On Nesterov's Nonsmooth Chebyshev-Rosenbrock Functions

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

Les Houches, 8 February 2016

Yurii Nesterov

Yurii Nesterov

- It seems we first met in 1988 at the Tokyo ISMP. We don't have a proof of this, but we do have a proof that we were both at the meeting: we both used the beautiful gray bag with the Samurai warrior design for many years, bringing it to other conferences long after everyone else abandoned theirs!

Yurii Nesterov

- It seems we first met in 1988 at the Tokyo ISMP. We don't have a proof of this, but we do have a proof that we were both at the meeting: we both used the beautiful gray bag with the Samurai warrior design for many years, bringing it to other conferences long after everyone else abandoned theirs!
- We definitely met in 1994 at the Ann Arbor ISMP, where I learned about the Nesterov-Todd primal-dual interior-point algorithm for SDP.

Yurii Nesterov

- It seems we first met in 1988 at the Tokyo ISMP. We don't have a proof of this, but we do have a proof that we were both at the meeting: we both used the beautiful gray bag with the Samurai warrior design for many years, bringing it to other conferences long after everyone else abandoned theirs!
- We definitely met in 1994 at the Ann Arbor ISMP, where I learned about the Nesterov-Todd primal-dual interior-point algorithm for SDP.
- We met again on many subsequent occasions, most notably during very enjoyable extended visits to Louvain-la-neuve in 2004 and 2008.

Yurii Nesterov

- It seems we first met in 1988 at the Tokyo ISMP. We don't have a proof of this, but we do have a proof that we were both at the meeting: we both used the beautiful gray bag with the Samurai warrior design for many years, bringing it to other conferences long after everyone else abandoned theirs!
- We definitely met in 1994 at the Ann Arbor ISMP, where I learned about the Nesterov-Todd primal-dual interior-point algorithm for SDP.
- We met again on many subsequent occasions, most notably during very enjoyable extended visits to Louvain-la-neuve in 2004 and 2008.
- Always a great pleasure to interact with this brilliant but modest colleague!

Introduction

Nonsmooth, Nonconvex Optimization

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

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

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

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

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

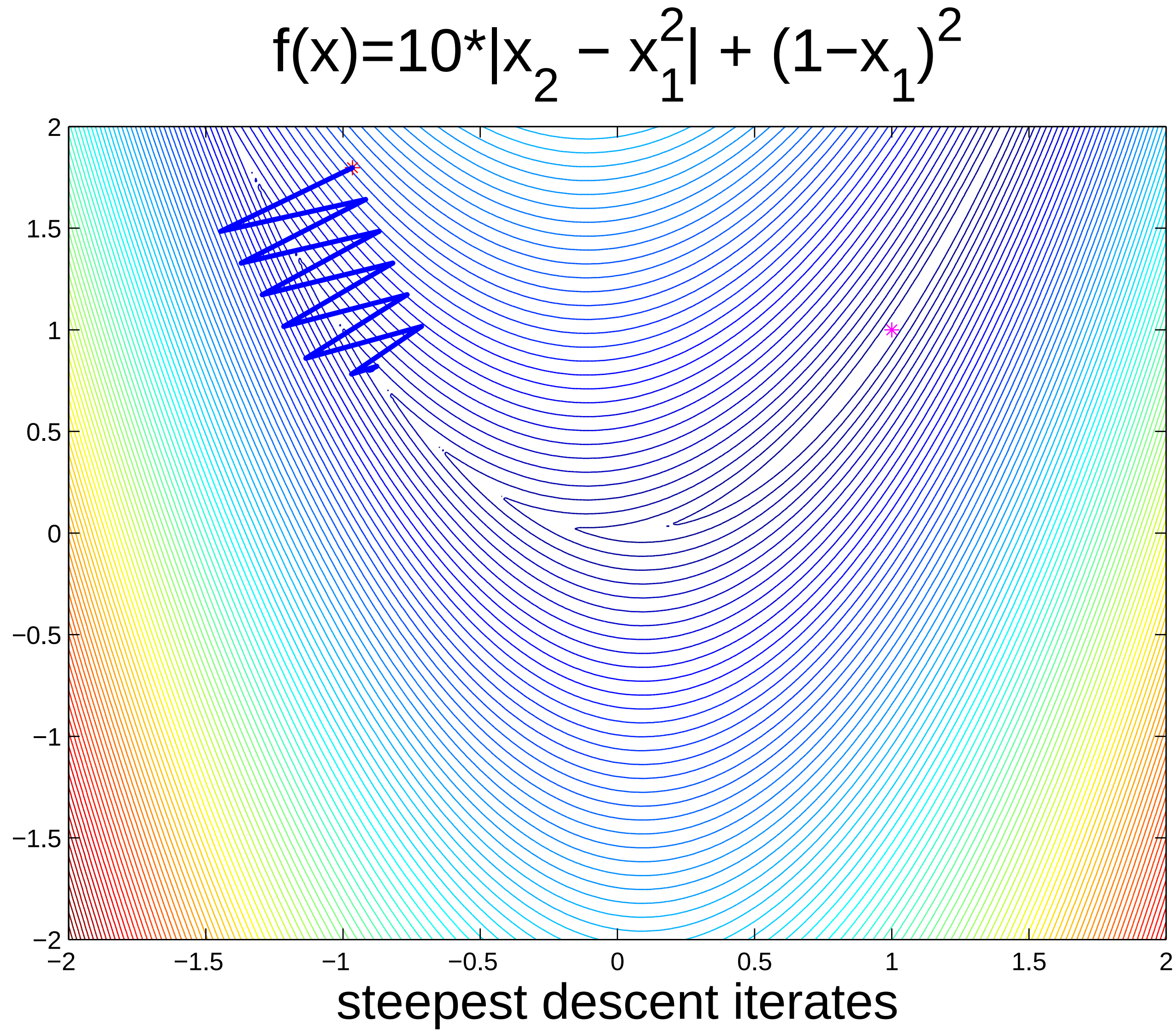
- Continuous
- Not differentiable everywhere, in particular often not differentiable at local minimizers
- Not convex
- Usually, but not always, locally Lipschitz

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

In fact, it's been known for several decades that at any given iterate, one should 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, one should 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, one should 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, one should 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

Methods Suitable for Nonsmooth Functions

In fact, it's been known for several decades that at any given iterate, one should 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

A completely different approach using randomized gradient-free methods: the first complexity result for nonsmooth, nonconvex optimization (Y. Nesterov and V. Spokoiny, JFoCM, 2015).

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).

The BFGS Method (“Full” Version)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

The BFGS Method ("Full" Version)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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

The BFGS Method ("Full" Version)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

The BFGS Method ("Full" Version)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

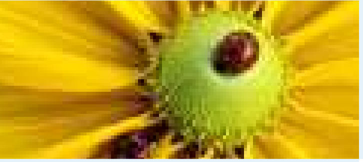
Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

Yurii Nesterov

Introduction

Nonsmooth, Nonconvex Optimization

Example

Methods Suitable for Nonsmooth Functions

Failure of Steepest Descent: Simpler Example

The BFGS Method ("Full" Version)

BFGS for Nonsmooth Optimization

With BFGS

Some Nonsmooth Analysis

Nesterov's Chebyshev-Rosenbrock Functions

Other Examples of Behavior of BFGS on Nonsmooth Functions

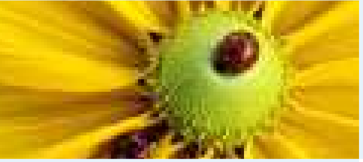
Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

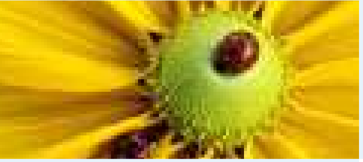
Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

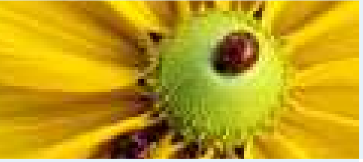
Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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 $V = I - \frac{1}{s^Ty}sy^T$



The BFGS Method ("Full" Version)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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)

Broyden, Fletcher, Goldfarb, Shanno independently, 1970

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 Armijo condition ensures "sufficient decrease" in f

The Wolfe condition ensures that the directional derivative along the line increases algebraically, which guarantees that $s^T y > 0$ 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 very 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 very 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 very 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 very 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 very 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 very 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!

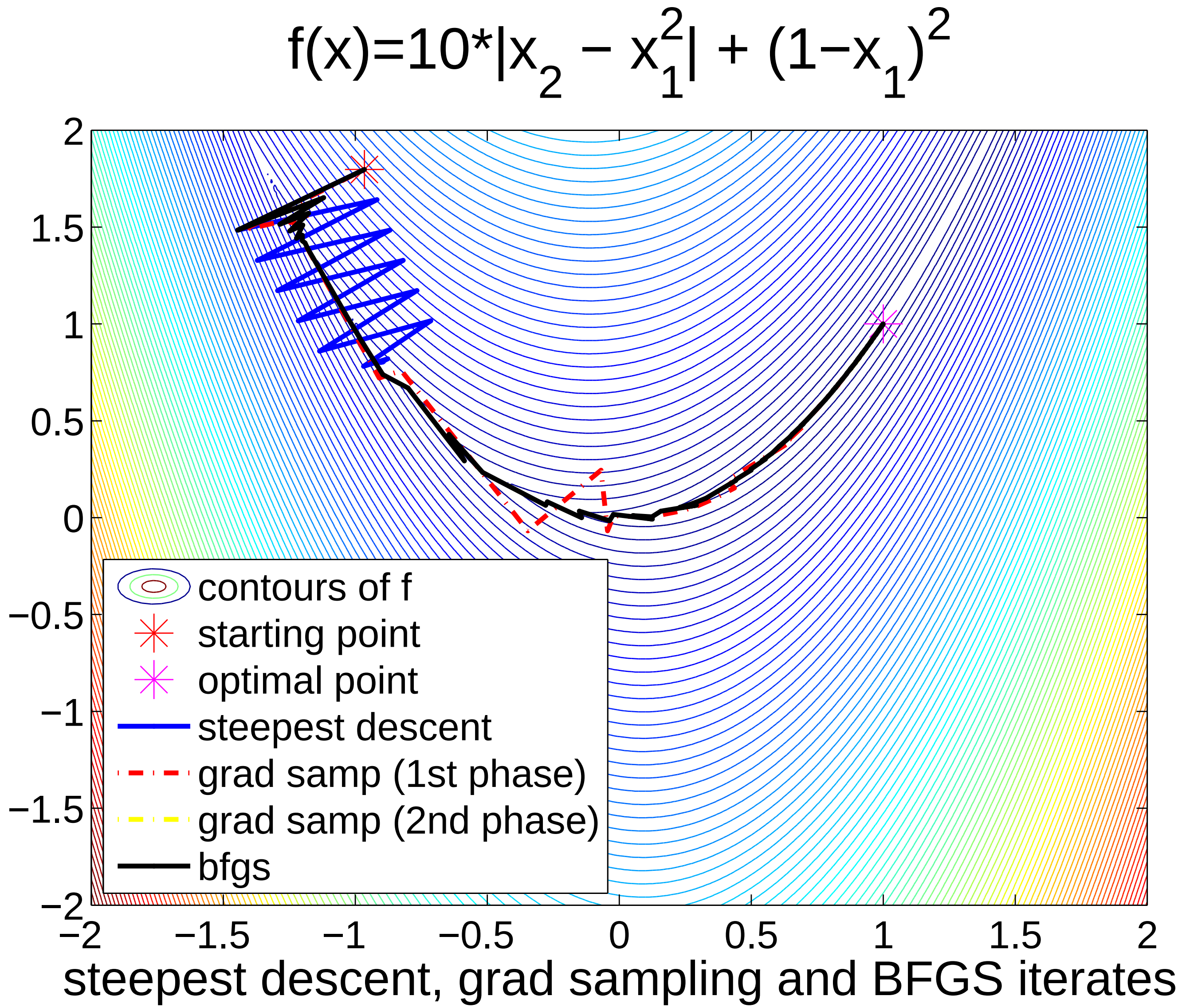
In the nonsmooth case, BFGS builds a very ill-conditioned inverse "Hessian" approximation, with some tiny eigenvalues converging to zero, corresponding to "infinitely large" curvature in the directions defined by the associated eigenvectors.

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*|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
contours of f
starting point
optimal point
steepest descent
grad samp (1st phase)
grad samp (2nd phase)
bfgs
steepest descent, grad sampling and BFGS iterates

Some Nonsmooth Analysis

The Clarke Subdifferential

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

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

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

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

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

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

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

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

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

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

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

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

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

If f is convex, $\partial^C f$ 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

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

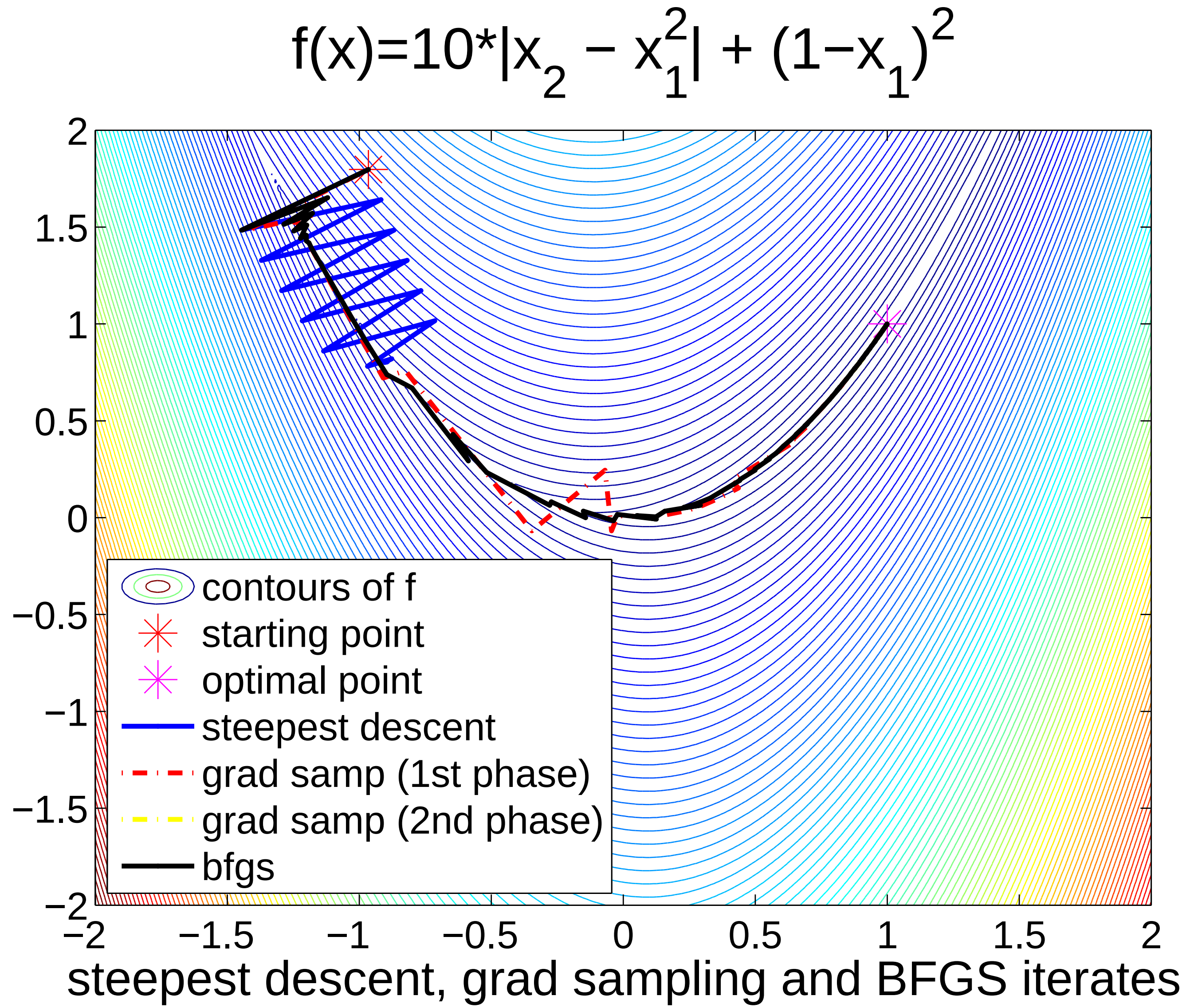
F.H. Clarke, 1973 (he used the name "generalized gradient").

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

If f is convex, $\partial^C f$ is the subdifferential of convex analysis.

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

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



Regularity

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

Regularity

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

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

Regularity

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

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

- All convex functions are regular

Regularity

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

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

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

Regularity

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

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

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

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

Partly Smooth Functions

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

Partly Smooth Functions

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

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

Partly Smooth Functions

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

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

Partly Smooth Functions

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

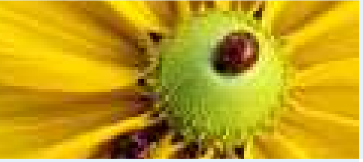
- its restriction to $\mathcal{M}$ is twice continuously differentiable near $\bar{x}$
- the Clarke subdifferential $\partial^C f$ is continuous on $\mathcal{M}$ near $\bar{x}$
- par $\partial^C f(\bar{x})$, the subspace parallel to the affine hull of the subdifferential of f at $\bar{x}$, is exactly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

Partly Smooth Functions

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

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

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



Partly Smooth Functions

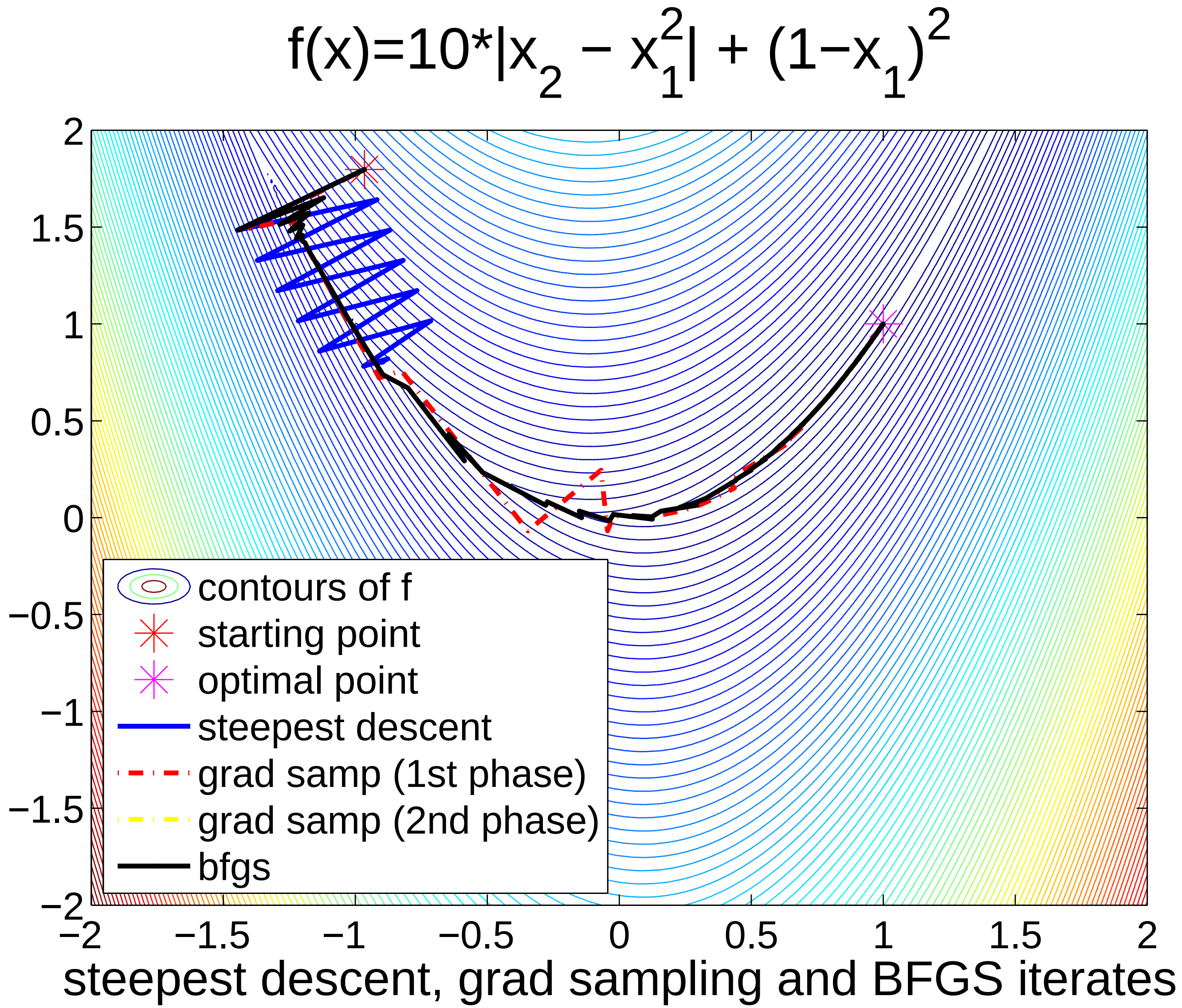
A regular function f is *partly smooth* at $\bar{x}$ relative to a manifold $\mathcal{M}$ containing $\bar{x}$ (A.S. Lewis 2003) if

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

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

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

Illustration of U and V-spaces on Same Example



Nesterov's Chebyshev-Rosenbrock Functions

Nesterov's First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

Nesterov's First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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

Nesterov's First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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 First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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 First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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 First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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 First Chebyshev-Rosenbrock Function

Nesterov (2008, private comm.): consider the function

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

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*!

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 $x \in \mathcal{M}_N$,

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

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 $[-1, 1]$

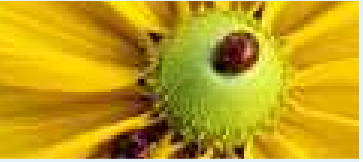
Why BFGS Takes So Many Iterations to Minimize N_2

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 $x \in \mathcal{M}_N$,

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

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!

Why BFGS Takes So Many Iterations to Minimize N_2

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

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

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!

Newton's method is not much faster, although it converges quadratically at the end.

Length of a Piecewise Linear Descent Path

F. Jarre (2013): if the second term (the sum) in Nesterov's smooth Chebyshev-Rosenbrock function N_2 is weighted by 400, any continuous piecewise linear descent path starting at $\hat{x}$ and leading to the global minimizer x^* has

at least 1.618^n linear segments.

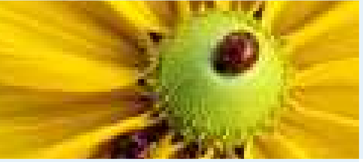
Nesterov's First C-R Function: Nonsmooth Case

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

Nesterov's First C-R Function: Nonsmooth Case

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

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 $\mathcal{M}_N$.



Nesterov's First C-R Function: Nonsmooth Case

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

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. $\mathcal{M}_N$, and $x^* = [1, 1, \ldots, 1]^T$ is its only stationary point.

Nesterov's First C-R Function: Nonsmooth Case

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

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. $\mathcal{M}_N$, and $x^* = [1, 1, \ldots, 1]^T$ is its only stationary point.

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

Nesterov's First C-R Function: Nonsmooth Case

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

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. $\mathcal{M}_N$, and $x^* = [1, 1, \ldots, 1]^T$ is its only stationary point.

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

Nesterov's First C-R Function: Nonsmooth Case

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

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. $\mathcal{M}_N$, and $x^* = [1, 1, \ldots, 1]^T$ is its only stationary point.

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

Nesterov's First C-R Function: Nonsmooth Case

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

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. $\mathcal{M}_N$, and $x^* = [1, 1, \ldots, 1]^T$ is its only stationary point.

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.

Nesterov's Second Nonsmooth C-R Function

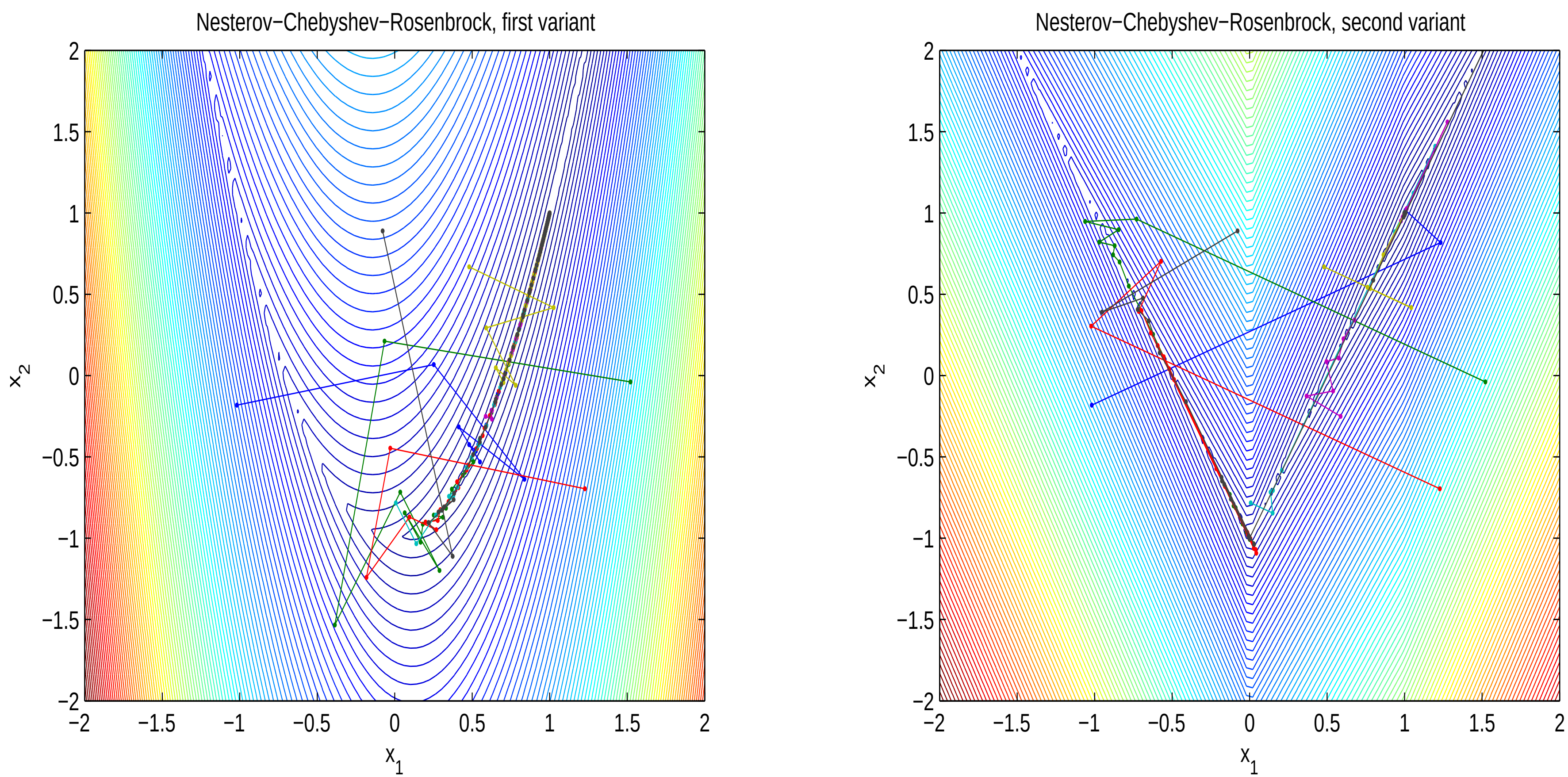
$$\widehat{N}_1(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|.$$

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

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

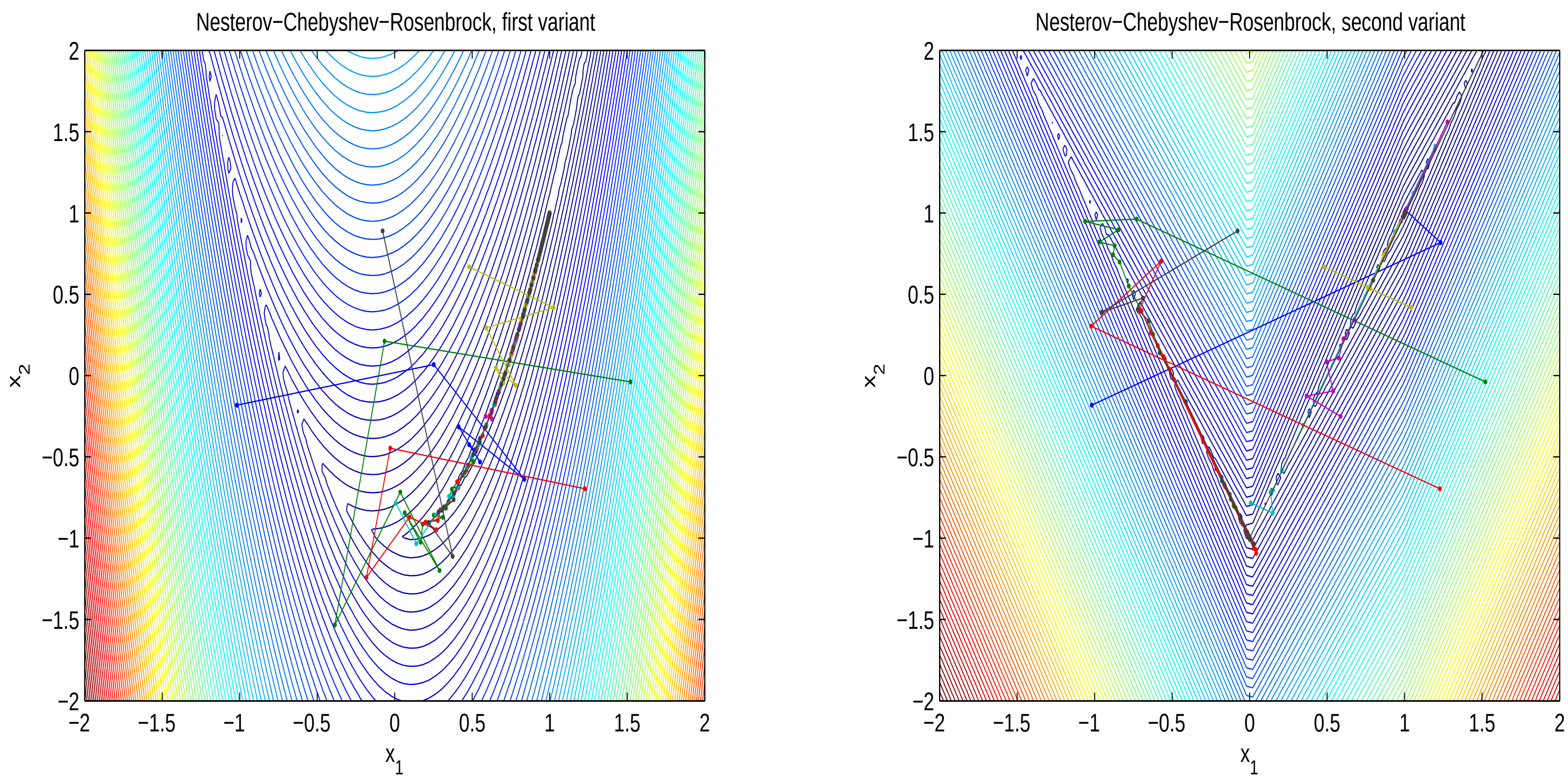
but S is not a manifold: it has "corners".

Contour Plots of 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.

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: $\widehat{N}_1'(x, d) < 0$.

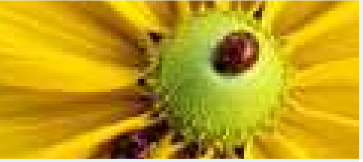
These two properties mean that $\widehat{N}_1$ is *not regular* at $[0, -1]^T$.

The Mordukhovich Subdifferential

B.S. Mordukhovich (1976), R.T. Rockafellar and R. J.-B. Wets (1998)

Consider a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ (not necessarily Lipschitz) and a point $\bar{x} \in \mathbb{R}^n$. A vector $\bar{v} \in \mathbb{R}^n$ is a *regular subgradient* of f at $\bar{x}$ (written $\bar{v} \in \hat{\partial} f(\bar{x})$) if

$$\liminf_{\substack{z \to \bar{x} \\ z \neq \bar{x}}} \frac{f(z) - f(\bar{x}) - \langle \bar{v}, z - \bar{x} \rangle}{|z - \bar{x}|} \geq 0.$$



The Mordukhovich Subdifferential

B.S. Mordukhovich (1976), R.T. Rockafellar and R. J.-B. Wets (1998)

Consider a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ (not necessarily Lipschitz) and a point $\bar{x} \in \mathbb{R}^n$. A vector $\bar{v} \in \mathbb{R}^n$ is a *regular subgradient* of f at $\bar{x}$ (written $\bar{v} \in \hat{\partial} f(\bar{x})$) if

$$\liminf_{\substack{z \to \bar{x} \\ z \neq \bar{x}}} \frac{f(z) - f(\bar{x}) - \langle \bar{v}, z - \bar{x} \rangle}{|z - \bar{x}|} \geq 0.$$

A vector $\bar{v} \in \mathbb{R}^n$ is a *Mordukhovich subgradient* of f at $\bar{x}$ (written $\bar{v} \in \partial^M f(\bar{x})$) if there exist sequences $\{x\}$ and $\{v\}$ in $\mathbb{R}^n$ satisfying

$$x \to \bar{x}$$
$$v \in \hat{\partial} f(x)$$
$$v \to \bar{v}.$$

The Mordukhovich Subdifferential

B.S. Mordukhovich (1976), R.T. Rockafellar and R. J.-B. Wets (1998)

Consider a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ (not necessarily Lipschitz) and a point $\bar{x} \in \mathbb{R}^n$. A vector $\bar{v} \in \mathbb{R}^n$ is a *regular subgradient* of f at $\bar{x}$ (written $\bar{v} \in \hat{\partial} f(\bar{x})$) if

$$\liminf_{\substack{z \to \bar{x} \\ z \neq \bar{x}}} \frac{f(z) - f(\bar{x}) - \langle \bar{v}, z - \bar{x} \rangle}{|z - \bar{x}|} \geq 0.$$

A vector $\bar{v} \in \mathbb{R}^n$ is a *Mordukhovich subgradient* of f at $\bar{x}$ (written $\bar{v} \in \partial^M f(\bar{x})$) if there exist sequences $\{x\}$ and $\{v\}$ in $\mathbb{R}^n$ satisfying

$$\begin{aligned} & x \to \bar{x} \\ & v \in \hat{\partial} f(x) \\ & v \to \bar{v}. \end{aligned}$$

We say f is *Mordukhovich stationary* at $\bar{x}$ if $0 \in \partial^M f(\bar{x})$.

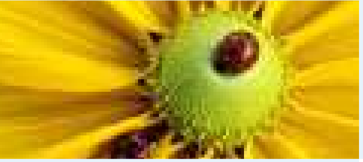
Relationship Between $\partial^C f$ and $\partial^M f$

For a locally Lipschitz function f, we have

$$\partial^C f(\bar{x}) = \text{conv}\ \partial^M f(\bar{x}).$$

and, if f is regular,

$$\partial^C f(\bar{x}) = \partial^M f(\bar{x}).$$



Relationship Between $\partial^C f$ and $\partial^M f$

For a locally Lipschitz function f, we have

$$\partial^C f(\bar{x}) = \text{conv}\ \partial^M f(\bar{x}).$$

and, if f is regular,

$$\partial^C f(\bar{x}) = \partial^M f(\bar{x}).$$

Example: let $g(x) = |x_1| - |x_2|$, $x \in \mathbb{R}^2$. Then

$$\partial^C g(0) = [-1, 1] \times [-1, 1] \quad \text{and} \quad \partial^M g(0) = [-1, 1] \times \{-1, 1\}$$

so g is not regular.

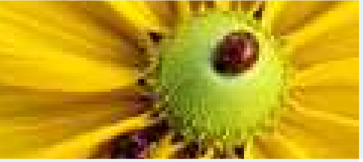
Theorem. For $n \geq 2$:

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

Back to Nesterov's Second Nonsmooth C-R Function

Theorem. For $n \geq 2$:

- $\widehat{N}_1$ has 2^{n-1} Clarke stationary points
- $\widehat{N}_1$ has exactly one Mordukhovich stationary point, the global minimizer x^*



Theorem. For $n \geq 2$:

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

M. Gürbüzbalaban and M.L.O., SIOPT, 2012.

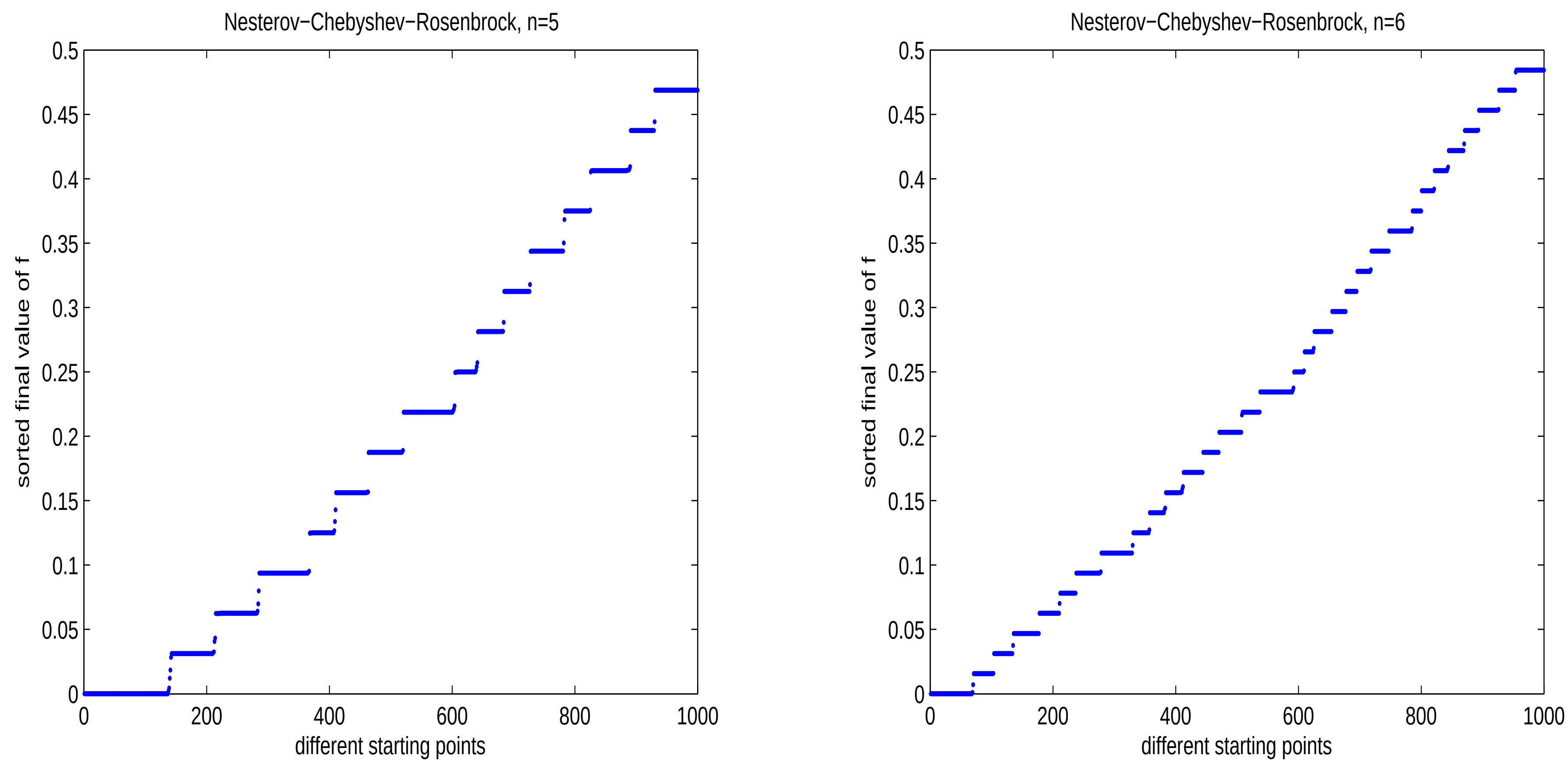
Theorem. For $n \geq 2$:

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

M. Gürbüzbalaban and M.L.O., SIOPT, 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.

Experiments using BFGS with Extended Precision

M.S. thesis by A. Kaku experimenting with Sherry Li's "double double" C++ package.

Experiments using BFGS with Extended Precision

M.S. thesis by A. Kaku experimenting with Sherry Li's "double double" C++ package.

"double double" is not the same as quadruple precision: each number is represented as the sum of two ordinary double precision numbers

Experiments using BFGS with Extended Precision

M.S. thesis by A. Kaku experimenting with Sherry Li's "double double" C++ package.

"double double" is not the same as quadruple precision: each number is represented as the sum of two ordinary double precision numbers

Thus, $1 + 10^{-30}$ and $1 + 10^{-300}$ are both valid "double double" numbers

Experiments using BFGS with Extended Precision

M.S. thesis by A. Kaku experimenting with Sherry Li's "double double" C++ package.

"double double" is not the same as quadruple precision: each number is represented as the sum of two ordinary double precision numbers

Thus, $1 + 10^{-30}$ and $1 + 10^{-300}$ are both valid "double double" numbers

In practice, it is just a convenient, inexpensive software implementation that approximates quadruple precision (approximately 32 decimal digits of accuracy instead of 16)

Experiments using BFGS with Extended Precision

M.S. thesis by A. Kaku experimenting with Sherry Li's "double double" C++ package.

"double double" is not the same as quadruple precision: each number is represented as the sum of two ordinary double precision numbers

Thus, $1 + 10^{-30}$ and $1 + 10^{-300}$ are both valid "double double" numbers

In practice, it is just a convenient, inexpensive software implementation that approximates quadruple precision (approximately 32 decimal digits of accuracy instead of 16)

Show plots from Kaku's thesis.

An Approach using Automatic Differentiation

Recent work by A. Griewank on automatic differentiation for nonsmooth optimization: leads to a more efficient method for optimization of Nesterov's *second* nonsmooth Chebyshev-Rosenbrock since it is able to efficiently exploit the piecewise-linearity of the function.

An Approach using Automatic Differentiation

Recent work by A. Griewank on automatic differentiation for nonsmooth optimization: leads to a more efficient method for optimization of Nesterov's *second* nonsmooth Chebyshev-Rosenbrock since it is able to efficiently exploit the piecewise-linearity of the function.

Starting at $\hat{x}$, it visits all 2^{n-1} Clarke stationary points, but it does not get stuck at any of them because it repeatedly solves LPs that define the piecewise linear path leading to the global minimum.

Other Examples of Behavior of BFGS on Nonsmooth Functions

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 $X \in S^N$.

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.

Minimizing a Product of Eigenvalues

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

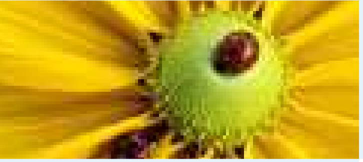
$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \lambda_N(X)$$

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.



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.

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

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

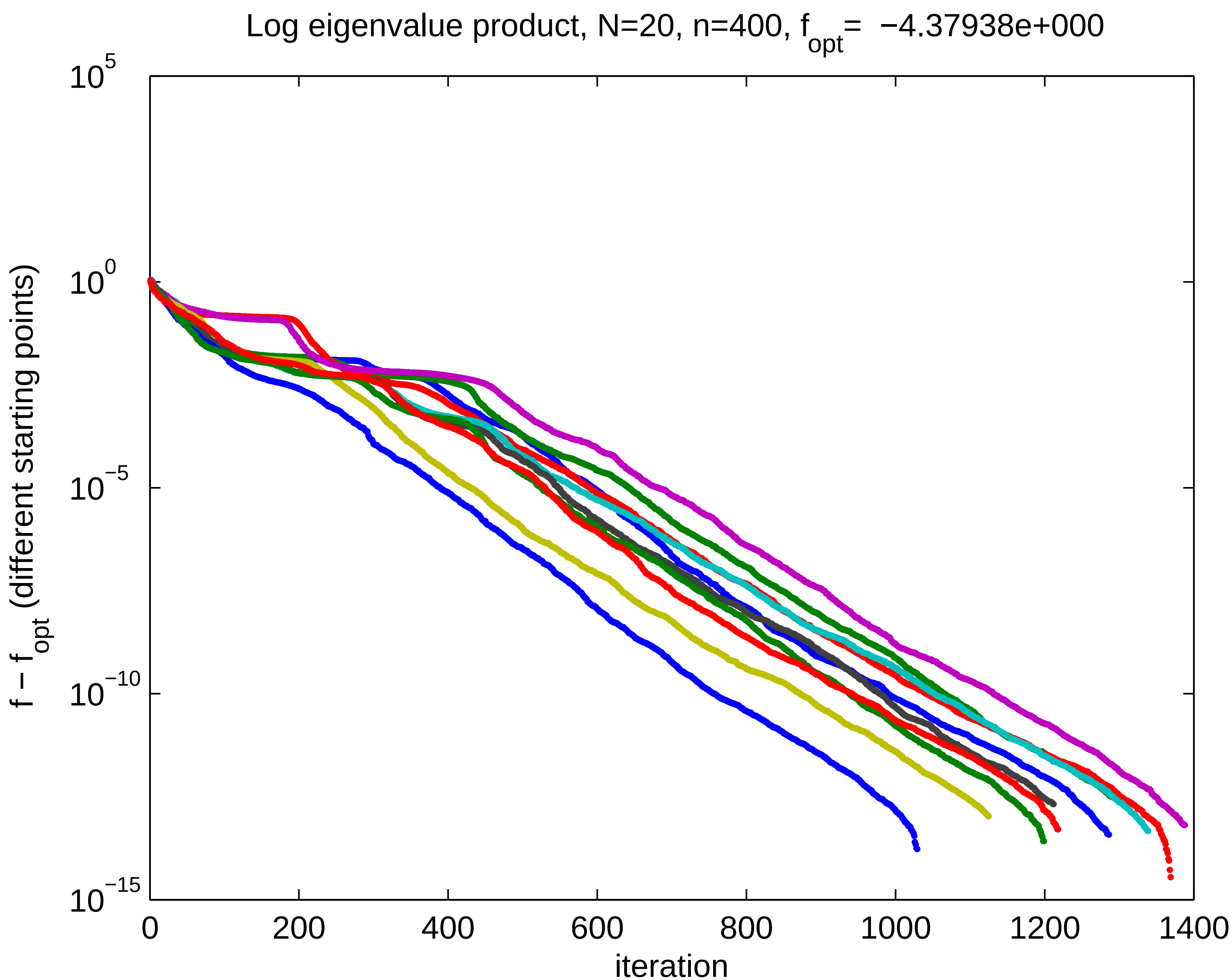
where $A \in S^N$ is fixed and $\circ$ is the Hadamard (componentwise) matrix product, subject to the constraints that X is positive semidefinite and has diagonal entries equal to 1.

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.

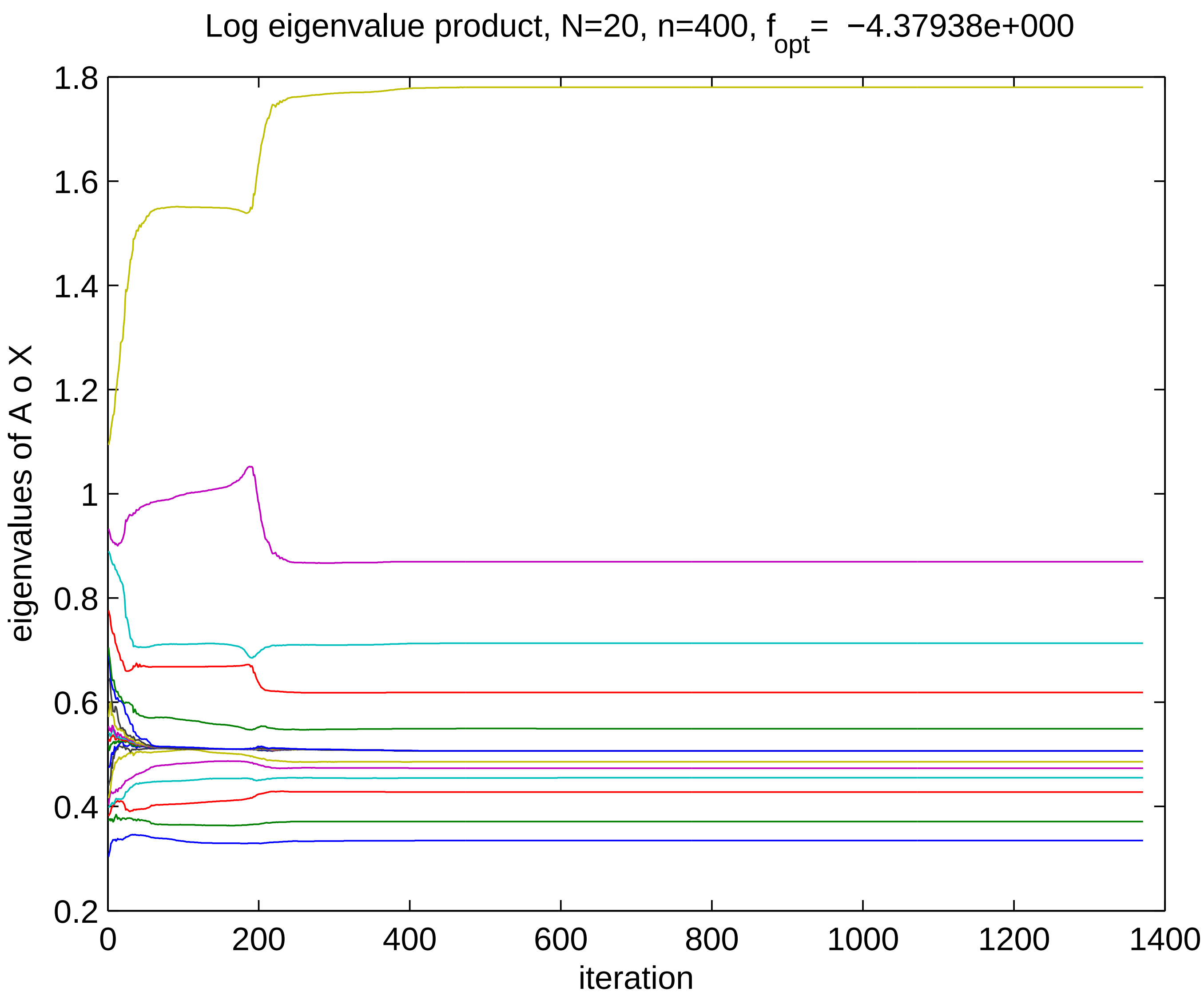
Application: entropy minimization in an environmental application (K.M. Anstreicher and J. Lee, 2004)

BFGS from 10 Randomly Generated Starting Points

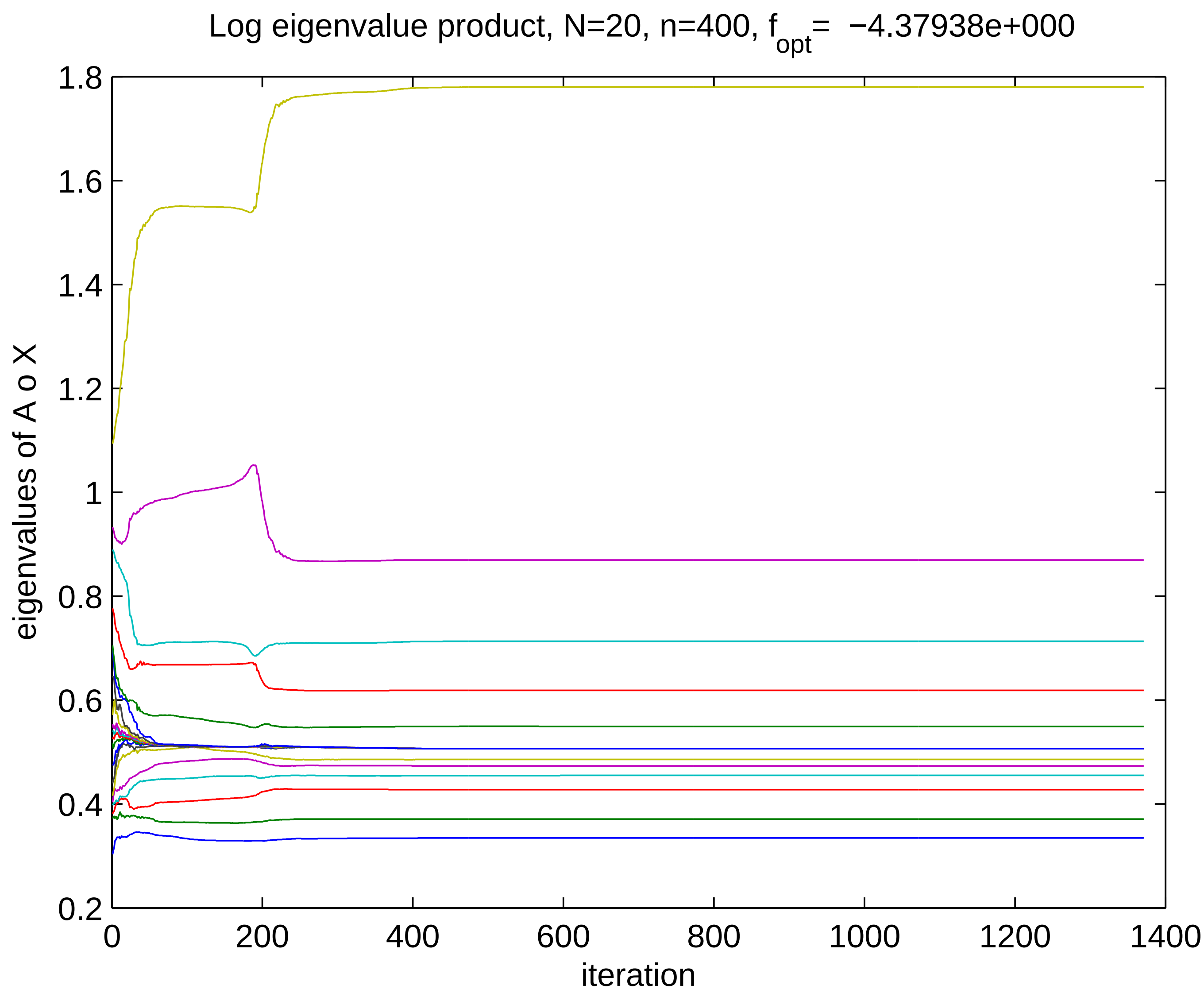


$f - f_{\text{opt}}$, where f_{opt} is least value of f found over all runs

Evolution of Eigenvalues of $A \circ X$

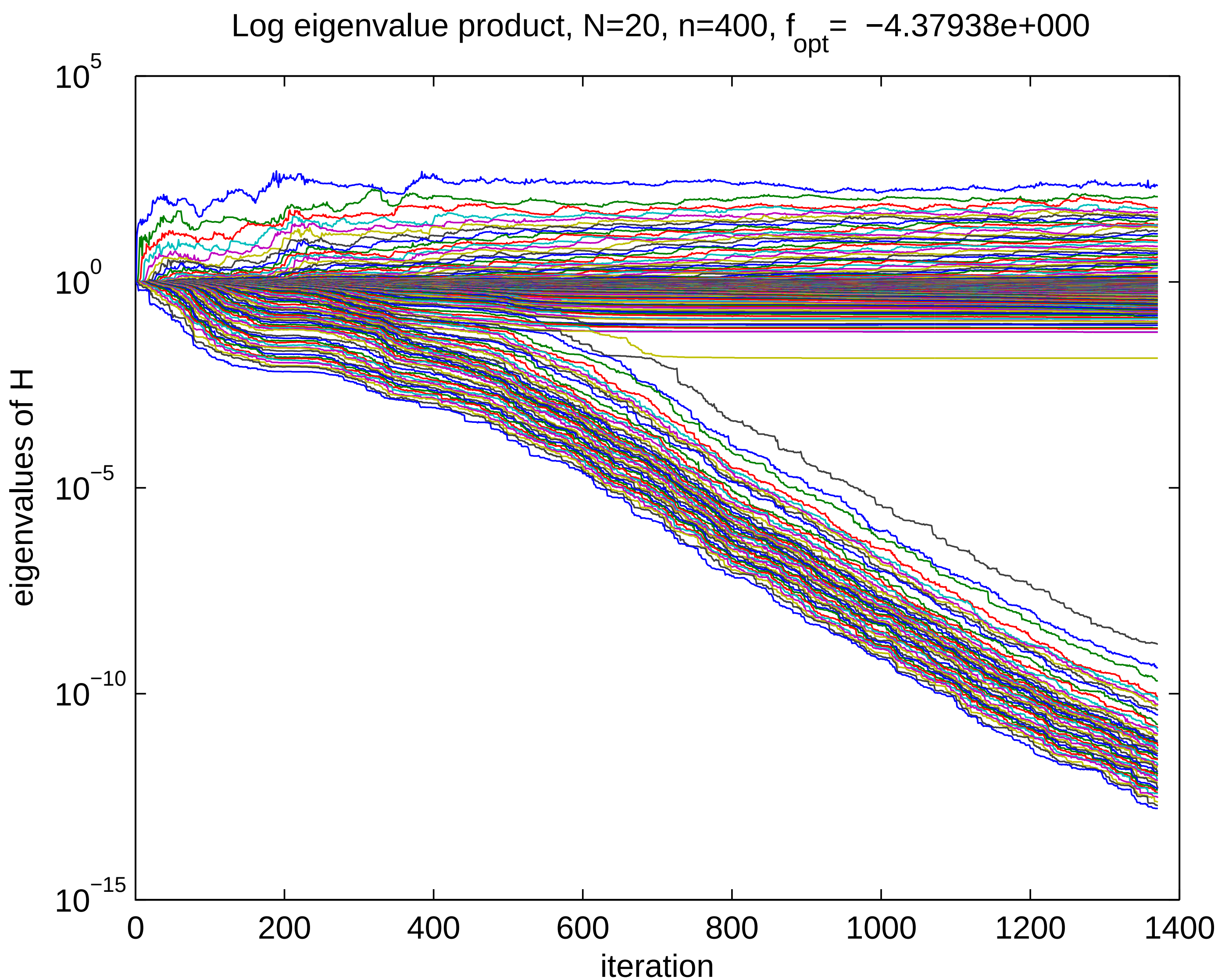


Evolution of Eigenvalues of $A \circ X$

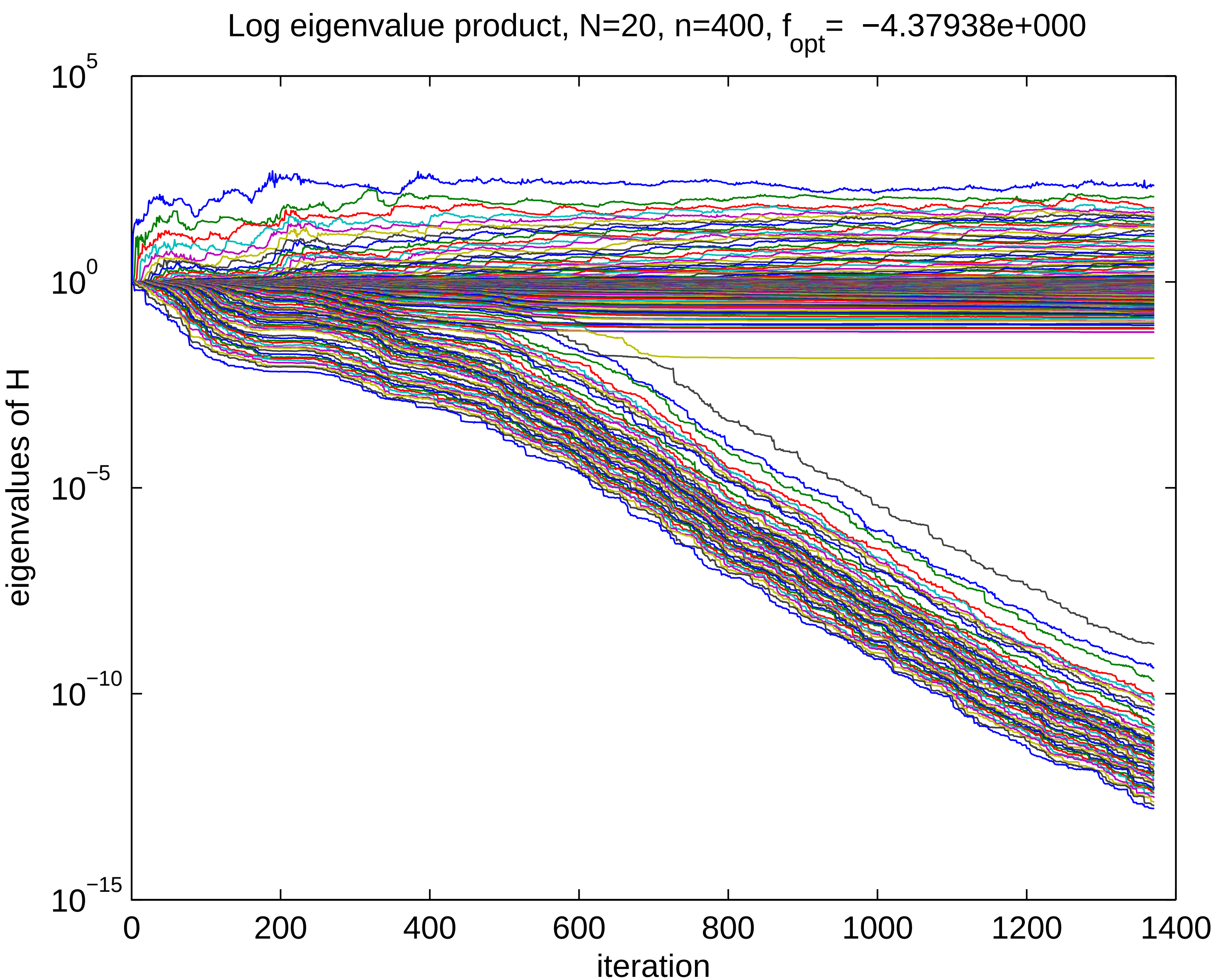


Note that $\lambda_6(X), \ldots, \lambda_{14}(X)$ coalesce

Evolution of Eigenvalues of H



Evolution of Eigenvalues of H



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

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.

And tiny eigenvalues of the BFGS matrix H approximating the "inverse Hessian" correspond to "infinite curvature": nonsmoothness in the V-space

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,

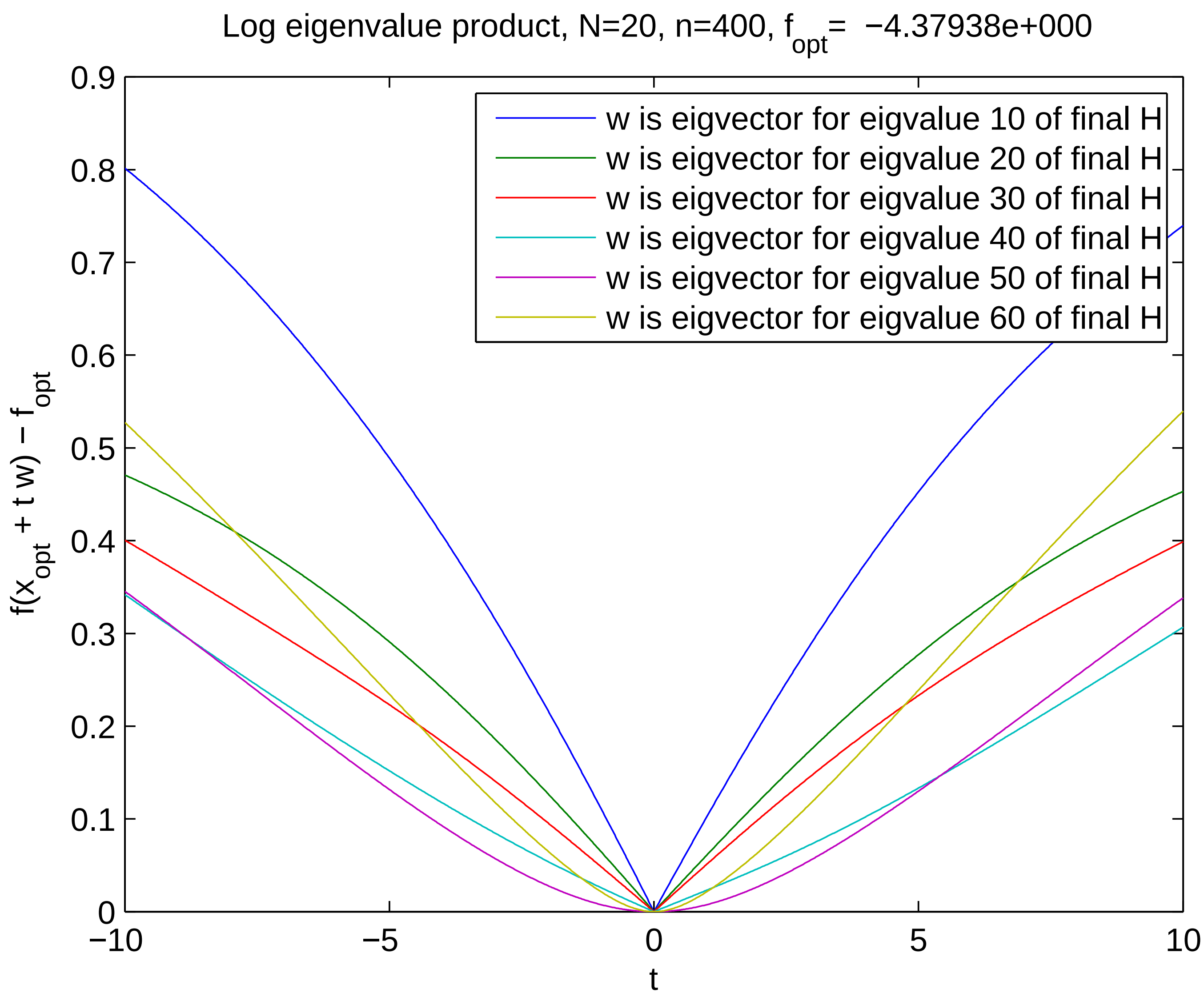
$$\lambda_6(A \circ X) \approx \ldots \approx \lambda_{14}(A \circ X).$$

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.

And tiny eigenvalues of the BFGS matrix H approximating the "inverse Hessian" correspond to "infinite curvature": nonsmoothness in the V-space

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*

Minimizing the Spectral Radius

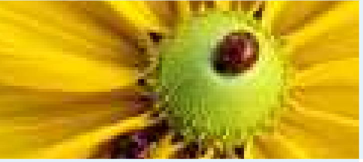
Given the discrete-time dynamical system with control input and measured output

$$z^{(k+1)} = Fz^{(k)} + Gu^{(k)}, \quad y^{(k)} = Hz^{(k)}$$

where $F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p}$, $H \in \mathbb{R}^{m \times n}$, the *static output feedback* problem is to find a controller $X \in \mathbb{R}^{p \times m}$ so that, setting $u^{(k)} = Xy^{(k)}$, all solutions of

$$z^{(k+1)} = (F + GXH)z^{(k)}$$

converge to zero, that is all eigenvalues of $F + GXH$ are inside the unit disk (Schur stable), or prove that this is not possible.



Minimizing the Spectral Radius

Given the discrete-time dynamical system with control input and measured output

$$z^{(k+1)} = Fz^{(k)} + Gu^{(k)}, \quad y^{(k)} = Hz^{(k)}$$

where $F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p}$, $H \in \mathbb{R}^{m \times n}$, the *static output feedback* problem is to find a controller $X \in \mathbb{R}^{p \times m}$ so that, setting $u^{(k)} = Xy^{(k)}$, all solutions of

$$z^{(k+1)} = (F + GXH)z^{(k)}$$

converge to zero, that is all eigenvalues of $F + GXH$ are inside the unit disk (Schur stable), or prove that this is not possible. Pose as optimization problem:

$$\min_{X \in \mathbb{R}^{p \times m}} \rho(F + GXH)$$

where ρ is spectral radius.

Minimizing the Spectral Radius

Given the discrete-time dynamical system with control input and measured output

$$z^{(k+1)} = Fz^{(k)} + Gu^{(k)}, \quad y^{(k)} = Hz^{(k)}$$

where $F \in \mathbb{R}^{n \times n}, G \in \mathbb{R}^{n \times p}$, $H \in \mathbb{R}^{m \times n}$, the *static output feedback* problem is to find a controller $X \in \mathbb{R}^{p \times m}$ so that, setting $u^{(k)} = Xy^{(k)}$, all solutions of

$$z^{(k+1)} = (F + GXH)z^{(k)}$$

converge to zero, that is all eigenvalues of $F + GXH$ are inside the unit disk (Schur stable), or prove that this is not possible.
Pose as optimization problem:

$$\min_{X \in \mathbb{R}^{p \times m}} \rho(F + GXH)$$

where ρ is spectral radius.
NP-hard if add bounds on entries of X
(V. Blondel and J. Tsitsiklis, 1996).

Nonsmooth Analysis of the Spectral Radius

The spectral radius ρ is not locally Lipschitz at matrices with multiple *active* eigenvalues (those attaining the maximal modulus).

Nonsmooth Analysis of the Spectral Radius

The spectral radius ρ is not locally Lipschitz at matrices with multiple *active* eigenvalues (those attaining the maximal modulus).

Nonsmooth analysis of ρ in this case, deriving $\partial^M \rho$, was given by J.V. Burke and M.L.O. (2001), J.V. Burke, A.S. Lewis and M.L.O. (2005), etc.

Nonsmooth Analysis of the Spectral Radius

The spectral radius ρ is not locally Lipschitz at matrices with multiple *active* eigenvalues (those attaining the maximal modulus).

Nonsmooth analysis of ρ in this case, deriving $\partial^M \rho$, was given by J.V. Burke and M.L.O. (2001), J.V. Burke, A.S. Lewis and M.L.O. (2005), etc.

But to apply BFGS, we assume that everywhere we evaluate ρ at $A(X) = F + GXH$, there is just one active real eigenvalue or active conjugate pair with multiplicity one, and break any "ties" arbitrarily.

Gradient of the Spectral Radius

Gradient of the spectral radius in real matrix space:

$$\nabla \rho(\tilde{A}) = \mathrm{Re} \frac{\mu}{|\mu|} \frac{1}{v^* u} v u^*$$

where v and u are right and left eigenvectors for the relevant active eigenvalue μ of $\tilde{A}$, which is assumed to be simple and have nonnegative imaginary part.

Gradient of the Spectral Radius

Gradient of the spectral radius in real matrix space:

$$\nabla \rho(\tilde{A}) = \mathrm{Re} \frac{\mu}{|\mu|} \frac{1}{v^* u} v u^*$$

where v and u are right and left eigenvectors for the relevant active eigenvalue μ of $\tilde{A}$, which is assumed to be simple and have nonnegative imaginary part.

Gradients may be arbitrarily large for μ nearly a multiple eigenvalue: spectral functions are not locally Lipschitz at an active multiple eigenvalue.

Gradient of the Spectral Radius

Gradient of the spectral radius in real matrix space:

$$\nabla \rho(\tilde{A}) = \mathrm{Re} \frac{\mu}{|\mu|} \frac{1}{v^* u} v u^*$$

where v and u are right and left eigenvectors for the relevant active eigenvalue μ of $\tilde{A}$, which is assumed to be simple and have nonnegative imaginary part.

Gradients may be arbitrarily large for μ nearly a multiple eigenvalue: spectral functions are not locally Lipschitz at an active multiple eigenvalue.

Break ties for active eigenvalue arbitrarily.

Gradient of the Spectral Radius

Gradient of the spectral radius in real matrix space:

$$\nabla \rho(\tilde{A}) = \mathrm{Re} \frac{\mu}{|\mu|} \frac{1}{v^* u} v u^*$$

where v and u are right and left eigenvectors for the relevant active eigenvalue μ of $\tilde{A}$, which is assumed to be simple and have nonnegative imaginary part.

Gradients may be arbitrarily large for μ nearly a multiple eigenvalue: spectral functions are not locally Lipschitz at an active multiple eigenvalue.

Break ties for active eigenvalue arbitrarily.

Since $\tilde{A}$ is real, take $\mathrm{Im}\ \mu \geq 0$ wlog.

Gradient of the Spectral Radius

Gradient of the spectral radius in real matrix space:

$$\nabla \rho(\tilde{A}) = \mathrm{Re} \frac{\mu}{|\mu|} \frac{1}{v^* u} v u^*$$

where v and u are right and left eigenvectors for the relevant active eigenvalue μ of $\tilde{A}$, which is assumed to be simple and have nonnegative imaginary part.

Gradients may be arbitrarily large for μ nearly a multiple eigenvalue: spectral functions are not locally Lipschitz at an active multiple eigenvalue.

Break ties for active eigenvalue arbitrarily.

Since $\tilde{A}$ is real, take $\mathrm{Im}\, \mu \geq 0$ wlog.

Defining $A(X) = F + GXH$, use ordinary chain rule to obtain gradients of $\rho(A(X))$ in the X space.

Numerical Results for some SOF Problems

Let F be an $n \times n$ Toeplitz matrix whose nonzeros are 0.5 on the main diagonal and first three superdiagonals and and the number -0.5 on the first subdiagonal. Not Schur stable.

Numerical Results for some SOF Problems

Let F be an $n \times n$ Toeplitz matrix whose nonzeros are 0.5 on the main diagonal and first three superdiagonals and and the number -0.5 on the first subdiagonal. Not Schur stable.

First set of experiments: set $n = 8$ and optimize over $X \in \mathbb{R}^{p \times m}$ with $p = 1$ (setting $G = [1, \ldots, 1]^T$), and consider m ranging from 0 to 8 (setting H to the matrix whose rows are the first m rows of the identity matrix).

Numerical Results for some SOF Problems

Let F be an $n \times n$ Toeplitz matrix whose nonzeros are 0.5 on the main diagonal and first three superdiagonals and and the number -0.5 on the first subdiagonal. Not Schur stable.

First set of experiments: set $n = 8$ and optimize over $X \in \mathbb{R}^{p \times m}$ with $p = 1$ (setting $G = [1, \ldots, 1]^T$), and consider m ranging from 0 to 8 (setting H to the matrix whose rows are the first m rows of the identity matrix).

For each m, run BFGS from 100 randomly generated starting points to search for local minimizers of $\rho(F + GXH)$ over X and plot eigenvalues of $F + GXH$ for the best X found.

Numerical Results for some SOF Problems

Let F be an $n \times n$ Toeplitz matrix whose nonzeros are 0.5 on the main diagonal and first three superdiagonals and and the number -0.5 on the first subdiagonal. Not Schur stable.

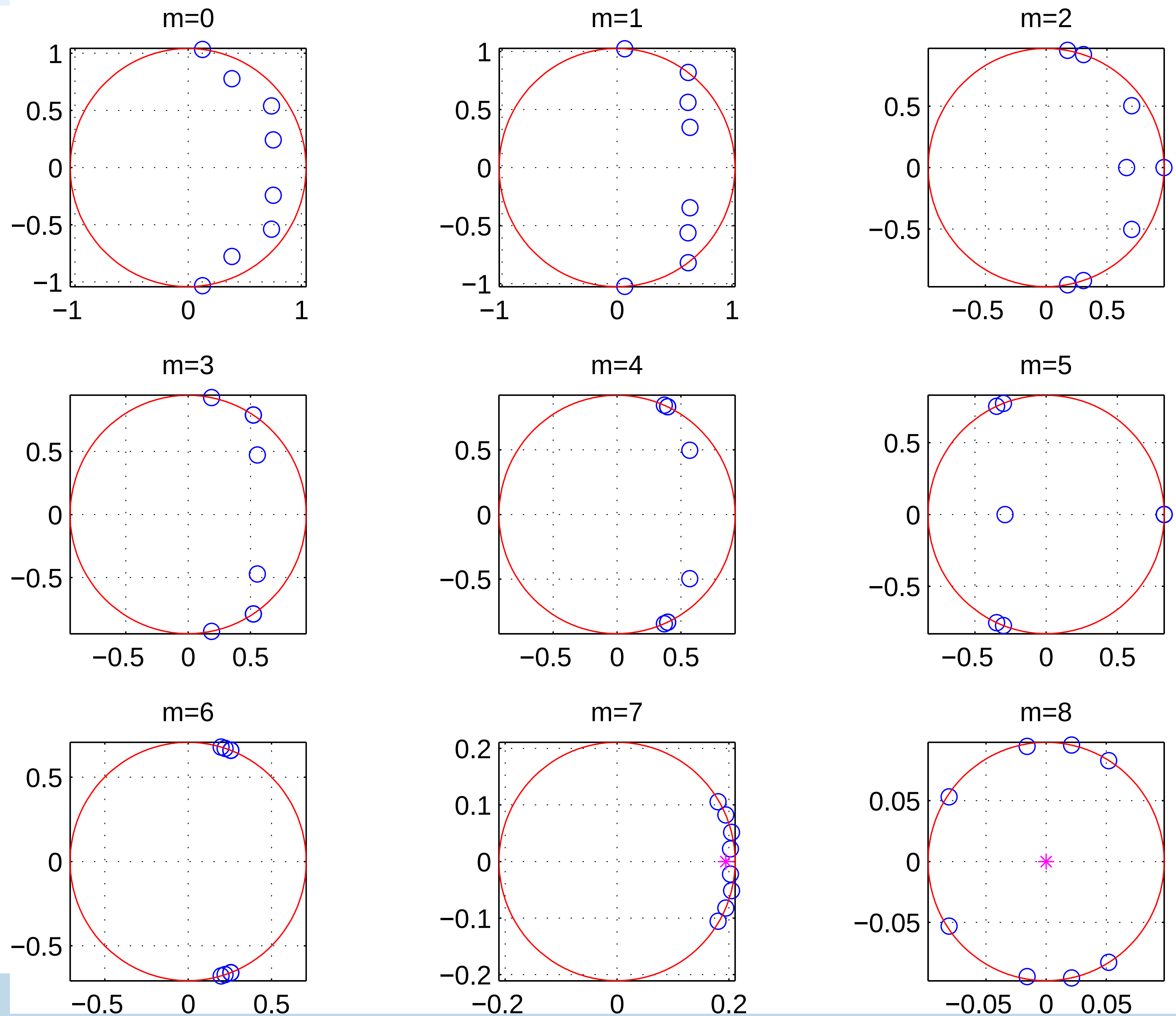
First set of experiments: set $n = 8$ and optimize over $X \in \mathbb{R}^{p \times m}$ with $p = 1$ (setting $G = [1, \ldots, 1]^T$), and consider m ranging from 0 to 8 (setting H to the matrix whose rows are the first m rows of the identity matrix).

For each m, run BFGS from 100 randomly generated starting points to search for local minimizers of $\rho(F + GXH)$ over X and plot eigenvalues of $F + GXH$ for the best X found.

Second set of experiments: $n = 15$, $p = 2$, with G having a second column $[1, -1, 1, -1, ..., 1]^T$.

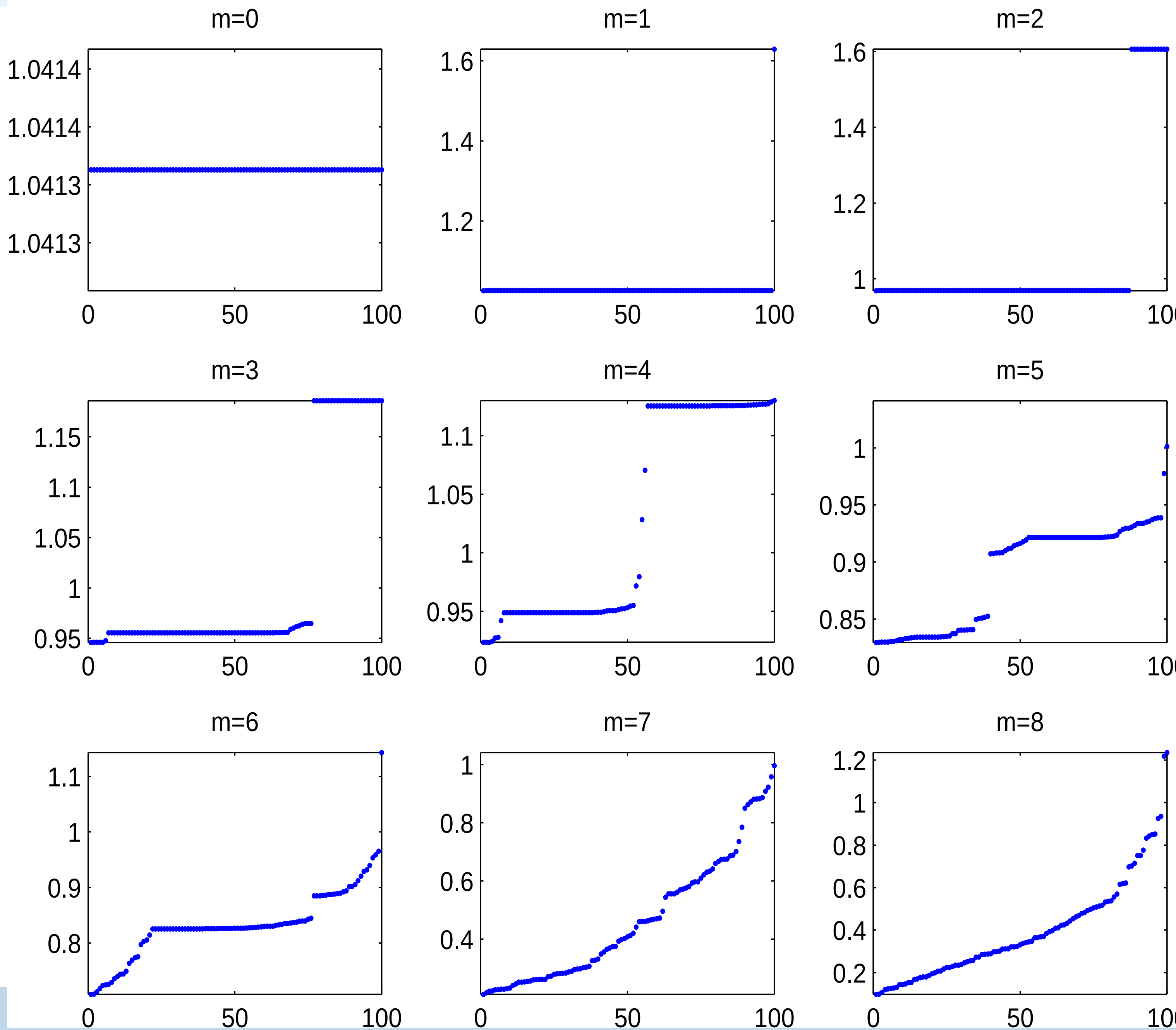
Yurii Nesterov

Optimized Eigenvalues: $n = 8$, $p = 1$

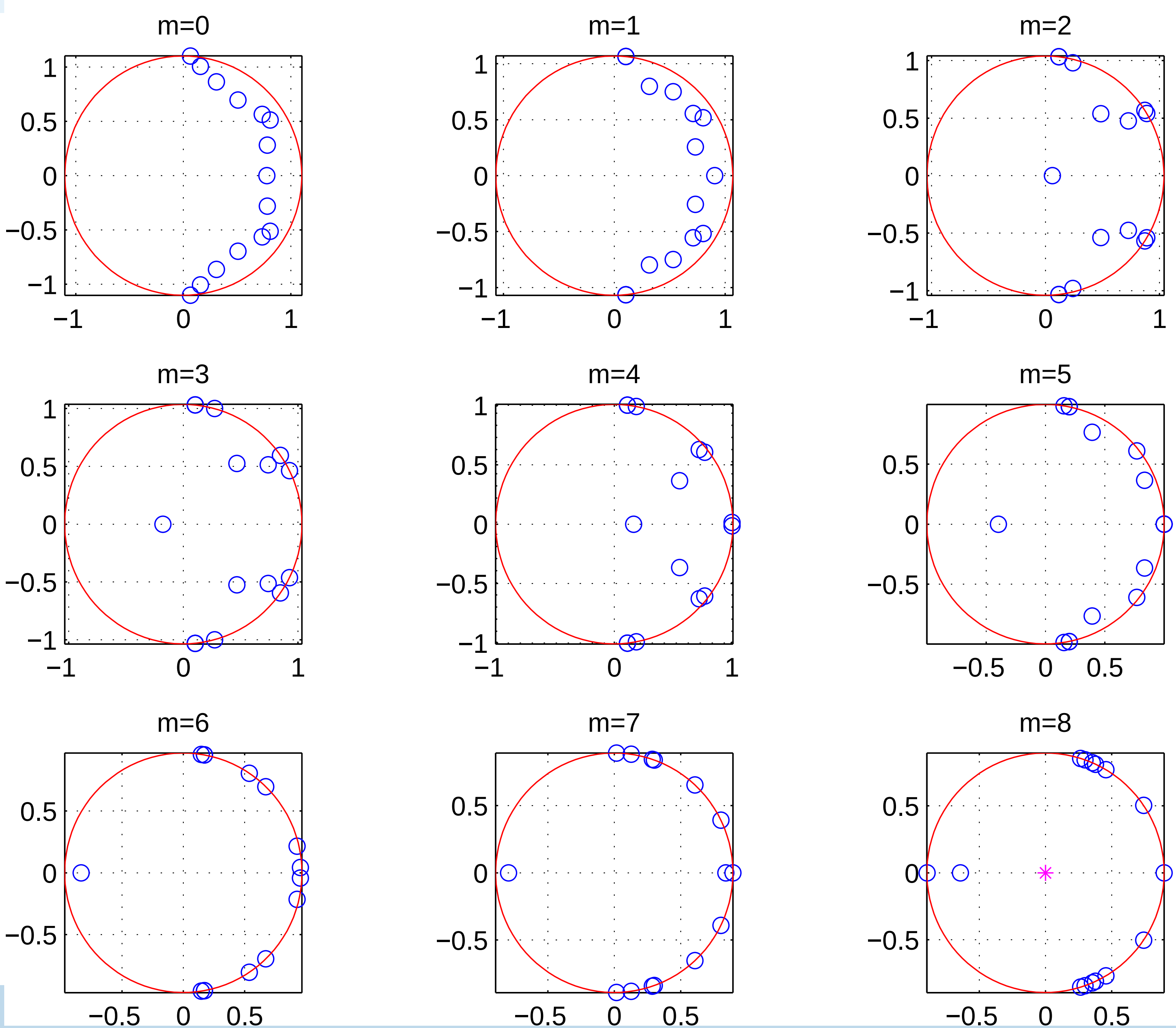


'*' : known optimal value for $m = 7$ and $m = 8$

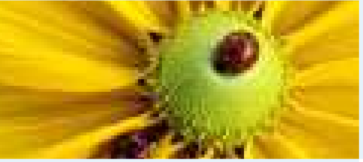
Sorted Final Values of ρ for 100 Runs of BFGS



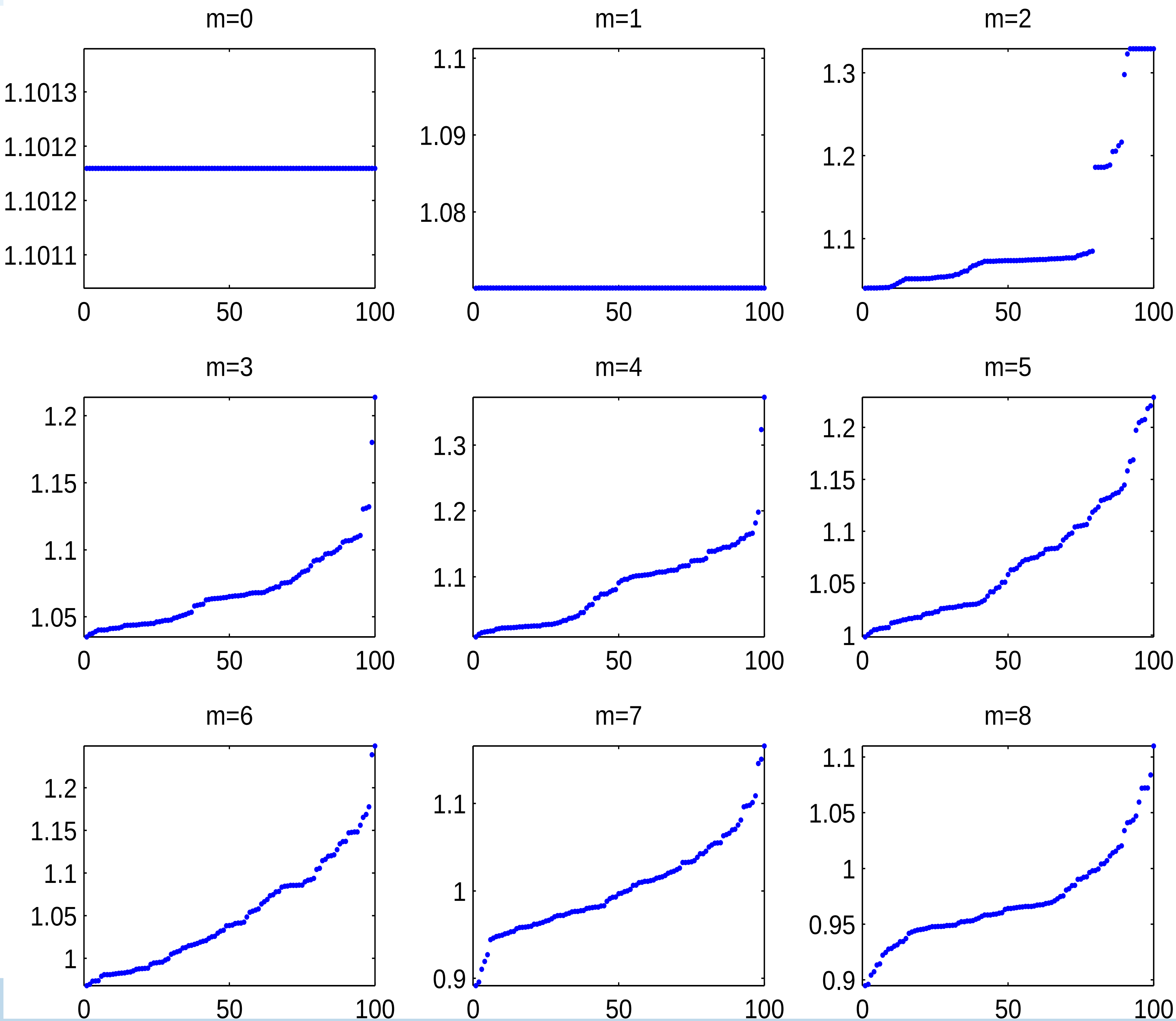
Optimized Eigenvalues: $n = 15$, $p = 2$



'*' : known optimal value for $m = 8$



Sorted Final Values of ρ for 100 Runs of BFGS



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:

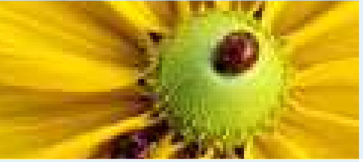
Challenge: Convergence of BFGS in Nonsmooth Case

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 $\bar{x}$

A.S. Lewis and M.L.O., Math Programming, 2013.

And Finally

Happy Birthday Yurii!