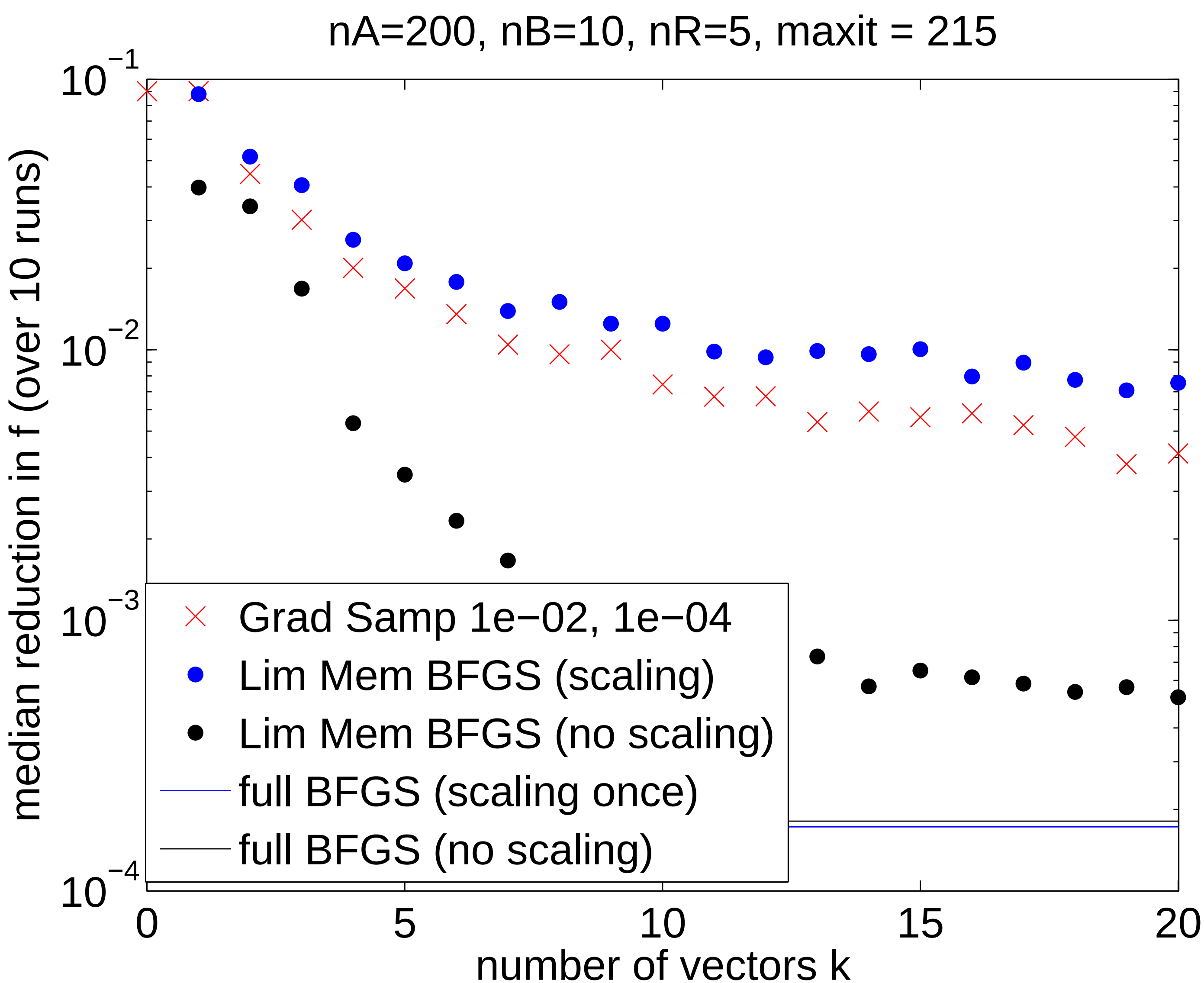




LM-BFGS much worse than full BFGS

Nonsmooth, Nonconvex: $n_A = 200, n_B = 10, n_R = 5$





LM-BFGS with scaling even worse than LM-Grad-Samp

Limited Effectiveness of Limited Memory BFGS

We see that that addition of nonsmoothness to a problem, convex or nonconvex, creates great difficulties for Limited Memory BFGS, even when the dimension of the V-space is less than the size of the memory, although it helps to turn off scaling. With scaling it may be no better than Limited Memory Gradient Sampling. More investigation of this is needed.

Other Ideas for Large Scale Nonsmooth Optimization

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.
- Smoothing! Lots of work on this has been done too, most notably by Yu. Nesterov.

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.
- Smoothing! Lots of work on this has been done too, most notably by Yu. Nesterov.
- Bundle methods, pioneered by C. Lemaréchal in the convex case and K. Kiwiel in the 1980s in the nonconvex case, and with lots of work done since, e.g. by P. Apkarian and D. Noll in small-scale control applications and by T.M.T. Do and T. Artières in large-scale machine learning applications.

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.
- Smoothing! Lots of work on this has been done too, most notably by Yu. Nesterov.
- Bundle methods, pioneered by C. Lemaréchal in the convex case and K. Kiwiel in the 1980s in the nonconvex case, and with lots of work done since, e.g. by P. Apkarian and D. Noll in small-scale control applications and by T.M.T. Do and T. Artières in large-scale machine learning applications.
- Lots of other recent work on nonconvexity in machine learning.

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.
- Smoothing! Lots of work on this has been done too, most notably by Yu. Nesterov.
- Bundle methods, pioneered by C. Lemaréchal in the convex case and K. Kiwiel in the 1980s in the nonconvex case, and with lots of work done since, e.g. by P. Apkarian and D. Noll in small-scale control applications and by T.M.T. Do and T. Artières in large-scale machine learning applications.
- Lots of other recent work on nonconvexity in machine learning.
- Adaptive Gradient Sampling (F.E. Curtis and X. Que).

Other Ideas for Large Scale Nonsmooth Optimization

- Exploit structure! Lots of work on this has been done.
- Smoothing! Lots of work on this has been done too, most notably by Yu. Nesterov.
- Bundle methods, pioneered by C. Lemaréchal in the convex case and K. Kiwiel in the 1980s in the nonconvex case, and with lots of work done since, e.g. by P. Apkarian and D. Noll in small-scale control applications and by T.M.T. Do and T. Artières in large-scale machine learning applications.
- Lots of other recent work on nonconvexity in machine learning.
- Adaptive Gradient Sampling (F.E. Curtis and X. Que).
- Automatic Differentiation (AD): (B. Bell, A. Griewank).

Concluding Remarks

Summary

Gradient Sampling is a simple method for nonsmooth, nonconvex optimization for which a convergence theory is known, but it is too expensive to use in most applications.

Summary

Gradient Sampling is a simple method for nonsmooth, nonconvex optimization for which a convergence theory is known, but it is too expensive to use in most applications.

BFGS — the full version — is remarkably effective on nonsmooth problems, but little theory is known. Our package HIFOO (H-infinity fixed order optimization) for controller design, primarily based on BFGS, has been used successfully in many applications.

Summary

Gradient Sampling is a simple method for nonsmooth, nonconvex optimization for which a convergence theory is known, but it is too expensive to use in most applications.

BFGS — the full version — is remarkably effective on nonsmooth problems, but little theory is known. Our package HIFOO (H-infinity fixed order optimization) for controller design, primarily based on BFGS, has been used successfully in many applications.

Limited Memory BFGS is not so effective on nonsmooth problems, but it seems to help to turn off scaling.

Summary

Gradient Sampling is a simple method for nonsmooth, nonconvex optimization for which a convergence theory is known, but it is too expensive to use in most applications.

BFGS — the full version — is remarkably effective on nonsmooth problems, but little theory is known. Our package HIFOO (H-infinity fixed order optimization) for controller design, primarily based on BFGS, has been used successfully in many applications.

Limited Memory BFGS is not so effective on nonsmooth problems, but it seems to help to turn off scaling.

Diabolical nonconvex problems such as Nesterov's Chebyshev-Rosenbrock problems can be very difficult, especially in the nonsmooth case.

A Final Quote

"Nonconvexity is scary to some, but there are vastly different types of nonconvexity (some of which are really scary!)"

— Yann LeCun

A Final Quote

"Nonconvexity is scary to some, but there are vastly different types of nonconvexity (some of which are really scary!)"

— Yann LeCun

Papers, software are available at **www.cs.nyu.edu/overton**.