# Aggregated bfGs

**Frank E. Curtis**, Lehigh University

joint work with

**Albert S. Berahas**, University of Michigan
**Baoyu Zhou**, Arizona State University

presented at

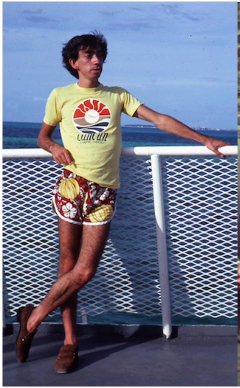
Donald Goldfarb Celebration Workshop

November 8, 2024

# To Don!

BFGS

# To Don!



L- BFGS

# Outline

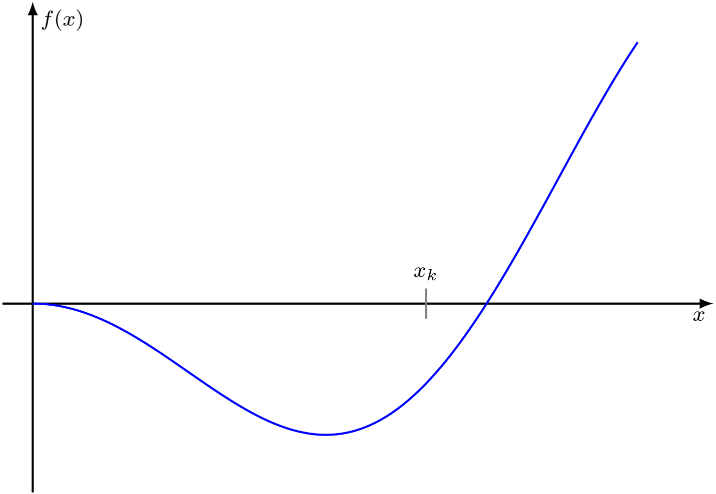
BFGS and L-BFGS

Aggregation

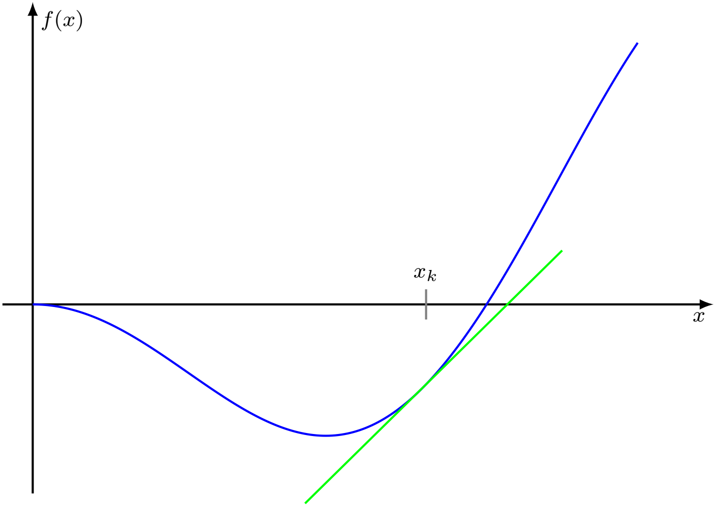
Conclusion

# Outline

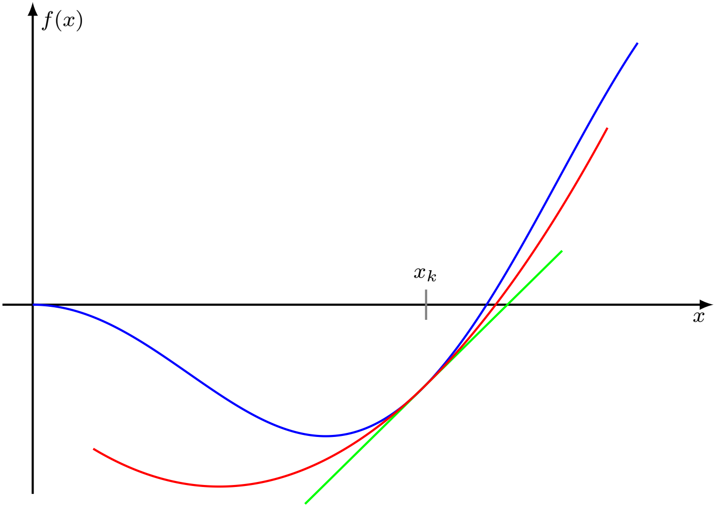
# Quasi-Newton



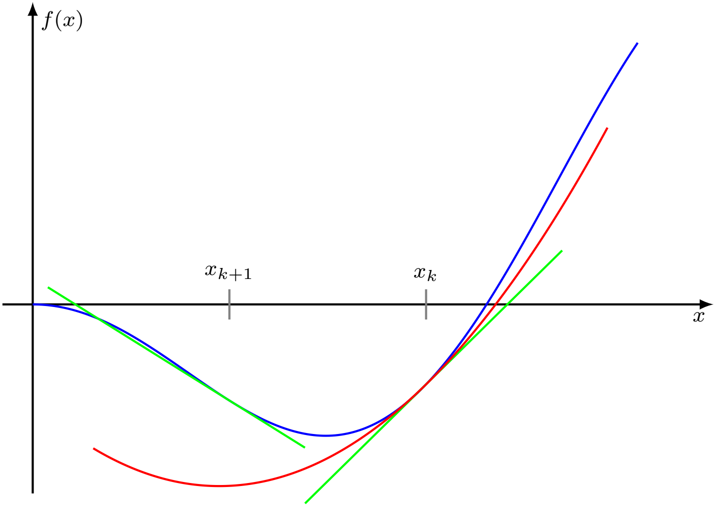
# Quasi-Newton



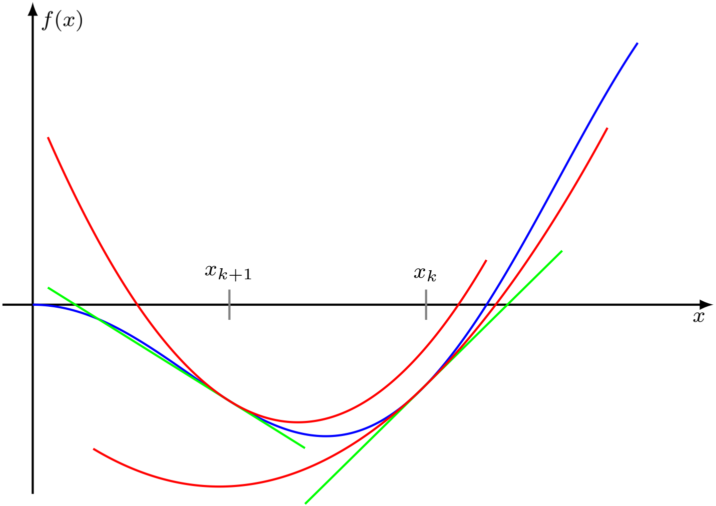
# Quasi-Newton



# Quasi-Newton



# Quasi-Newton



# Notation

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

# The "G" paper

## A Family of Variable-Metric Methods Derived by Variational Means

By Donald Goldfarb

**Abstract.** A new rank-two variable-metric method is derived using Greenstadt's variational approach [*Math. Comp.*, this issue]. Like the Davidon-Fletcher-Powell (DFP) variable-metric method, the new method preserves the positive-definiteness of the approximating matrix. Together with Greenstadt's method, the new method gives rise to a one-parameter family of variable-metric methods that includes the DFP and rank-one methods as special cases. It is equivalent to Broyden's one-parameter family [*Math. Comp.*, v. 21, 1967, pp. 368–381]. Choices for the inverse of the weighting matrix in the variational approach are given that lead to the derivation of the DFP and rank-one methods directly.

# BFGS update

Minimal deviation from $W_k$ subject to secant equation:

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

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

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).

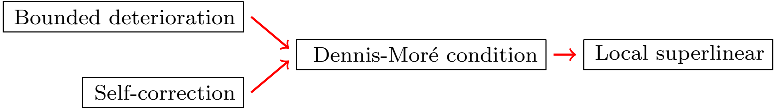
- $P_k$ yields $H_k$-orthogonal projection onto $\operatorname{span}(s_k)$.
- $Q_k$ yields $H_k$-orthogonal projection onto $\operatorname{span}(s_k)^{\perp_{H_k}}$.

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

- 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, e.g., for strongly convex objectives:




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

# Self-Correction

## Theorem 1 (Self-correcting properties of BFGS)

*Suppose* $H_1 \succ 0$ *and for some* $(r_1, r_2)$ *the sequence* $\{(s_k, y_k)\}$ *satisfies*

$$r_1 \leq \frac{s_k^T y_k}{\|s_k\|_2^2} \quad \text{and} \quad \frac{\|y_k\|_2^2}{s_k^T y_k} \leq r_2.$$

*Then, for any* $p \in (0,1)$, *there exist* $(\lambda_1, \lambda_2, \lambda_3) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ *such that, for any* $K \geq 2$, *the following hold for at least* $\lceil pK \rceil$ *values of* $k \in [K]$:

$$\lambda_1 \leq \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \text{and} \quad \lambda_2 \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \lambda_3.$$

Proved by monitoring changes in the generalized distance function

$$\psi(H) = \operatorname{tr}(H) + \log(\det(H)),$$

which corresponding to the negative log-determinant distance generating function.

# L-BFGS

The algorithm generates $\{(s_k, y_k)\}$, and BFGS generates $\{W_k\}$, where for all $k \in \mathbb{N}$ one sets

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

In iteration $k \in \mathbb{N}$, L-BFGS uses only $\{(s_j, y_j)\}_{j=k-m}^{k}$, and "applies" the update $m$ times.

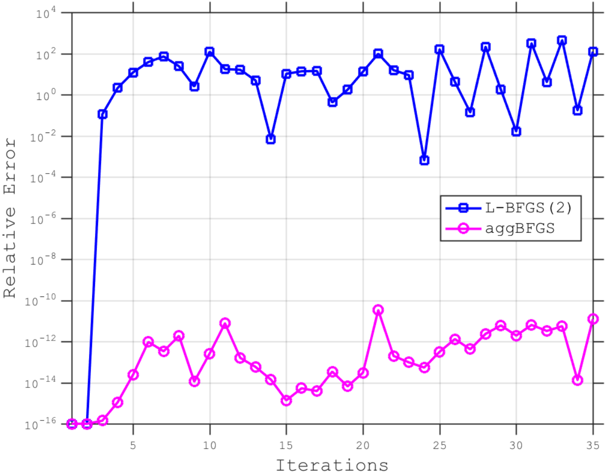
- Notably, **the superlinear convergences guarantees of BFGS are lost. . .**

# Outline

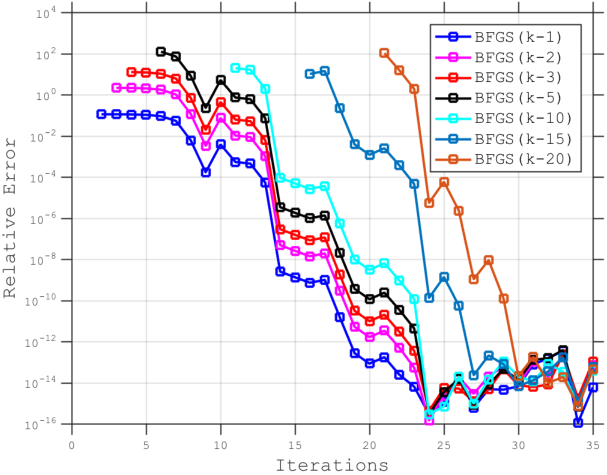
# Motivating questions

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

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

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

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

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

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

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

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

# 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

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

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

## General case

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

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

### 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 $b$:

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

# Computing $A$ and $b$

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

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

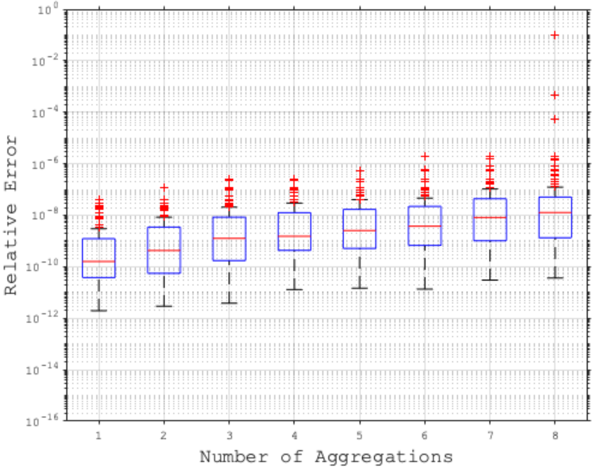
$$R_{1:m} = \tilde{R}_{1:m}$$

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

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

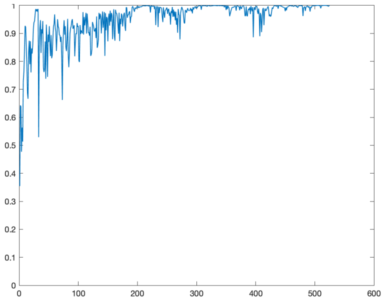
"How much does all of this cost?"

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

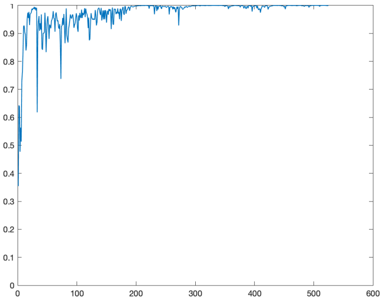
"When does $s_{k-m} = S_{k-m+1:k}\tau$ ever hold?"

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

# `eigenb`, $n = 50$

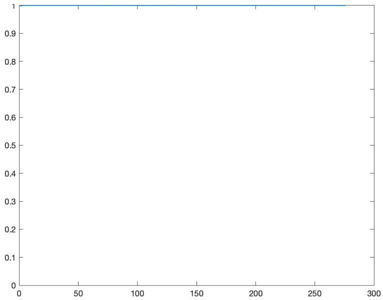



$m = 10$

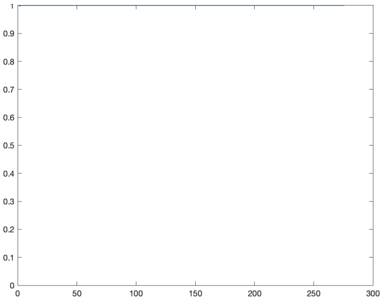



$m = 20$

# chainwoo, $n = 1000$

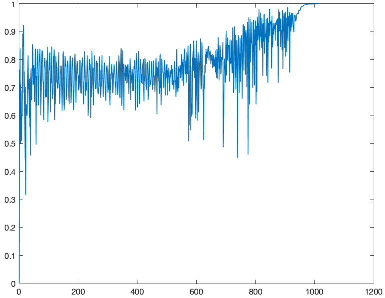



$m = 10$

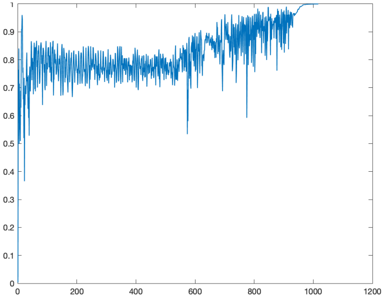



$m = 20$

# `broydn7d`, $n = 1000$




$m = 10$



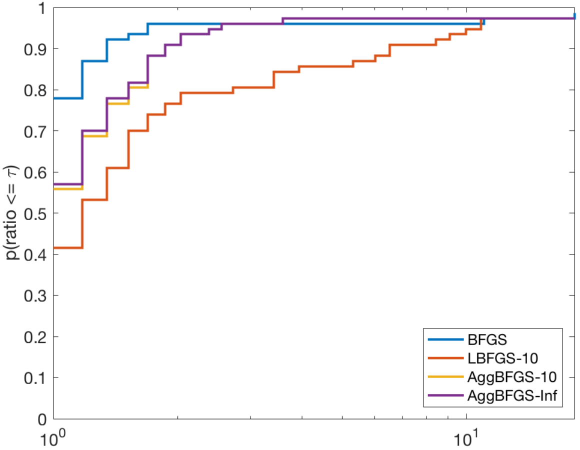

$m = 20$

# Ideas for $m \ll n$

Rotate $s_{k-m}$ to lie in 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




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

# Outline

BFGS and L-BFGS

Aggregation

Conclusion

# Summary

Closing the gap between BFGS and L-BFGS through displacement aggregation.

- If $m = n$, information perfectly preserved $\implies$ L-BFGS can be superlinear!
- If $m < n$, Agg-BFGS($m$) performance can still be better than L-BFGS($m$).

Mathematical Programming
https://doi.org/10.1007/s10107-021-01621-6

FULL LENGTH PAPER

Series A

**Limited-memory BFGS with displacement aggregation**

**Albert S. Berahas[1] · Frank E. Curtis[1] · Baoyu Zhou[1]**

Received: 13 January 2020 / Accepted: 4 January 2021