# New Quasi-Newton Ideas for (Non)smooth Optimization

**Frank E. Curtis**, Lehigh University

presented at

International Conference on Continuous Optimization

Berlin, Germany

August 8, 2019


# Acknowledgments






# References




⋆ A. Berahas, F. E. Curtis, and B. Zhou.
Limited-Memory BFGS with Displacement Aggregation.
arXiv 1903.03471, 2019.

⋆ F. E. Curtis, D. P. Robinson, and B. Zhou.
A Self-Correcting Variable-Metric Algorithm Framework for Nonsmooth Optimization.
*IMA Journal of Numerical Analysis*, 10.1093/imanum/drz008, 2019.

# Outline

# Outline

# This talk

This is a talk about algorithm efficiency.

Much of my work: exploiting inexactness for scalable constrained optimization.

SQP:

- Byrd, Curtis, & Nocedal, 2008 & 2010
- Curtis, Nocedal, & Wächter, 2009
- Curtis, Johnson, Robinson, & Wächter, 2014

Interior-point:

- Curtis, Schenk, & Wächter, 2010; w/ Huber, 2012
- Curtis, Gould, Robinson, & Toint, 2017

Augmented Lagrangian:

- Curtis, Jiang, & Robinson, 2015; w/ Gould, 2016

# Practical efficiency, not worst-case complexity

I have also worked on worst-case complexity for nonconvex optimization.

Achieving good/optimal complexity for practical algorithms.

- Curtis, Robinson, & Samadi, 2017 & 2018
- Curtis, Lubberts, & Robinson, 2018
- Curtis & Robinson, 2019
- Newton-CG (forthcoming)

Regional complexity analysis for nonconvex optimization.

- Curtis & Robinson, 2018

# Efficiency

This talk is about efficient passing of information.

Aggregation in quasi-Newton methods

- Filling in the gap between L-BFGS and BFGS

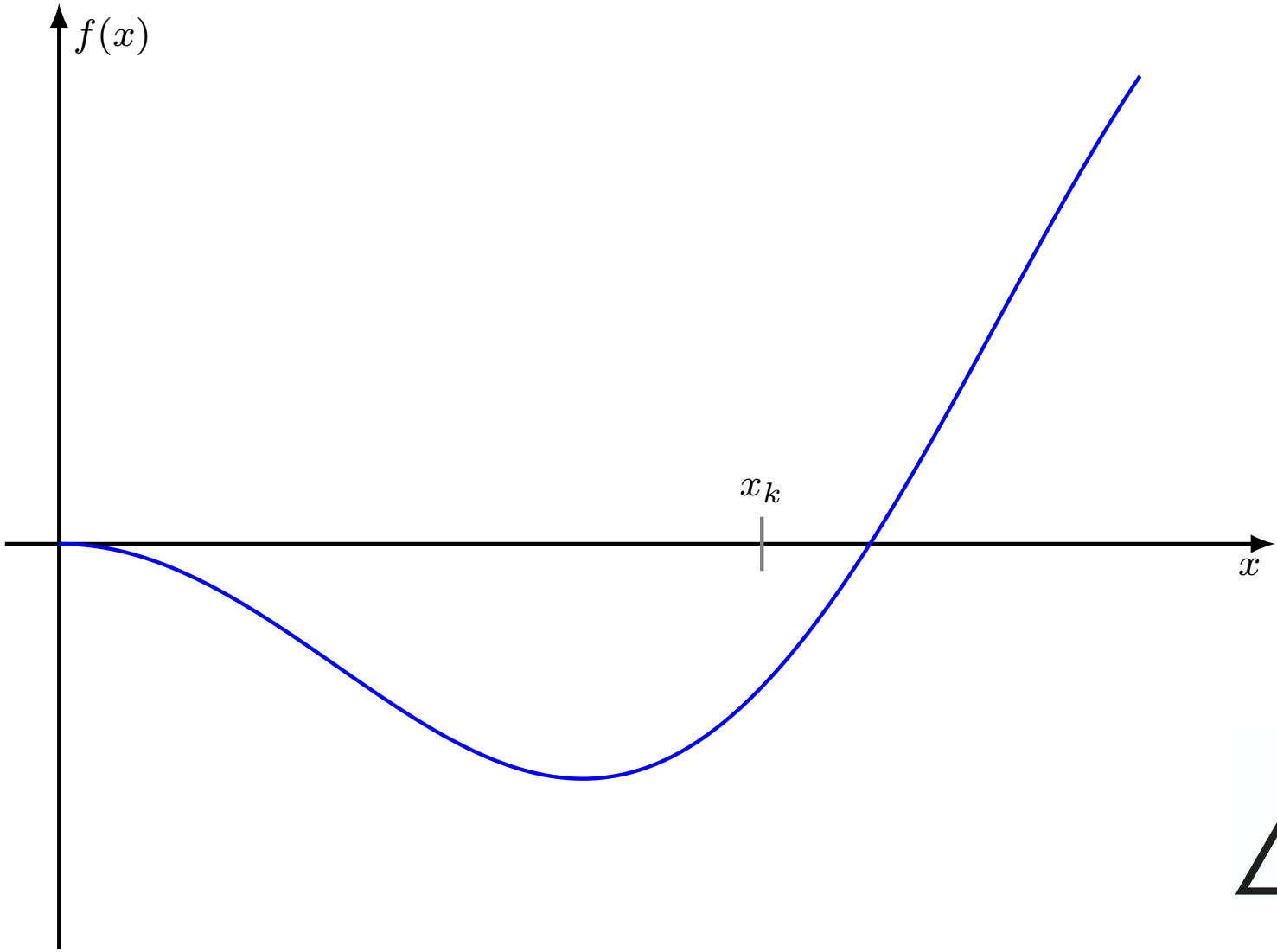
Quasi-Newton methods for nonsmooth optimization

- Adaptivity, inexactness, and aggregation

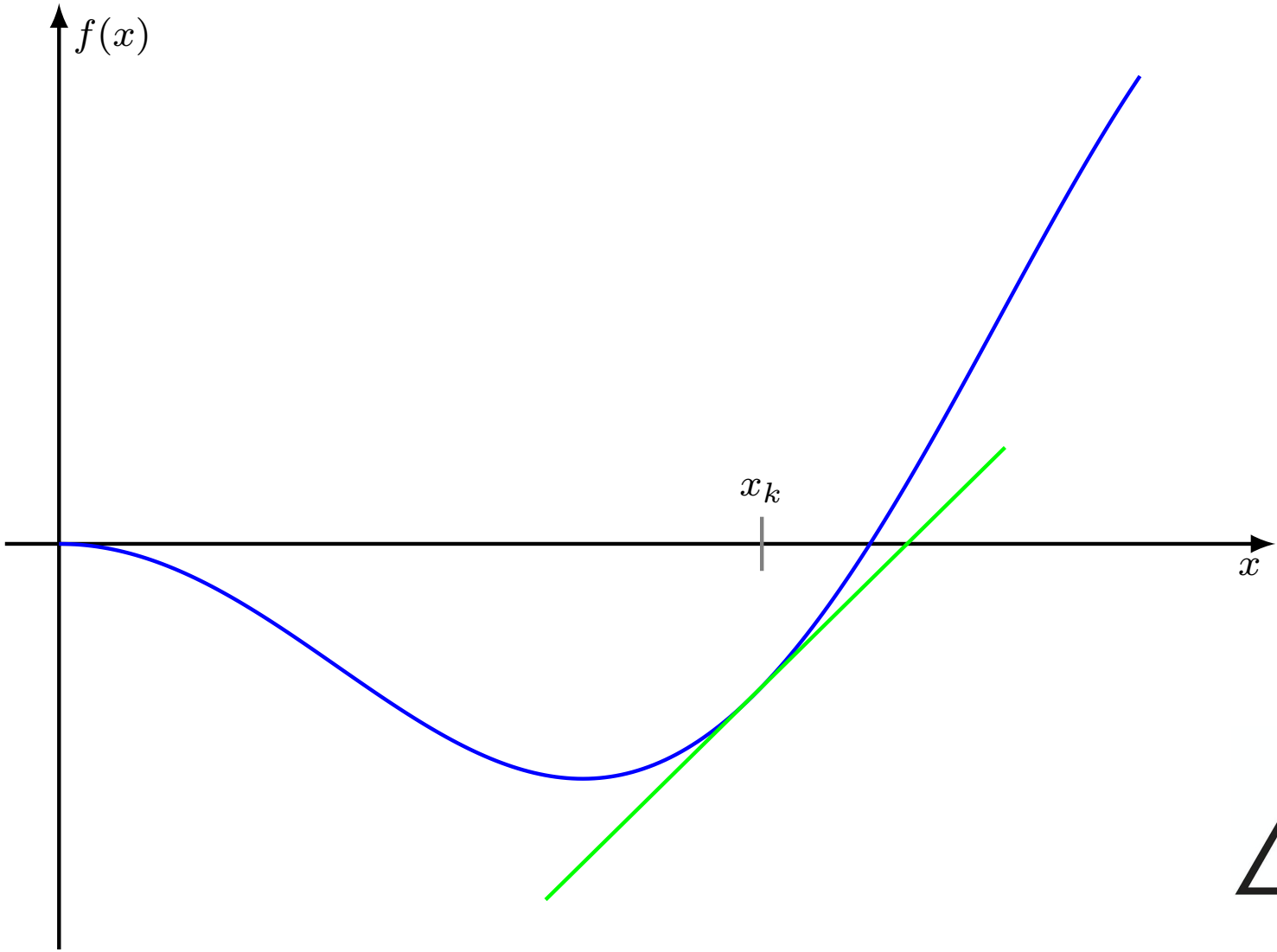
[conveying information in a compact form]

# Outline

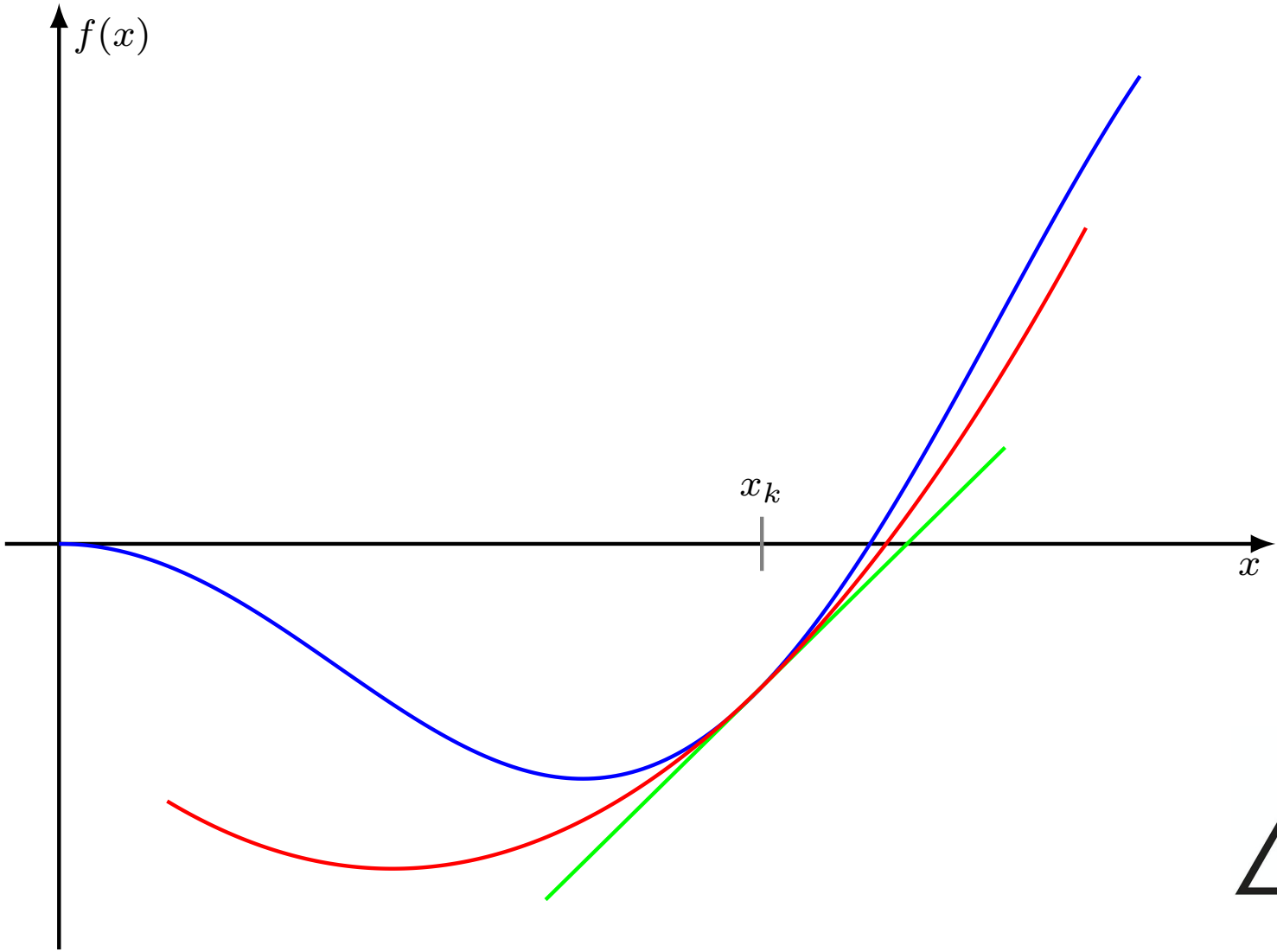
# Motivation by illustration



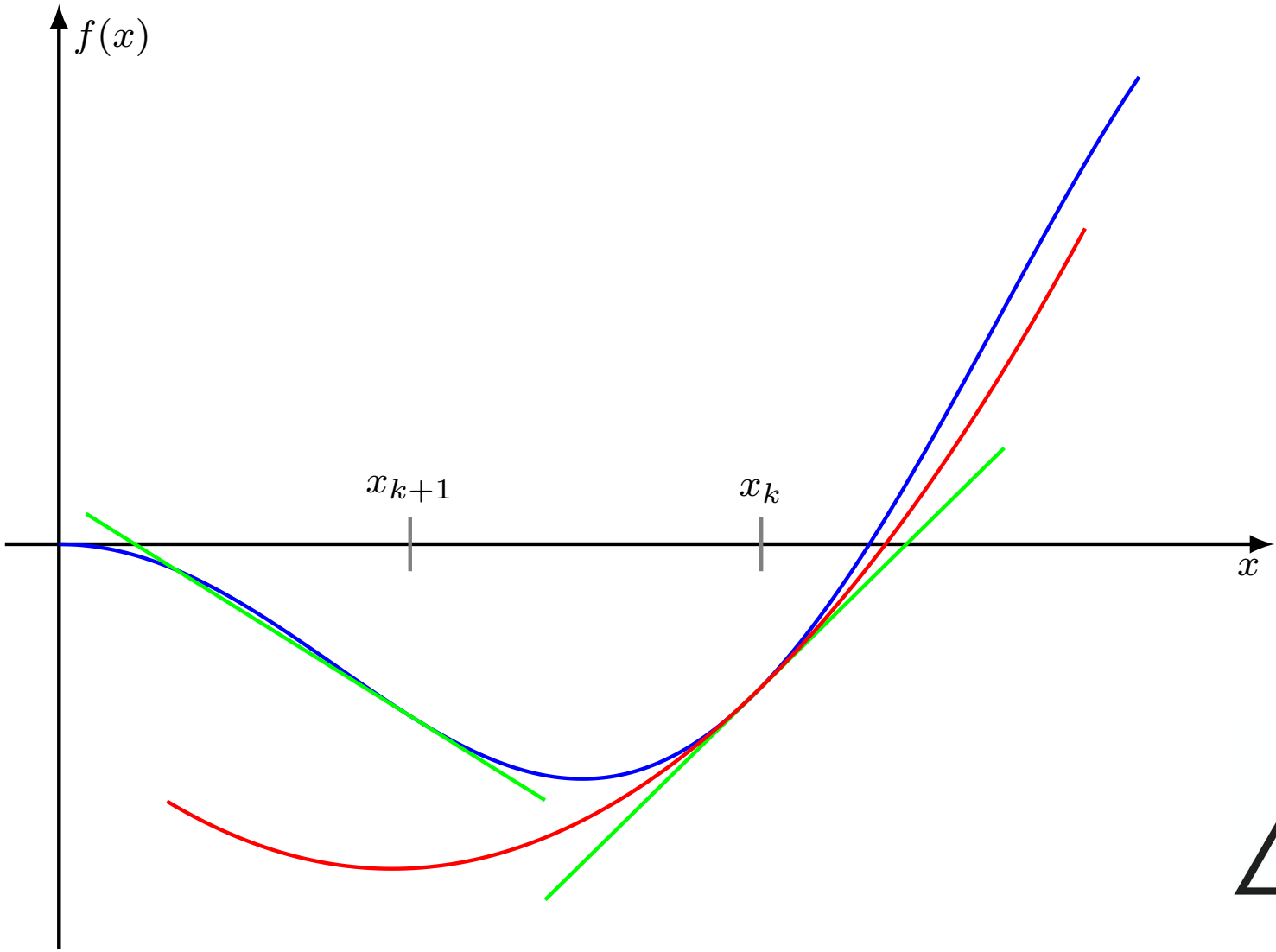

# Motivation by illustration



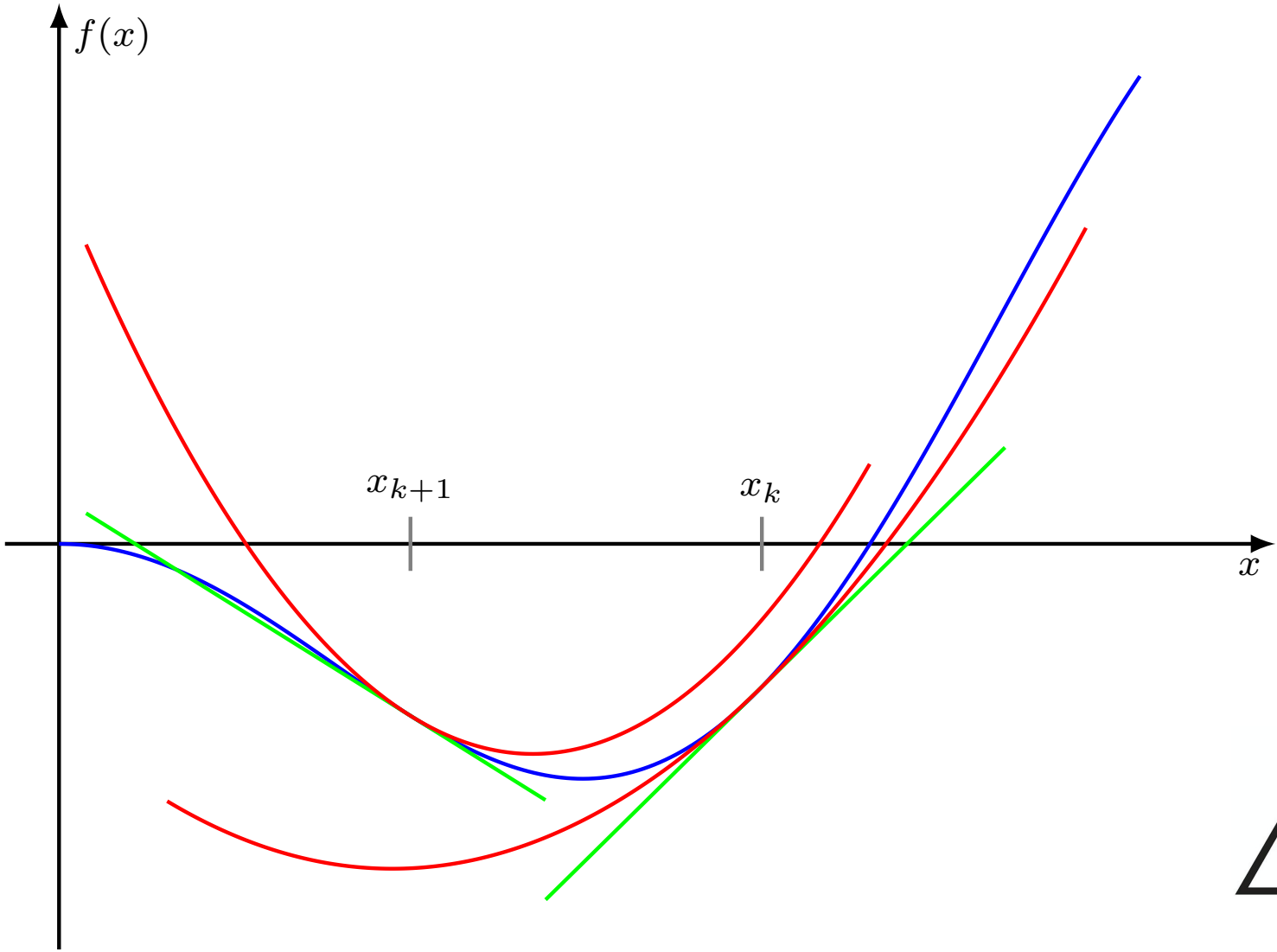

# Motivation by illustration




# Motivation by illustration




# Motivation by illustration




# Notation

$$
\begin{aligned}
x_{k+1} - x_k =: s_k &: \text{ iterate displacement} \\
\nabla f(x_{k+1}) - \nabla f(x_k) =: y_k &: \text{ gradient displacement} \\
H_k &: \text{ Hessian approximation} \\
W_k &: \text{ inverse Hessian approximation}
\end{aligned}
$$

# BFGS update

Minimal deviation from $W_k$ subject to secant equation:

$$\min_{W \in \mathbb{R}^{n \times n}} \|W - W_k\|$$
$$\text{s.t. } W = W^T,\ W y_k = s_k$$

Using weighted Frobenius norm (w/ weight matrix satisfying secant equation):

$$W_{k+1} \leftarrow \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right)^T W_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}$$

Using the Sherman-Morrison-Woodbury formula:

$$H_{k+1} \leftarrow \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)^T H_k \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right) + \frac{y_k y_k^T}{s_k^T y_k}$$

# Geometric properties of Hessian update: Burke, Lewis, Overton (2007)

Consider the matrices (which only depend on $s_k$ and $H_k$):

$$P_k := \frac{s_k s_k^T H_k}{s_k^T H_k s_k} \quad \text{and} \quad Q_k := I - P_k.$$

Both $H_k$-orthogonal projection matrices (i.e., idempotent and $H_k$-self-adjoint).

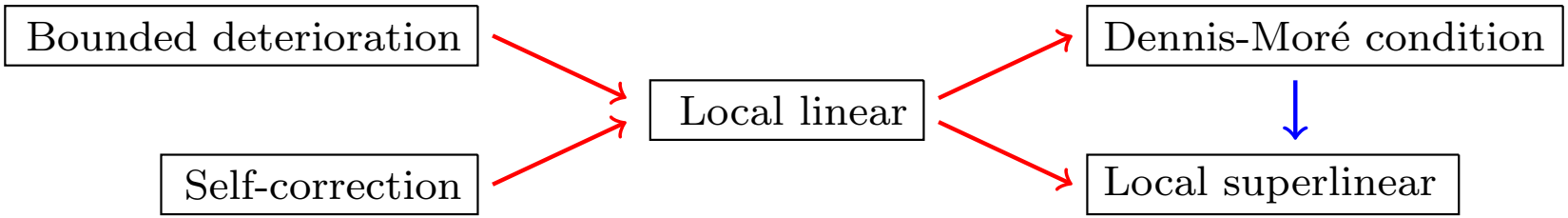
- $P_k$ yields $H_k$-orthogonal projection onto $\operatorname{span}(s_k)$.
- $Q_k$ yields $H_k$-orthogonal projection onto $\operatorname{span}(s_k)^{\perp_{H_k}}$.

$$H_{k+1} \leftarrow \underbrace{\left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)^T H_k \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)}_{\text{rank } n-1} + \underbrace{\frac{y_k y_k^T}{s_k^T y_k}}_{\text{rank } 1}$$

- Curvature projected out along $\operatorname{span}(s_k)$
- Curvature corrected by $\frac{y_k y_k^T}{s_k^T y_k} = \left(\frac{y_k y_k^T}{\|y_k\|_2^2}\right)\left(\frac{\|y_k\|_2^2}{y_k^T W_{k+1} y_k}\right)$ (inverse Rayleigh).

# Theory of BFGS

BFGS can be superlinearly convergent, but study preliminary theory!




- Broyden, Dennis, & Moré, 1973
- Dennis & Moré, 1974
- Powell, 1976
- Werner, 1978
- Ritter, 1979 & 1981
- Byrd & Nocedal, 1987

# Bounded deterioration

Proved for the Hessian and inverse Hessian forms of the update. Let

$$\|M\|_{\text{BFGS}} \equiv \|\nabla^2 f(x_*)^{1/2} M \nabla^2 f(x_*)^{1/2}\|_F.$$

## Theorem 1 (Bounded deterioration (simplified))

*Suppose $f$ is twice continuously differentiable with Lipschitz continuous Hessian. Then, for all $k \in \mathbb{N}$ and some $(\alpha, \beta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$, one finds*

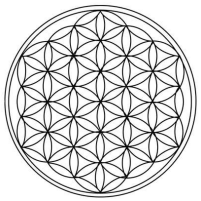
$$\|W_{k+1} - \nabla^2 f(x_*)^{-1}\|_{\text{BFGS}} \leq (1 + \alpha\sigma_k)\|W_k - \nabla^2 f(x_*)^{-1}\|_{\text{BFGS}} + \beta\sigma_k,$$

*where*

$$\sigma_k := \max\{\|x_k - x_*\|_2, \|x_{k+1} - x_*\|_2\}.$$

- Implies $\|x_{k+1} - x_*\|_2 \leq r\|x_k - x_*\|_2$ for any $r \in (0, 1)$.
- Surprisingly, $W_1 \approx \nabla^2 f(x_*)^{-1}$ not needed for superlinear convergence.

# Convergence of BFGS approximations



Do BFGS approximations converge to the true (inverse) Hessian?

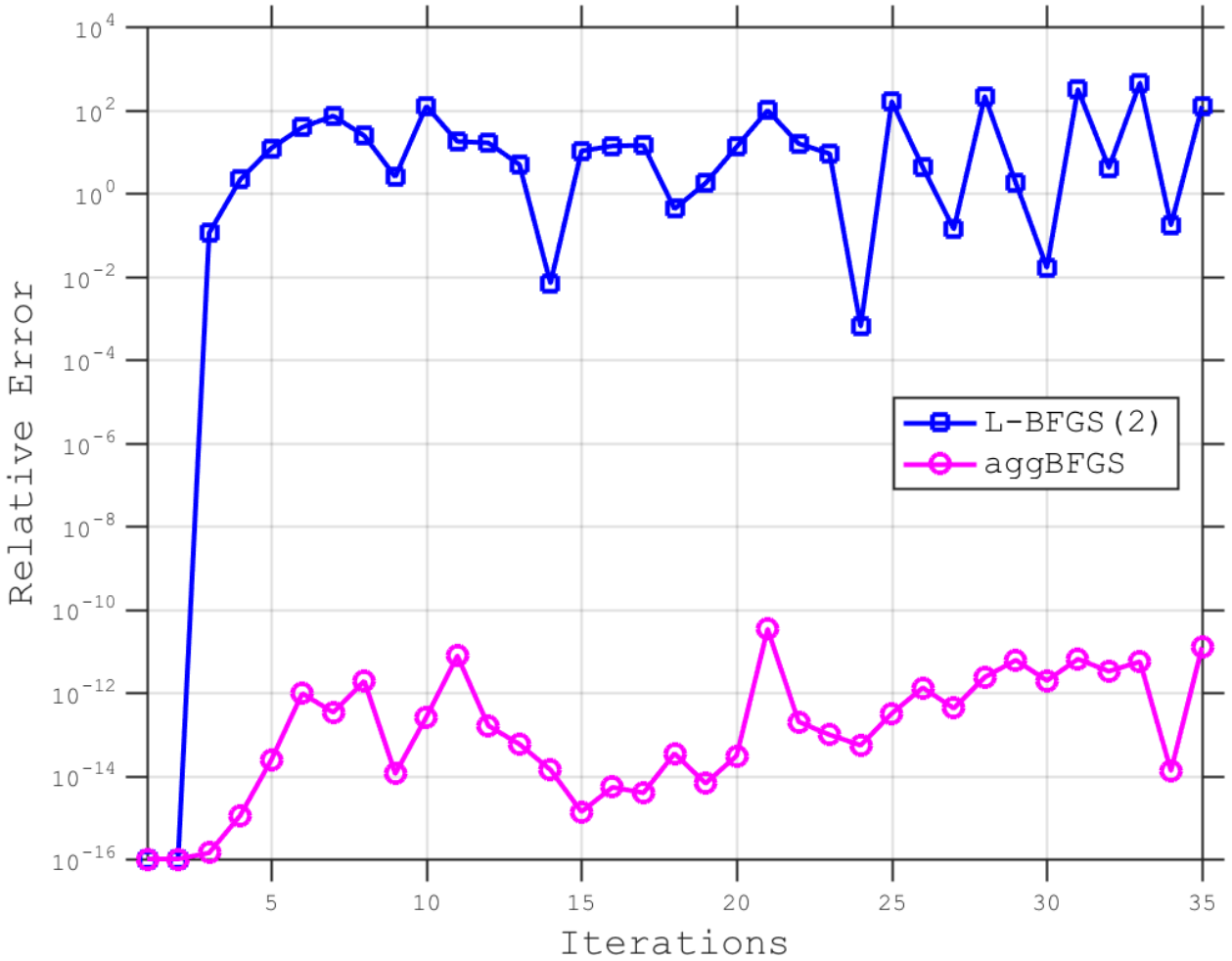
- Not always, even for quadratics; Ge & Powell, 1983.
- SR1?

# Outline

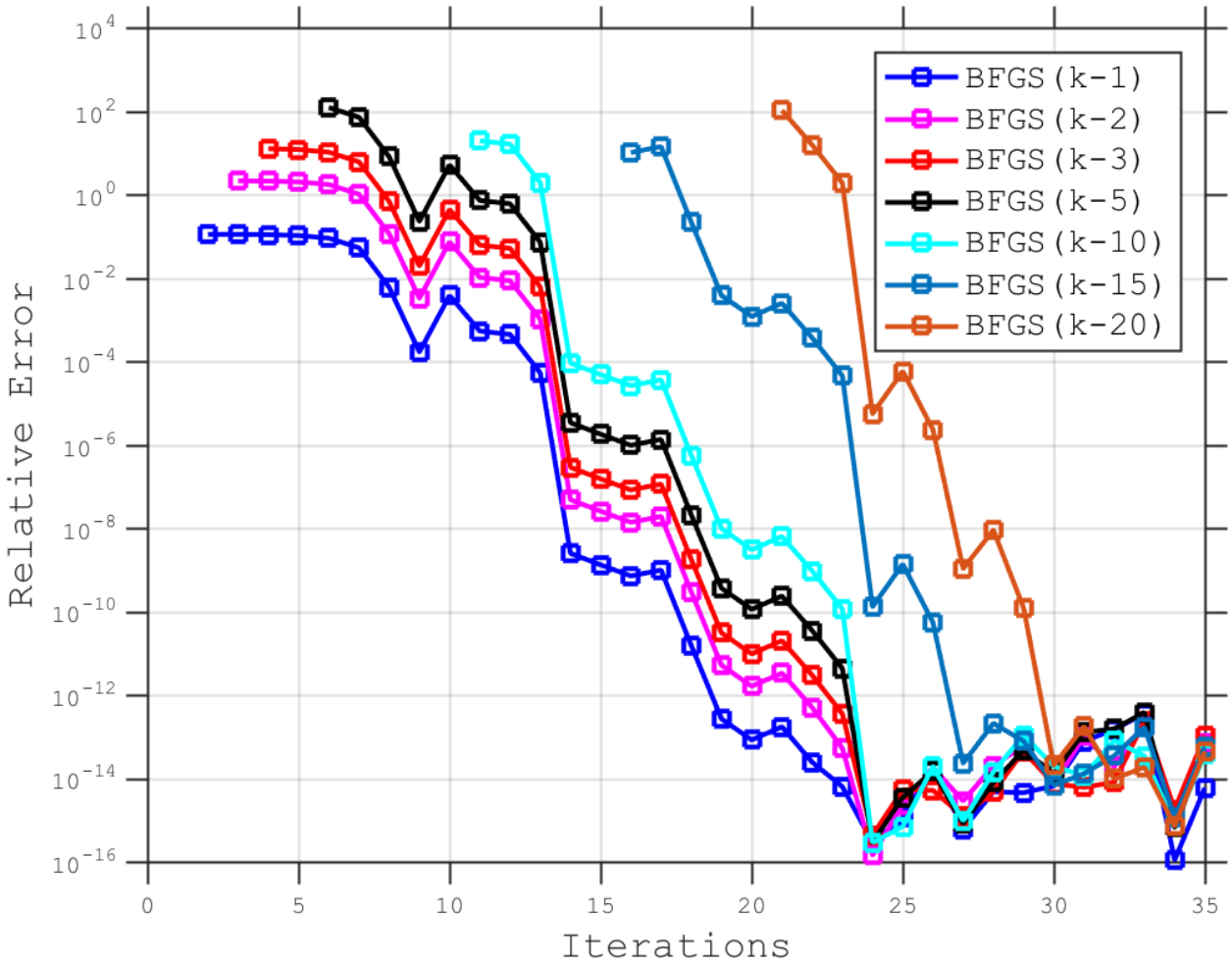
# Motivating questions

- What lies *between* L-BFGS (linear) and BFGS (superlinear)?
- ... can increase $m$, but do we need $m \to \infty$ to achieve superlinearity?
- Does L-BFGS($n$) behave equivalently to BFGS?
- No, but can we *aggregate* information?
- ... so *Agg*-BFGS($m$) $\equiv$ BFGS (with $m \leq n$)?

# Is L-BFGS($n$) $\equiv$ BFGS?



# How long does information from early pairs *linger*?



# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k)}_{\text{“stored”}}$

L-BFGS:

Agg-BFGS:

# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS:

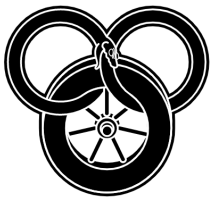
Agg-BFGS:

# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k)}_{\text{stored}}$

Agg-BFGS:

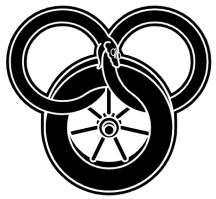


# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS: $\underbrace{(s_0, y_0)}_{\text{lost}}, \underbrace{(s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{stored}}$

Agg-BFGS:



# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS: $\underbrace{(s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{stored}}$

Agg-BFGS:

# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{"stored"}}$

L-BFGS: $\underbrace{(s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{stored}}$

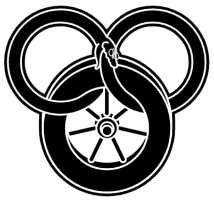
Agg-BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k)}_{\text{stored}}$

# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS: $\underbrace{(s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{stored}}$

Agg-BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{pre-aggregation}}$

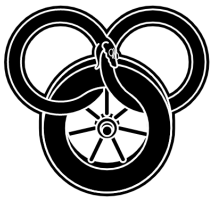


# BFGS vs. L-BFGS vs. Agg-BFGS

BFGS: $\underbrace{(s_0, y_0), (s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{“stored”}}$

L-BFGS: $\underbrace{(s_1, y_1), \ldots, (s_k, y_k), (s_{k+1}, y_{k+1})}_{\text{stored}}$

Agg-BFGS: $\underbrace{(s_1, \tilde{y}_1), \ldots, (s_k, \tilde{y}_k), (s_{k+1}, \tilde{y}_{k+1})}_{\text{aggregated}}$



# Parallel consecutive iterate displacements

$$\begin{aligned}
\text{BFGS}(W, S_{1:m}, Y_{1:m}): &\ \text{BFGS matrix with initial } W \succ 0 \text{ and pairs in} \\
S_{1:m}: &\ \begin{bmatrix} s_1 & \cdots & s_m \end{bmatrix} \\
Y_{1:m}: &\ \begin{bmatrix} y_1 & \cdots & y_m \end{bmatrix} \\
\text{where } \rho: &\ \begin{bmatrix} 1/(s_1^T y_1) & \cdots & 1/(s_m^T y_m) \end{bmatrix}^T > 0
\end{aligned}$$

## Theorem 2

*Suppose* $s_j = \tau s_{j+1}$ *for some* $j \in \{1, \ldots, m-1\}$ *and* $\tau \in \mathbb{R}$. *Then, with*

$$\begin{aligned}
\tilde{S} &= \begin{bmatrix} s_1 & \cdots & s_{j-1} & s_{j+1} & \cdots & s_m \end{bmatrix} \\
\text{and } \tilde{Y} &= \begin{bmatrix} y_1 & \cdots & y_{j-1} & y_{j+1} & \cdots & y_m \end{bmatrix},
\end{aligned}$$

*yields* $\text{BFGS}(W, S, Y) = \text{BFGS}(W, \tilde{S}, \tilde{Y})$ *for any* $W \succ 0$.

## General case

From the compact form of BFGS updates, one should consider:

$$\tilde{Y}_{1:m} = Y_{1:m} + W^{-1} S_{1:m} \begin{bmatrix} A & 0 \end{bmatrix} + y_0 \begin{bmatrix} b \\ 0 \end{bmatrix}^T \qquad (\star)$$

### Theorem 3

*Suppose*

- $W \succ 0$,
- $S_{1:m}$ *has linearly independent columns,*
- $s_0 = S_{1:m}\tau$ *for some* $\tau \in \mathbb{R}^m$.

*Then, there exists* $A \in \mathbb{R}^{m\times(m-1)}$ *and* $b \in \mathbb{R}^{m-1}$ *such that* $(\star)$ *yields*

$$\mathrm{BFGS}(W, S_{0:m}, Y_{0:m}) = \mathrm{BFGS}(W, S_{1:m}, \tilde{Y}_{1:m}).$$

# Computing $A$ and $b$

The compact form involves the matrix:

$$R_{1:m} = \begin{bmatrix} s_1^T y_1 & \cdots & s_1^T y_m \\ & \ddots & \vdots \\ & & s_m^T y_m \end{bmatrix}$$

The key equations that one needs to satisfy to compute $A$ and $b$:

$$\begin{bmatrix} b \\ 0 \end{bmatrix} = -\rho_0 (S_{1:m}^T Y_{1:m} - R_{1:m})^T \tau$$

$$R_{1:m} = \tilde{R}_{1:m}$$

$$(\tilde{Y}_{1:m} - Y_{1:m})^T W (\tilde{Y}_{1:m} - Y_{1:m}) = \left( \frac{1}{\rho_0} + \|y_0\|_W^2 \right) \begin{bmatrix} b \\ 0 \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix}^T - \begin{bmatrix} A & 0 \end{bmatrix}^T (S_{1:m}^T Y_{1:m} - R_{1:m}) - (S_{1:m}^T Y_{1:m} - R_{1:m})^T \begin{bmatrix} A & 0 \end{bmatrix}$$

# Computing $A$ and $b$

The key equations that one needs to satisfy to compute $A$ and $b$:

$$\begin{bmatrix} b \\ 0 \end{bmatrix} = -\rho_0 (S_{1:m}^T Y_{1:m} - R_{1:m})^T \tau$$

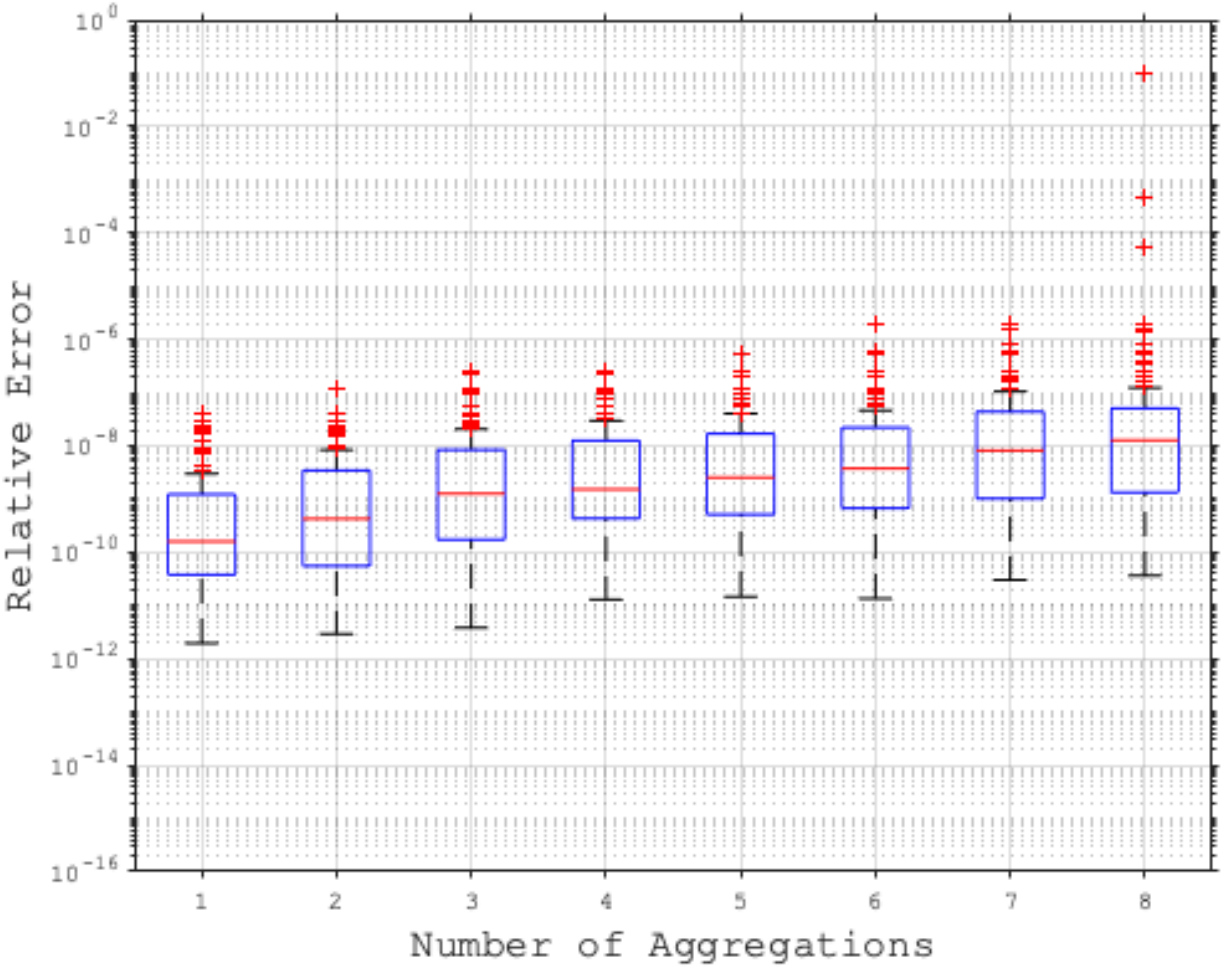
$$R_{1:m} = \tilde{R}_{1:m}$$

$$(\tilde{Y}_{1:m} - Y_{1:m})^T W (\tilde{Y}_{1:m} - Y_{1:m}) = \left( \frac{1}{\rho_0} + \|y_0\|_W^2 \right) \begin{bmatrix} b \\ 0 \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix}^T - \begin{bmatrix} A & 0 \end{bmatrix}^T (S_{1:m}^T Y_{1:m} - R_{1:m}) - (S_{1:m}^T Y_{1:m} - R_{1:m})^T \begin{bmatrix} A & 0 \end{bmatrix}$$

Iterative procedure to compute elements of $A$:

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m-1} \\ a_{2,1} & \ddots & \vdots \\ \vdots & \ddots & a_{m-1,m-1} \\ a_{m,1} & \cdots & a_{m,m-1} \end{bmatrix}$$

# Agg-BFGS, $n = 128$



# Playing devil's advocate

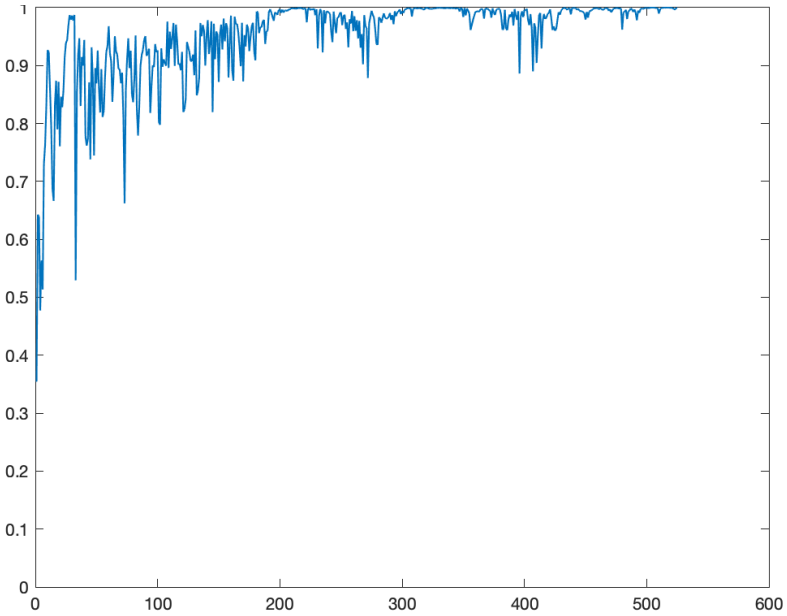
"How much does all of this cost?"

- $\mathcal{O}(m^2 n) + \mathcal{O}(m^4)$
- (LBFGS $= \mathcal{O}(4mn)$)
- Hence, only reasonable for small $m$.
- *More* expensive than BFGS for $m = n$!

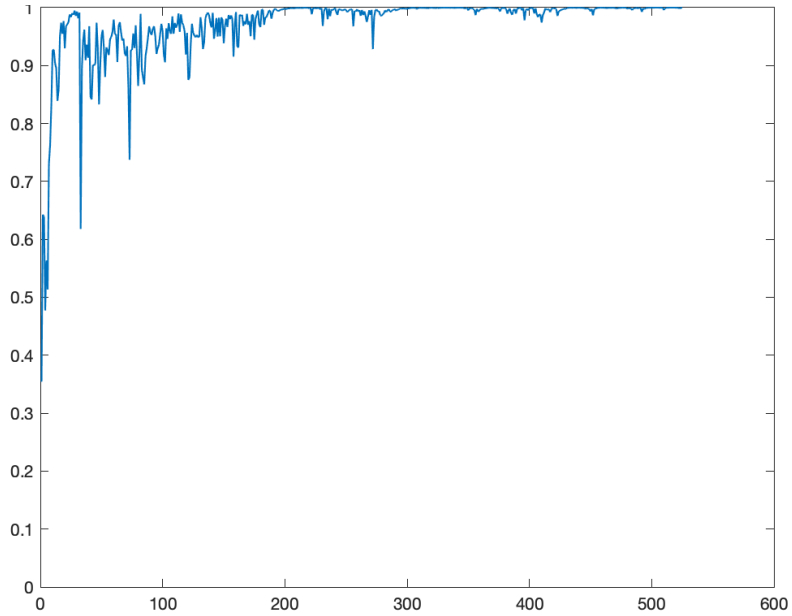
"When does $s_{k-m} = S_{k-m+1:k}\tau$ ever hold?"

- Rarely holds exactly.
- However, one finds it's often close!

`eigenb`, $n = 50$

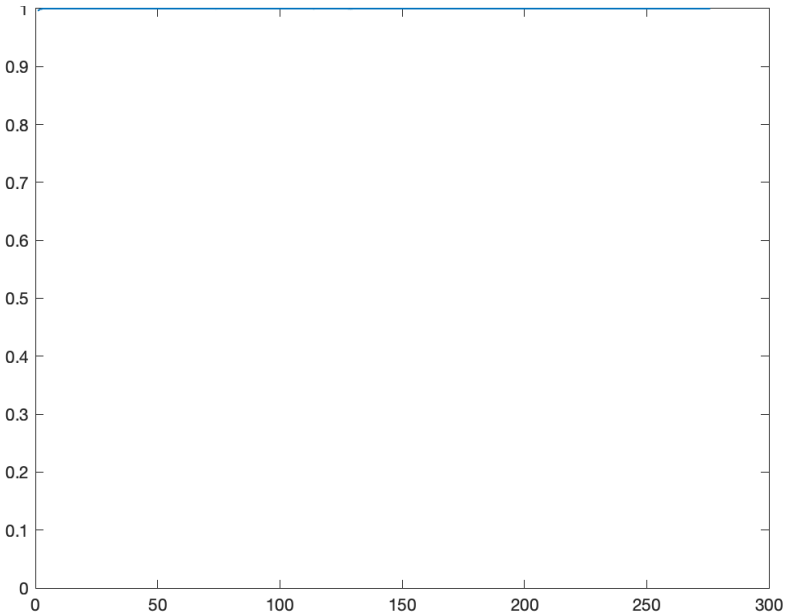



$m = 10$

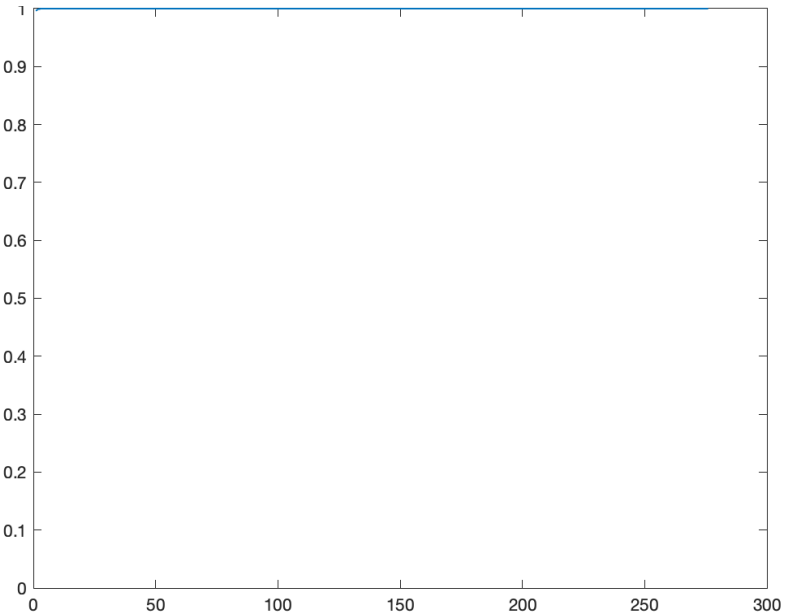



$m = 20$

# chainwoo, $n = 1000$

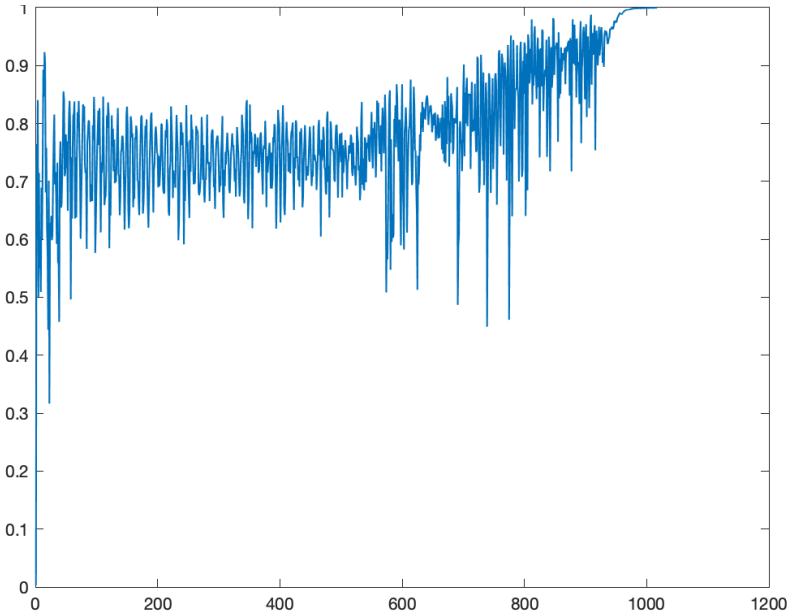



$m = 10$

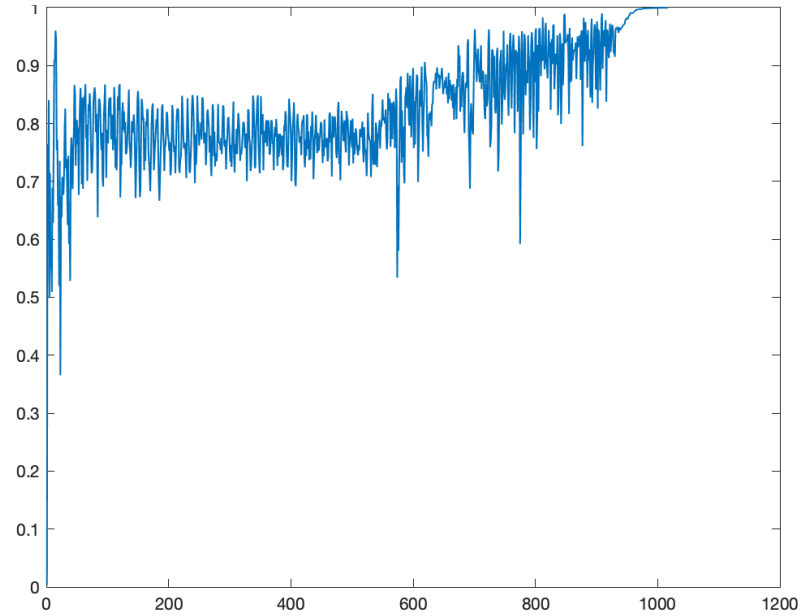



$m = 20$

# broydn7d, $n = 1000$




$m = 10$



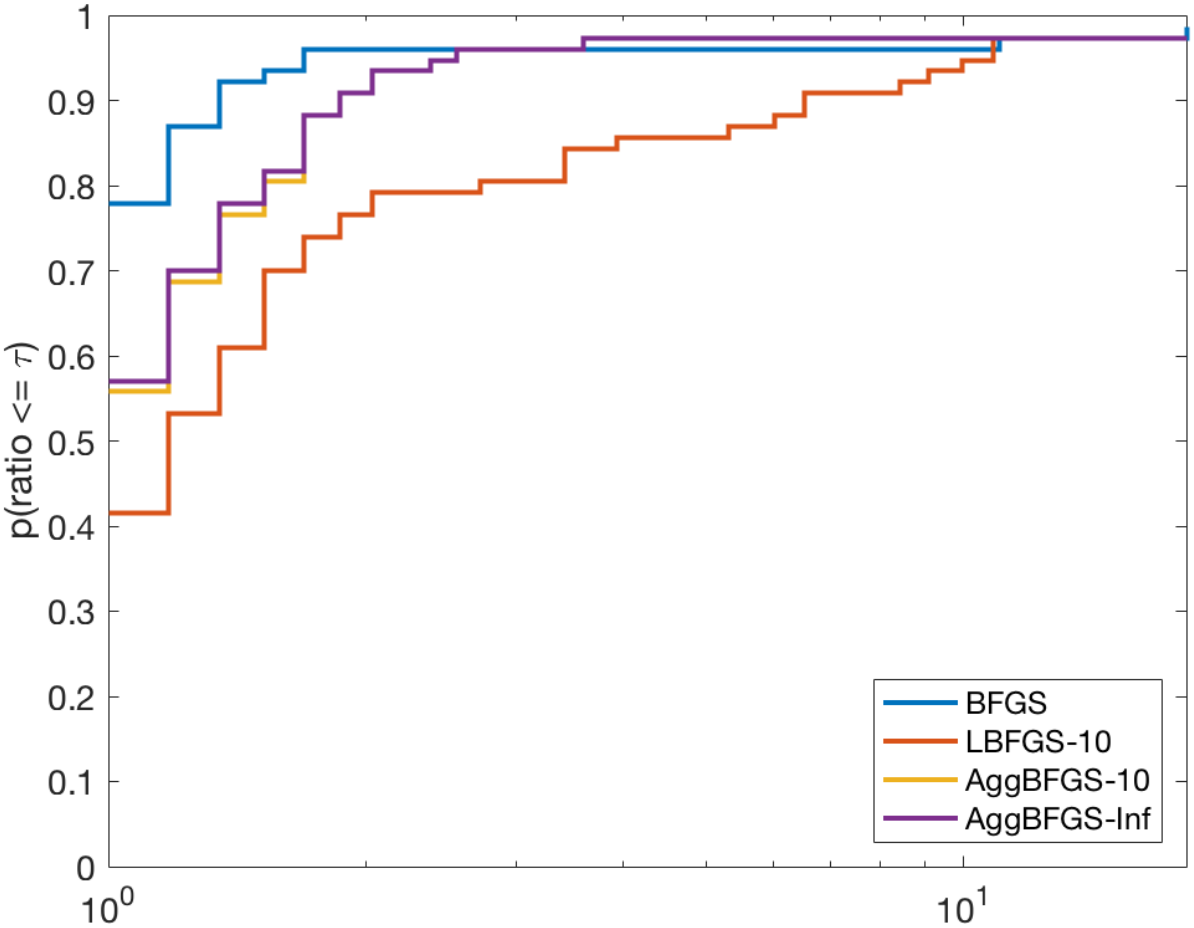

$m = 20$

# Ideas for $m \ll n$

Rotate $s_{k-m}$ to lie in $\text{span}\{s_{k-m+1}, \ldots, s_k\}$.

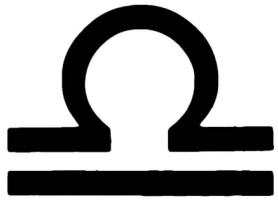
- Apply same rotation to $y_{k-m}$ to ensure $s_{k-m}^T y_{k-m} > 0$(?)
- Use as trigger for increasing history.
- Or use accuracy measure.

# Preliminary results




$n \in \{2, \ldots, 20\}$, $m = 10$

# Final thoughts



Overall, when aggregation is accurate...

- Information perfectly preserved.
- Agg-BFGS($m$) performance can be better than L-BFGS($m$).

Implementation is not trivial.

- Currently implementing preconditioning strategies.
- Plan to release Matlab and C++ `AggQN` objects.

Why AggQN?

- Adaptation to DFP updates is straightforward.
- For SR1 and Broyden class, not so easy.

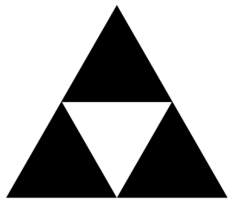
# Outline

# Quasi-Newton methods for nonsmooth optimization

Using quasi-Newton methods in nonsmooth optimization is not new.[†]

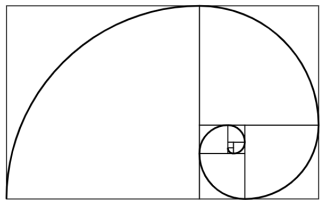
To name only a few...

- Bonnans et al. (1995)
- Hiriart-Urruty & Lemaréchal (1993)
- Kiwiel (2000)
- Lemaréchal (1982)
- Mifflin et al. (1998)
- Vlček & Lukšan (2001)
- Lewis & Overton (2013)



[†] Also other variable metric methods; e.g, Shor's R algorithm.

# What's new?



What could I say that is new?

Previously proposed methods either

- Use a bundle method (or other) as the pillar, with BFGS on top; or
- Use pure BFGS, but with limited theory.

Our goal:

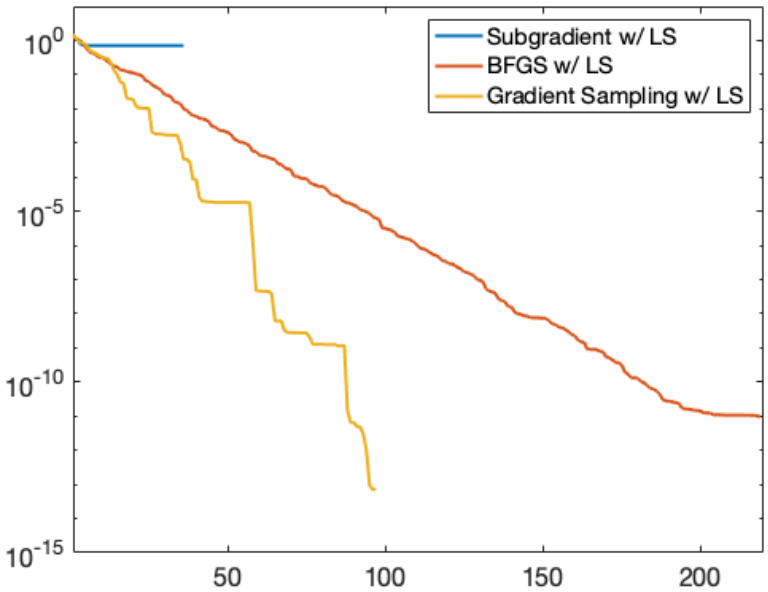
- Use pure BFGS as the pillar, and
- only use damping, cutting planes, gradient sampling, etc. as needed.

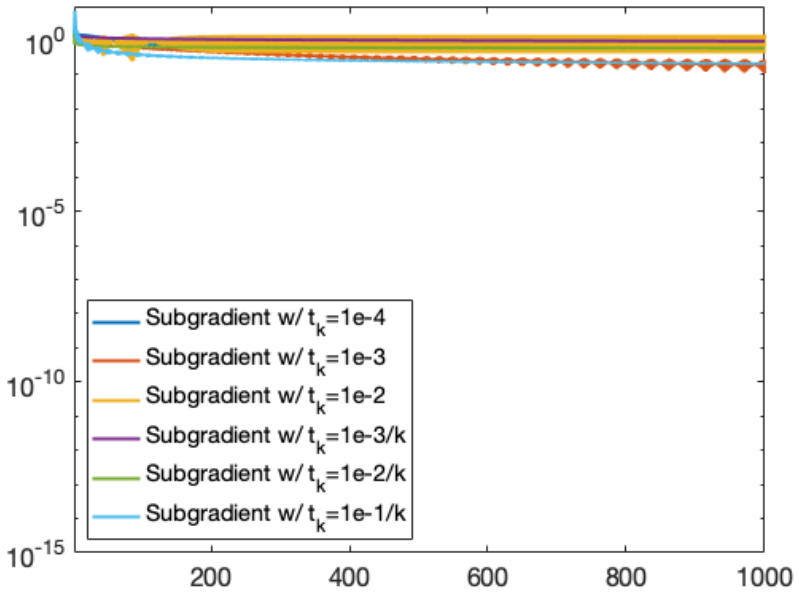
Distinction may seem subtle, but in practice can be significant.

# Nonsmooth Rosenbrock

"But hold on. Why not something simpler?"

$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2$$

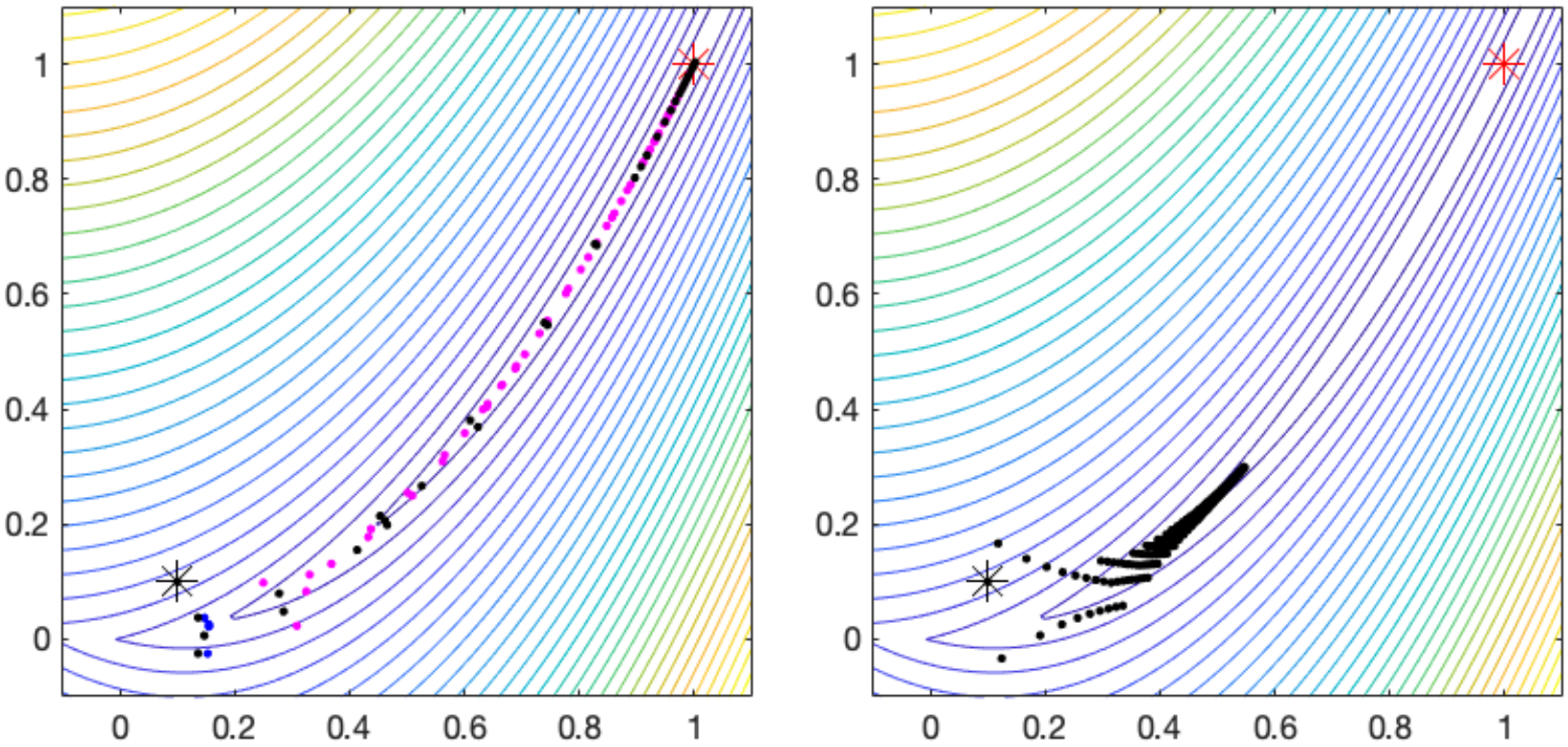


# Nonsmooth Rosenbrock

"But hold on. Why not something simpler?"

$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2$$



# Search direction computation

At $x_k \in \mathbb{R}^n$, a smooth optimization algorithm computes $d_k \leftarrow x_k^* - x_k$, where

$$x_k^* \in \arg\min_{x \in \mathbb{R}^n} \; f(x_k) + \nabla f(x_k)^T (x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k)$$
$$\text{s.t. } \|x - x_k\| \leq \delta_k.$$

For nonsmooth $f$, with sets of points, scalars, and (sub)gradients

$$\{x_{k,j}\}_{j=1}^m, \quad \{f_{k,j}\}_{j=1}^m, \quad \text{and} \quad \{g_{k,j}\}_{j=1}^m,$$

one solves the primal subproblem

$$\min_{x \in \mathbb{R}^n} \left( \max_{j \in \{1,\ldots,m\}} \{f_{k,j} + g_{k,j}^T (x - x_{k,j})\} + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \right) \quad \text{(P)}$$
$$\text{s.t. } \|x - x_k\| \leq \delta_k.$$

- Bundle methods: Lemaréchal, 1980; Wolfe, 1975; Schramm & Zowe, 1992
- Gradient sampling: Burke, Lewis, & Overton, 2005; Kiwiel, 2007.

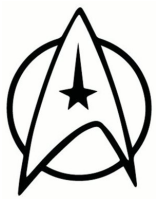
# Dual subproblem

With $G_k \leftarrow [g_{k,1} \ \cdots \ g_{k,m}]$, it is typically more efficient to solve the dual

$$\begin{aligned} \sup_{(\omega,\gamma)\in\mathbb{R}^m_+\times\mathbb{R}^n} \quad & -\tfrac{1}{2}(G_k\omega+\gamma)^T W_k(G_k\omega+\gamma) + b_k^T\omega - \delta_k\|\gamma\|_* \\ \text{s.t.} \quad & \mathbb{1}_m^T\omega = 1. \end{aligned} \tag{D}$$

The primal solution can then be recovered by

$$x_k^* \leftarrow x_k - W_k \underbrace{(G_k\omega_k+\gamma_k)}_{\tilde{g}_k}.$$



If $f_{k,j} \approx f_k$ for all $j$, $W_k = I$, and $\delta_k = \infty$, then (D) computes

$$\tilde{g}_k = G_k\omega_k = \text{minimum norm element in } \operatorname{conv}\{g_{k,1},\ldots,g_{k,m}\}.$$

# Self-correcting properties of BFGS updates with $\{(s_k, v_k)\}$

## Theorem 4 (Byrd & Nocedal, 1989)

*Suppose that, for all $k$, there exists $\{\eta, \theta\} \subset \mathbb{R}_{++}$ such that*

$$\eta \leq \frac{s_k^T v_k}{\|s_k\|_2^2} \quad \textit{and} \quad \frac{\|v_k\|_2^2}{s_k^T v_k} \leq \theta. \tag{KEY}$$

*Then, for any $p \in (0,1)$, there exist constants $\{\iota, \kappa, \lambda\} \subset \mathbb{R}_{++}$ such that, for any $K \geq 2$, the following relations hold for at least $\lceil pK \rceil$ values of $k \in \{1, \ldots, K\}$:*

$$\iota \leq \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \textit{and} \quad \kappa \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \lambda.$$

## Proof, Main Idea.

Show that the sequence of values

$$\{\phi(H_k)\} = \{\mathrm{tr}(H_k) - \ln \det(H_k)\}$$

is bounded above by a function that increases *at most linearly* over $k$.

**Algorithm** Variable-Metric Algorithm Framework

1: Choose $x_1 \in \mathbb{R}^n$.
2: Choose a symmetric positive definite $W_1 \in \mathbb{R}^{n \times n}$.
3: Choose $\alpha \in (0, 1)$
4: **for all** $k \in \mathbb{N} := \{1, 2, \dots\}$ **do**
5: Solve (P)–(D) such that setting

$$G_k \leftarrow \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix},$$
$$s_k \leftarrow -W_k(G_k\omega_k + \gamma_k),$$
$$\text{and} \quad x_{k+1} \leftarrow x_k + s_k$$

6: yields (potentially after a line search)

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\alpha(G_k\omega_k + \gamma_k)^T W_k(G_k\omega_k + \gamma_k).$$

7: Choose $y_k \in \mathbb{R}^n$.
8: Set $\beta_k \leftarrow \min\{\beta \in [0, 1] : v(\beta) := \beta s_k + (1 - \beta)y_k$ satisfies (KEY)$\}$.
9: Set $v_k \leftarrow v(\beta_k)$.
10: Set

$$W_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T W_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}.$$

11: **end for**

# Self-correction to prove convergence

## Corollary 5

*Suppose the conditions of Theorem 4 hold. Then, for any $p \in (0,1)$, there exist constants $\{\mu, \nu\} \subset \mathbb{R}_{++}$ such that, for any $K \geq 2$, the following relations hold for at least $\lceil pK \rceil$ values of $k \in \{1, \ldots, K\}$:*

$$\mu \|g_k\|_2^2 \leq g_k^T W_k g_k \quad \text{and} \quad \|W_k g_k\|_2^2 \leq \nu \|g_k\|_2^2$$

## Theorem 6

*There exists an infinite subset of iteration numbers $\mathcal{K}$ over which*

$$\lim_{k \in \mathcal{K}, k \to \infty} \|G_k \omega_k + \gamma_k\|_2 = 0 \quad \text{and} \quad \lim_{k \in \mathcal{K}, k \to \infty} \|s_k\|_2 = 0.$$

# Last piece of the puzzle

For different algorithm variants, the limits

$$\lim_{k\in\mathcal{K},k\to\infty} \|G_k\omega_k + \gamma_k\|_2 = 0 \quad \text{and} \quad \lim_{k\in\mathcal{K},k\to\infty} \|s_k\|_2 = 0.$$

helps to show that for some $\mathcal{K}'$ that

$$\lim_{k\in\mathcal{K},k\to\infty} \|G_k\omega_k\|_2 = 0.$$

Hence, with gradients in $G_k$ coming from $\mathbb{B}(x_k, \epsilon_k)$, then

$$\epsilon_k \to 0 \quad \Longrightarrow \quad \text{any limit point is (Clarke) stationarity for } f.$$

# Nonsmooth test problems

| Name | Convex? | $f(x_0)$ | $f(x_*)$ |
|---|---|---|---|
| maxq | Yes | 2500.0 | 0.0 |
| mxhilb | Yes | 4.5 | 0.0 |
| chained lq | Yes | 49.0 | -69.3 |
| chained cb3 1 | Yes | 980.0 | 98.0 |
| chained cb3 2 | Yes | 980.0 | 98.0 |
| active faces | No | 3.9 | 0.0 |
| brown function 2 | No | 98.0 | 0.0 |
| chained mifflin 2 | No | 232.8 | -34.8 |
| chained crescent 1 | No | 292.3 | 0.0 |
| chained crescent 2 | No | 292.3 | 0.0 |

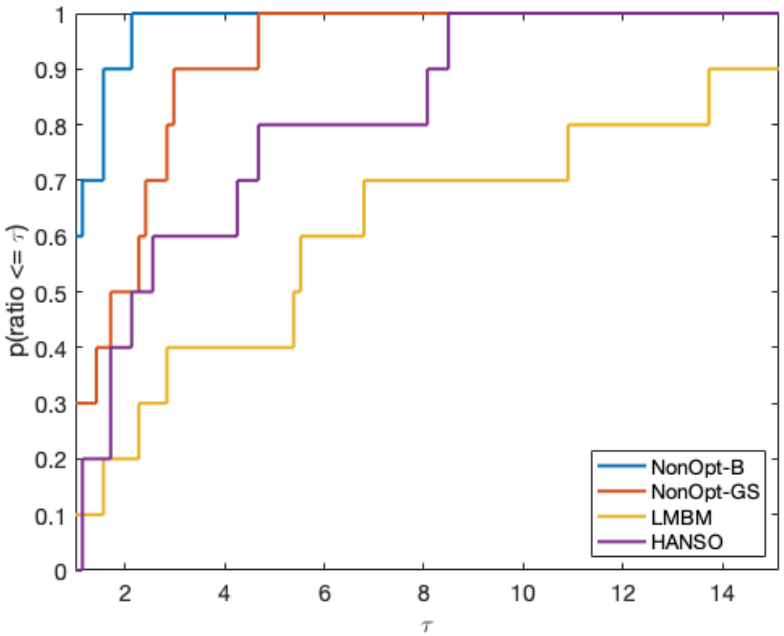
Codes:

- NonOpt (← our code)
- LMBM: Karmitsa/Haarala, Miettinen, & Mäkelä, 2004 & 2007
- HANSO: Lewis & Overton, 2012; Burke, Lewis, & Overton, 2005

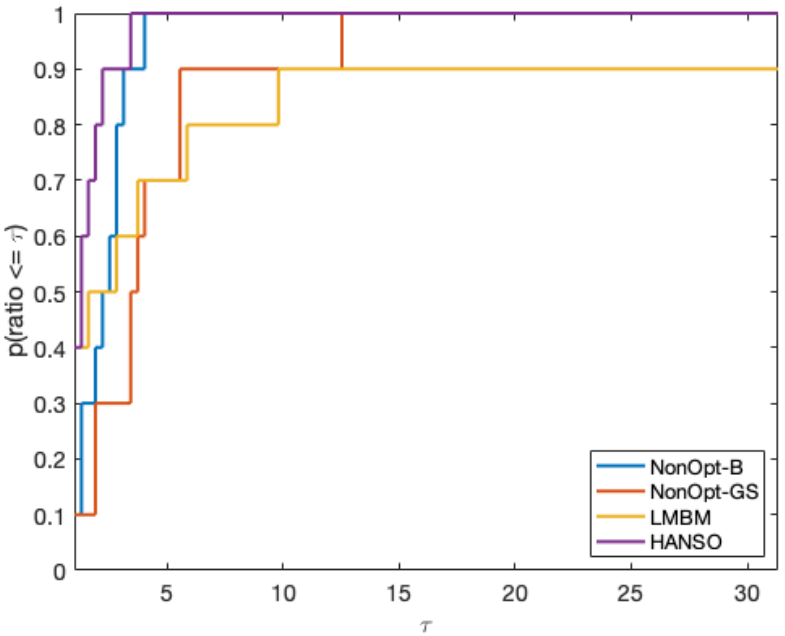
Termination:

- NonOpt terminates for all problems with $\|G_k\omega_k\| \leq 10^{-4}$.
- LMBM terminates 9/10 times due to small objective improvement.
- HANSO terminates 5/10 times due to small objective improvement.

# Numerical results, nonsmooth test problems




Iterations



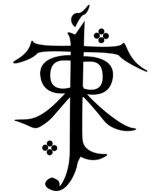

Function evaluations

# Chained Crescent 1

```
+-----------------------------------------------------------+
|            NonOpt = Nonsmooth Optimization Solver         |
| Please visit http://coral.ise.lehigh.edu/frankecurtis/nonopt |
+-----------------------------------------------------------+

Number of variables.............. : 50
Direction computation strategy... : CuttingPlane
Inverse Hessian update strategy.. : BFGS
Line search strategy............. : WeakWolfe
Point set update strategy........ : Proximity
QP solver strategy............... : ActiveSet
Symmetric matrix strategy........ : Dense

------------------------------------------------------------------------------------------------------------------------
  Iter.    Objective    Stat. Rad.   Trust Rad.  |Points|    QP Error    |G. Combo.|     |Step|      Stepsize    Correction
------------------------------------------------------------------------------------------------------------------------
     0  +2.9225e+02  +7.0000e-01  +7.0000e+04         0  +0.0000e+00  +7.0000e+00  +7.0000e+00  +5.0000e-01  +0.0000e+00
     1  +2.8775e+02  +7.0000e-01  +7.0000e+04         1  +0.0000e+00  +7.0000e+00  +1.7567e+00  +1.0000e+00  +0.0000e+00
...
    19  +5.2006e-08  +1.0000e-04  +7.0000e+00        12  +7.7753e-16  +1.7149e-04  +8.5751e-05  +1.0000e+00  +0.0000e+00
    20  +2.2457e-08  +1.0000e-04  +7.0000e+00        14  +2.2204e-16  +8.3220e-05  +4.1619e-05

EXIT: Stationary point found.

Objective........................ : 2.245677e-08

Number of iterations............. : 20
Number of inner iterations....... : 35
Number of QP iterations.......... : 16
Number of function evaluations... : 82
Number of gradient evaluations... : 35

CPU seconds...................... : 0.006477
CPU seconds in NonOpt............ : 0.005924
CPU seconds in evaluations....... : 0.000553
```

# Outline

# References



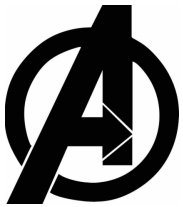
Quasi-Newton ideas...

- aggregating information
- using "pre-superlinear" results for nonsmooth optimization
- (...also works for stochastic optimization!)


⋆ A. Berahas, F. E. Curtis, and B. Zhou.
Limited-Memory BFGS with Displacement Aggregation.
arXiv 1903.03471, 2019.

⋆ F. E. Curtis, D. P. Robinson, and B. Zhou.
A Self-Correcting Variable-Metric Algorithm Framework for Nonsmooth Optimization.
*IMA Journal of Numerical Analysis*, 10.1093/imanum/drz008, 2019.

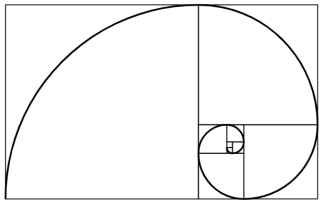
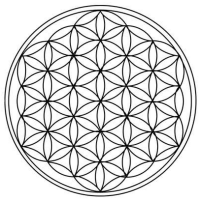
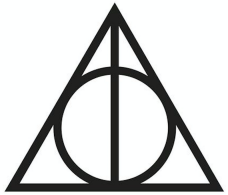
# Symbols

| | |
|---|---|
|  | Ankh, Key of Life |
|  | Avengers |
|  | Batman |
|  | Biohazard |

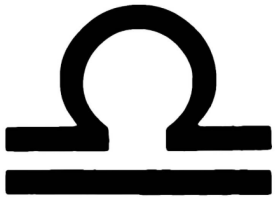
# Symbols (cont.)

| | |
|---|---|
|  | Caduceus, Hermes' staff |
|  | Celtic Trinity Knot / Trefoil knot |
|  | Darwin fish |
|  | Eye of Horus |

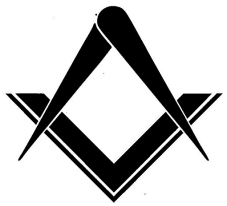
# Symbols (cont.)

| | |
|---|---|
|  | Fibonacci spiral, Golden ratio |
|  | Flower of Life |
|  | Harry Potter, Deathly Hallows |
|  | House Stark, Game of Thrones |

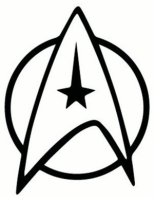
# Symbols (cont.)

| | |
|---|---|
|  | Khanda, Sikh |
|  | Legend of Zelda / Hōjō Clan |
|  | Libra, Zodiac |
|  | Lord of the Rings, Tolkien |

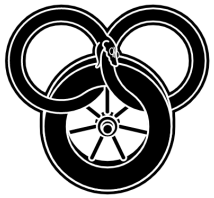
# Symbols (cont.)

| | |
|---|---|
|  | Masonic Square, Freemasons |
|  | Mockingjay, Hunger Games |
|  | Om |
|  | Peace sign |

# Symbols (cont.)

| | |
|---|---|
|  | Rebel Alliance, Star Wars |
|  | Recycling |
|  | Rod of Asclepius |
|  | Starfleet, Star Trek |

# Symbols (cont.)

| | |
|---|---|
|  | Transgender |
|  | Water, Confucianism |
|  | Wheel of Time |
|  | Yin and yang |