

Handling Nonpositive Curvature in a Limited Memory Steepest Descent Method

Frank E. Curtis, Lehigh University

joint work with

Wei Guo, Lehigh University

MOPTA Conference — Bethlehem, PA, USA

21 July 2015



Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

Summary

Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

Summary

Context

Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with **large n** , consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

We are interested in stationary points “downhill” from an initial iterate x_0

Context

Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with **large n** , consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

We are interested in stationary points “downhill” from an initial iterate x_0

Supposing f is continuously differentiable and we compute gradients...

Convex f

- ▶ Global/local convergence (rate) guarantees
- ▶ **Efficient methods**

Nonconvex f

- ▶ Global/local convergence (rate) guarantees
- ▶ **Efficient methods???**

Newton's method with line searches

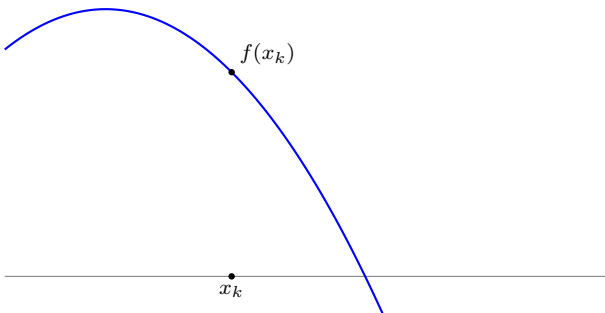
How is nonpositive curvature handled in a line-search-Newton method?

$$x_{k+1} = x_k + \alpha_k s_k, \quad \text{where } s_k = -H_k^{-1} \nabla f(x_k)$$

Newton's method with line searches

How is nonpositive curvature handled in a line-search-Newton method?

$$x_{k+1} = x_k + \alpha_k s_k, \quad \text{where } s_k = -H_k^{-1} \nabla f(x_k)$$

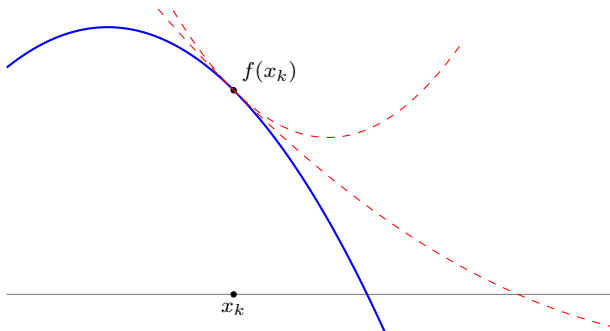


Explicit/Implicit Hessian modification

Newton's method with line searches

How is nonpositive curvature handled in a line-search-Newton method?

$$x_{k+1} = x_k + \alpha_k s_k, \quad \text{where } s_k = -H_k^{-1} \nabla f(x_k)$$



Explicit/Implicit Hessian modification (heuristic, expensive)

Newton's method with trust regions

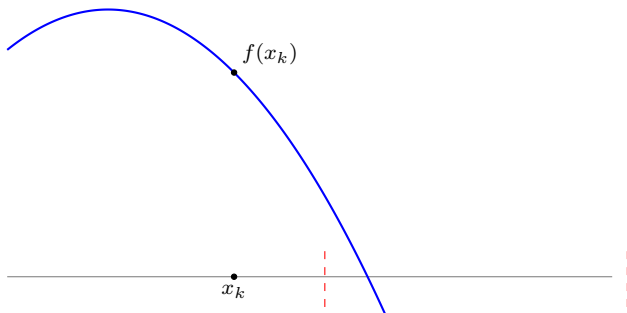
How is nonpositive curvature handled in a trust-region-Newton method?

$$x_{k+1} = x_k + s_k, \quad \text{where } s_k = \arg \min_{\|s\|_2 \leq \delta_k} f_k + \nabla f(x_k)^T s + \frac{1}{2} s^T H_k s$$

Newton's method with trust regions

How is nonpositive curvature handled in a trust-region-Newton method?

$$x_{k+1} = x_k + s_k, \quad \text{where } s_k = \arg \min_{\|s\|_2 \leq \delta_k} f_k + \nabla f(x_k)^T s + \frac{1}{2} s^T H_k s$$

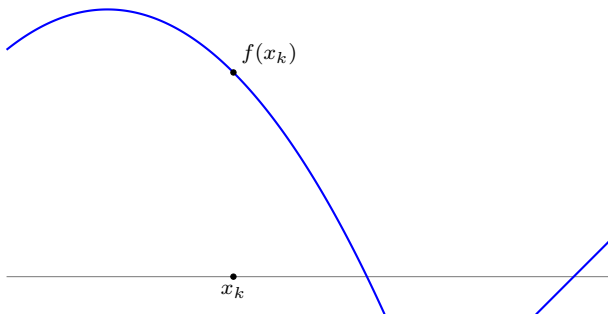


Trust region updates (heuristic, expensive/inexact w/ Steihaug-Toint, GLTR)

Quasi-Newton methods

How is nonpositive curvature handled in a quasi-Newton method?

$$x_{k+1} = x_k + s_k, \quad \text{where } s_k = -H_k^{-1} \nabla f(x_k)$$



Trust regions (SR1, BFGS) or line searches (BFGS w/ Wolfe conditions!)

Steepest descent

We are interested in steepest descent methods:

Algorithm 1 Steepest Descent Framework

- 1: Input $x_0 \in \mathbb{R}^n$.
 - 2: **for** $k \in \mathbb{N} := \{0, 1, 2, \dots\}$ **do**
 - 3: Compute $g_k \leftarrow \nabla f(x_k)$.
 - 4: **Choose** $\alpha_k \in (0, \infty)$.
 - 5: Set $x_{k+1} \leftarrow x_k - \alpha_k g_k$.
 - 6: **end for**
-

All that remains to be determined are the stepsizes $\{\alpha_k\}$

Why steepest descent?

Is it because of widespread interest in “optimal” steepest descent methods?

- ▶ **No.** We are not interested in algorithm complexity analyses (yet).

Is it because we believe they can outperform quasi-Newton methods?

- ▶ **Not for convex problems.** For those, we only hope to be competitive.

Then why?

- ▶ When function/gradient evaluations are relatively cheap, then it may be beneficial to “move quickly” as opposed to “sitting” and computing a step.
- ▶ Handling nonpositive curvature continues to be a pervasive difficulty. There may be new approaches for handling it with only gradient information.
- ▶ Steepest descent methods can be easier to extend for solving constrained problems, e.g., via projected gradient methods.

Exploiting previously computed information

For a given $k \in \mathbb{N}_+ := \{1, 2, 3, \dots\}$, our strategy for computing α_k involves

$$\{x_k, x_{k-1}, \dots, x_{k-m}\} \quad \text{and} \quad \{g_k, g_{k-1}, \dots, g_{k-m}\}$$

Barzilai and Borwein (1988):

- ▶ $m = 1$
- ▶ “two-point step size gradient method”

Fletcher (2012):

- ▶ $m \geq 1$
- ▶ “limited memory steepest descent method”

In both cases:

- ▶ Ideas based on minimizing strictly convex quadratics
- ▶ Ideas generalize when minimizing other convex functions
- ▶ **However, unclear how to handle nonpositive curvature**

Preview of contributions

- ▶ New strategies for handling nonpositive curvature in steepest descent
- ▶ Consider both BB-type and LMSD methods (former is special case of latter)
- ▶ Ideas based on employing cubic models when nonpositive curvature is present
- ▶ Globalization is straightforward with nonmonotone line search
- ▶ Maintain local convergence properties near strict local minimizers
- ▶ Numerical experiments are promising

Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

Summary

Main idea

For a given $k \in \mathbb{N}$, how should we choose α_k ?

- ▶ Define the displacements

$$s_k := x_k - x_{k-1} \quad \text{and} \quad y_k := g_k - g_{k-1}$$

(Recall the classical secant equation $H_k s_k = y_k$)

- ▶ Consider the one-dimensional least-squares problems

$$\min_{\bar{q} \in \mathbb{R}} \frac{1}{2} \|(\bar{q}I) s_k - y_k\|_2^2 \quad \text{and} \quad \min_{\hat{q} \in \mathbb{R}} \frac{1}{2} \|s_k - (\hat{q}^{-1}I) y_k\|_2^2$$

which have the unique solutions¹

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} \quad \text{and} \quad \hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k}$$

- ▶ Both $\bar{q}I$ and $\hat{q}I$ represent simple approximations of the Hessian $\nabla^2 f(x_k)$

¹For simplicity, assume here that $s_k \neq 0$, $y_k \neq 0$, and $s_k^T y_k \neq 0$

Main idea (continued)

- ▶ Consider minimizing, along $s = -\alpha g_k$, the quadratic model given by

$$f(x_k) + g_k^T s + \frac{1}{2} s^T (q_k I) s \approx f(x_k + s)$$

- ▶ For $q_k = \bar{q}_k$ and $q_k = \hat{q}_k$, respectively, we obtain the stepsizes

$$\bar{\alpha}_k := \frac{1}{\bar{q}_k} = \frac{s_k^T s_k}{s_k^T y_k} \quad \text{and} \quad \hat{\alpha}_k := \frac{1}{\hat{q}_k} = \frac{s_k^T y_k}{y_k^T y_k}$$

- ▶ These represent the two BB stepsize alternatives

Theoretical analyses and extensions

For strictly convex quadratics, R-linearly convergent / R-superlinear local rate:

- ▶ Barzilai and Borwein (1988)
- ▶ Raydon (1993)
- ▶ Dai and Liao (2002)

Algorithm extensions:

- ▶ Raydon (1997)
- ▶ Dai, Yuan, and Yuan (2002)
- ▶ Yuan (2006)
- ▶ De Asmundis, Serafino, Riccio, and Toraldo (2013)
- ▶ Xiao, Wang, and Wang (2010)
- ▶ Biglari and Solimanpur (2013)
- ▶ Kafaki and Fatemi (2013)

All except the last essentially ignore issues related to nonpositive curvature. The typical strategy, when $s_k^T y_k < 0$, is to set α_k to a predetermined constant.

A closer look

Letting θ_k be the angle between s_k and y_k , we have

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} = \cos \theta_k \frac{\|y_k\|_2}{\|s_k\|_2}$$

and $\hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k} = \frac{1}{\cos \theta_k} \frac{\|y_k\|_2}{\|s_k\|_2}.$

Letting $y_k = u_k + v_k$ where u_k is the projection of y_k onto $\text{span}(s_k)$, we have

$$\bar{q}_k := \frac{s_k^T y_k}{s_k^T s_k} = \frac{s_k^T u_k}{s_k^T s_k}$$

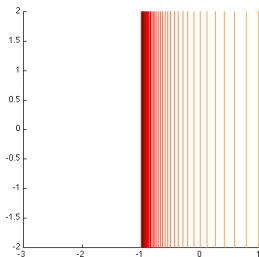
and $\hat{q}_k := \frac{y_k^T y_k}{s_k^T y_k} = \frac{u_k^T u_k + v_k^T v_k}{s_k^T u_k} = \bar{q}_k + \frac{v_k^T v_k}{s_k^T u_k}.$

Key observations:

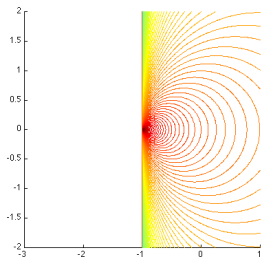
- ▶ $|\bar{q}_k| \leq |\hat{q}_k|$, which implies $|\bar{\alpha}_k| \geq |\hat{\alpha}_k|$
- ▶ The “bar” quantities only observe displacement of gradient along s_k , whereas the “hat” quantities observe the entire gradient displacement

Visualizing the BB stepsizes

Figure: Suppose $g_{k-1} = (-1, 0)$ and $\alpha_{k-1} = 1$ so that $s_k = -\alpha_{k-1}g_{k-1} = (1, 0)$. The contours illustrate the stepsize α_k as a function of the gradient g_k .



(a) $q_k = \bar{q}_k$



(b) $q_k = \hat{q}_k$

Key observations:

- ▶ Extremely different stepsizes when $s_k^T y_k > 0$ and vectors are \sim orthogonal
- ▶ No contours in left-hand sides since $s_k^T y_k < 0$ leads to constant stepsize!

Basics of our strategy

Compute \hat{q}_k

- ▶ If $\hat{q}_k > 0$, then set $\alpha_k \leftarrow 1/\hat{q}_k$
- ▶ If $\hat{q}_k < 0$, then consider the cubic model

$$m_k(s) = f(x_k) + g_k^T s + \frac{1}{2} \hat{q}_k \|s\|_2^2 + \frac{1}{6} c_k \|s\|_2^3 \approx f(x_k + s)$$

Choose $c_k > 0$ so that minimizing m_k along $s = -\alpha g_k$ yields a good stepsize

- ▶ If $\hat{q}_k = 0$, then handle as a special case (see later)

Choosing the cubic term coefficient

Idea #1:

- ▶ Choosing c_k to minimize the least-squares error

$$\|\nabla m_k(-s_k) - g_{k-1}\|_2^2$$

- ▶ ... leads to the choice

$$c_k \leftarrow \frac{1}{2\|s_k\|_2}(\bar{q}_k - \hat{q}_k)$$

Idea #2:

- ▶ Choosing c_k so the curvature of m_k at $-s_k$ along s_k is equal to \bar{q}_k , i.e.,

$$s_k^T \nabla^2 m_k(-s_k) s_k = \bar{q}_k \|s_k\|_2^2,$$

- ▶ ... leads to the choice

$$c_k \leftarrow \frac{1}{\|s_k\|_2}(\bar{q}_k - \hat{q}_k)$$

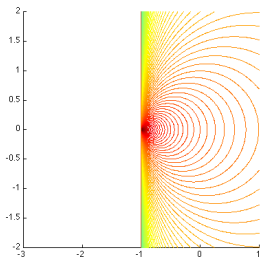
Key observations:

- ▶ Both suggest a similar strategy! (Coefficients only differ by a constant.)
- ▶ If $s_k^T y_k < 0$ and $s_k \nparallel y_k$, then $\bar{q}_k > \hat{q}_k$, so $c_k > 0$.

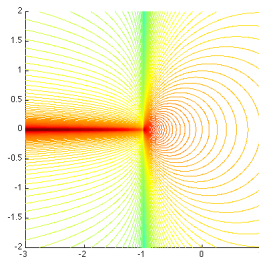
Overall, we take the difference of two second-order directional derivative estimates in order to approximate a third-order directional derivative

Visualizing our stepsizes

Figure: Suppose $g_{k-1} = (-1, 0)$ and $\alpha_{k-1} = 1$ so that $s_k = -\alpha_{k-1}g_{k-1} = (1, 0)$. The contours illustrate the stepsize α_k as a function of the gradient g_k .



(a) $q_k = \hat{q}_k$, constant stepsize for $q_k < 0$



(b) $q_k = \hat{q}_k$, $c_k \geq 0$

Complete algorithm

Algorithm 2 BB-type Method with Cubic Regularization

```

1: Choose  $(\omega, \Omega) \in \mathbb{R} \times \mathbb{R}$  satisfying  $0 < \omega \leq \Omega$  and  $c \in \mathbb{R}_+ := \{c \in \mathbb{R} : c > 0\}$ .
2: Choose  $x_0 \in \mathbb{R}^n$  and set  $f_0 \leftarrow f(x_0)$ .
3: Choose  $\alpha_0 \in [\omega, \Omega]$ .
4: Set  $g_0 \leftarrow \nabla f(x_0)$ .
5: if  $g_0 = 0$  then return the stationary point  $x_0$ . end if
6: Set  $x_1 \leftarrow x_0 - \alpha_0 g_0$  and  $f_1 \leftarrow f(x_1)$ .
7: Set  $k \leftarrow 1$ .
8: loop
9:   Set  $g_k \leftarrow \nabla f(x_k)$ ,  $s_k \leftarrow x_k - x_{k-1}$ , and  $y_k \leftarrow g_k - g_{k-1}$ .
10:  if  $g_k = 0$  then return the stationary point  $x_k$ . end if
11:  if  $y_k = 0$  or  $s_k^T y_k = -\|s_k\|_2 \|y_k\|_2 < 0$  then
12:    Set  $\alpha_k \leftarrow \Omega$ .
13:  else if  $s_k^T y_k = 0$  then
14:    Set  $\alpha_k \leftarrow \omega$ .
15:  else
16:    Set  $q_k \leftarrow y_k^T y_k / s_k^T y_k$ .
17:    if  $q_k > 0$  then Set  $c_k \leftarrow 0$ . else Set  $c_k \leftarrow \frac{c}{\|s_k\|_2} \left( \frac{s_k^T y_k}{s_k^T s_k} - q_k \right)$ . end if
18:    if  $q_k > 0$  then Set  $\alpha_k \leftarrow 1/q_k$ . else Set  $\alpha_k \leftarrow \frac{-q_k + \sqrt{q_k^2 + 2c_k \|g_k\|_2}}{c_k \|g_k\|_2}$ . end if
19:    Replace  $\alpha_k$  by its projection onto the interval  $[\omega, \Omega]$ .
20:  end if
21:  Set  $x_{k+1} \leftarrow x_k - \alpha_k g_k$  and  $f_{k+1} \leftarrow f(x_{k+1})$ .
22:  Set  $k \leftarrow k + 1$ .
23: end loop

```

Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

Summary

Key result

Consider the minimization of $\frac{1}{2}x^T Ax$ for $A \succ 0$.

Theorem 1 (Finite termination of steepest descent)

Suppose that A has n distinct eigenvalues

$$0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n.$$

If $\alpha_{i-1} \leftarrow \lambda_i^{-1}$ for all $i \in \{1, \dots, n\}$, then $g_n = 0$.

Fletcher's main idea

Obtain stepsizes by approximating reciprocals of the eigenvalues of A .

- ▶ At x_k and with $m \geq 1$, consider the matrix of previous gradients

$$G_k := [g_{k-m} \quad \cdots \quad g_{k-1}].$$

- ▶ For all $j \in \{1, \dots, m\}$, we have the following property:

$$x_k - x_{k-j} \in \text{span}\{g_{k-j}, Ag_{k-j}, A^2g_{k-j}, \dots, A^{j-1}g_{k-j}\}.$$

- ▶ This Krylov sequence provides m distinct eigenvalue estimates (Ritz values).
- ▶ In particular, with the QR-decomposition $G_k = Q_k R_k$, the Ritz values are

eigenvalues of T_k , where $T_k = Q_k^T A Q_k$ is tridiagonal.

Computational efficiency

In fact, T_k can be obtained without access to A .

- ▶ Computing the partially extended Cholesky factorization

$$G_k^T \begin{bmatrix} G_k & g_k \end{bmatrix} = R_k^T \begin{bmatrix} R_k & r_k \end{bmatrix},$$

we have

$$T_k = \begin{bmatrix} R_k & r_k \end{bmatrix} J_k R_k^{-1},$$

where J_k is a matrix with only $2m$ nonzeros depending on previous stepsizes.

- ▶ With $m = 1$, we obtain the first BB alternative! That is, $T_k = \bar{q}_k$.

The second BB alternative can be obtained by computing harmonic Ritz values as

$$\text{eigenvalues of } (P_k^{-1} T_k)^{-1}$$

with the pentadiagonal matrix

$$P_k = R_k^{-T} J_k^T \begin{bmatrix} R_k & r_k \\ 0 & \rho_k \end{bmatrix}^T \begin{bmatrix} R_k & r_k \\ 0 & \rho_k \end{bmatrix} J_k R_k^{-1}.$$

Fletcher's LMSD method

Main components:

- ▶ Construct G_k and compute factorization of $G_k^T [G_k \quad g_k] \in \mathbb{R}^{m \times (m+1)}$.
- ▶ Construct $T_k \in \mathbb{R}^{m \times m}$ and compute its eigenvalues.
- ▶ Employ reciprocals of eigenvalues as stepsizes in next m iterations.

Issues for nonquadratics:

- ▶ T_k is not tridiagonal (but is upper Hessenberg).
- ▶ Eigenvalues are not necessarily real.
- ▶ Real eigenvalues are not necessarily positive.
- ▶ Globalization?

Our approach

We essentially employ the following procedure:

- ▶ Follow Fletcher's strategy of computing \tilde{T}_k by **symmetrizing** T_k
- ▶ Compute a pentadiagonal matrix \tilde{P}_k that approximates P_k
- ▶ Compute Ritz and harmonic Ritz values (ordered largest to smallest):

$$\{\bar{q}_k, \bar{q}_{k+1}, \dots, \bar{q}_{k+m-1}\} \text{ as eigenvalues of } \tilde{T}_k$$

and $\{\hat{q}_k, \hat{q}_{k+1}, \dots, \hat{q}_{k+m-1}\}$ as eigenvalues of $(\tilde{P}_k^{-1}\tilde{T}_k)^{-1}$

Theorem 2

If \tilde{T}_k and \tilde{P}_k are well-defined and invertible, then the eigenvalues are interlaced:

$$|\bar{q}_{k+j}| \leq |\hat{q}_{k+j}| \quad \text{for all } j \in \{0, \dots, m-1\}$$

- ▶ Recall $|\bar{q}_k| \leq |\hat{q}_k|$ in $m = 1$ case!
- ▶ For iterations $k, k+1, \dots, k+m-1$, take a pair of eigenvalues and apply a similar approach as in the $m = 1$ case
- ▶ **Reduces to Fletcher's approach if f is convex quadratic**

Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

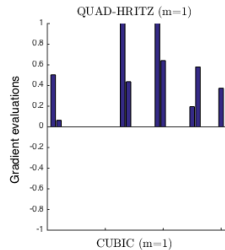
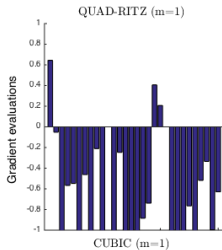
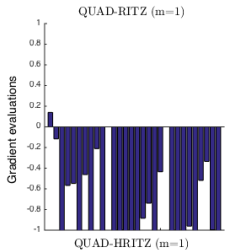
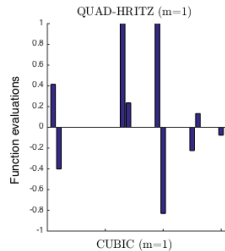
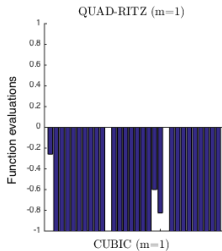
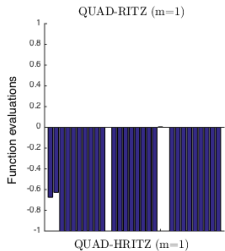
Summary

Implementation

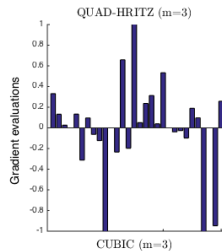
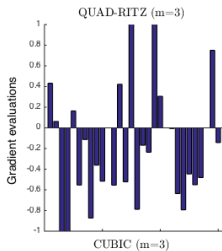
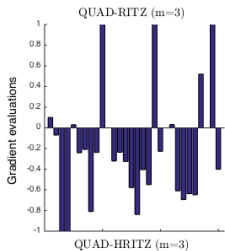
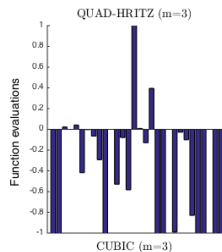
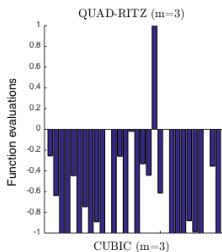
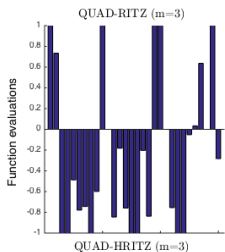
- ▶ Matlab implementation of our approach versus...
- ▶ Matlab implementation of two variants of Fletcher's method:
 - ▶ Ritz or harmonic Ritz values
 - ▶ For $m = 1$, reduces to a BB-type method
 - ▶ For handling nonpositive curvature, set stepsize to large constant
- ▶ Zhang and Hager nonmonotone line search (for all) to ensure convergence
- ▶ Test for $m \in \{1, 3, 5\}$
- ▶ Ran all unconstrained CUTEst problems with $n \geq 50$ that required at least 5 seconds and for which nonpositive curvature was encountered
- ▶ Successful if/when

$$\|g_k\|_\infty \leq 10^{-8} \max\{\|g_0\|_\infty, 1\}$$

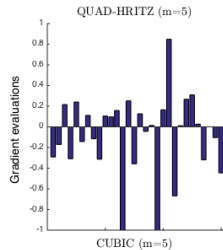
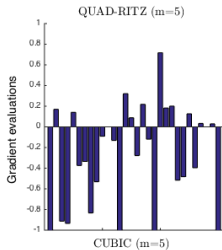
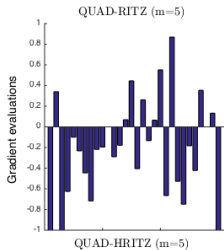
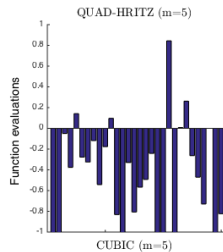
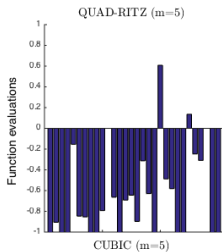
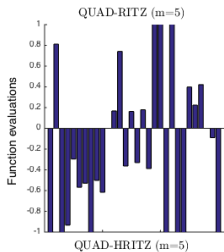
Outperforming factors: $m = 1$



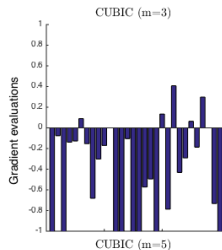
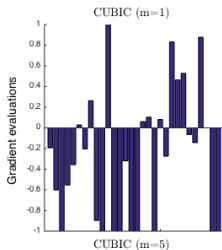
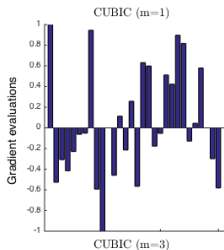
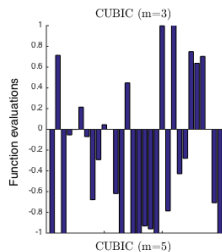
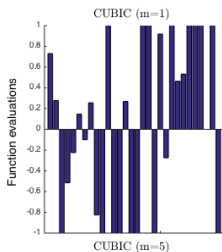
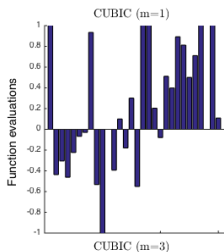
Outperforming factors: $m = 3$



Outperforming factors: $m = 5$



Outperforming factors: $m \in \{1, 3, 5\}$



Outline

Motivation

Barzilai-Borwein-type (BB-type) Method

Limited Memory Steepest Descent (LMSD) Method

Numerical Experiments

Summary

Contributions

- ▶ New strategies for handling nonpositive curvature in steepest descent
 - ▶ Consider both BB-type and LMSD methods (former is special case of latter)
 - ▶ Ideas based on employing cubic models when nonpositive curvature is present
 - ▶ Globalization is straightforward with nonmonotone line search
 - ▶ Maintain local convergence properties near strict local minimizers
 - ▶ Numerical experiments are promising
- ★ F. E. Curtis and W. Guo.
Handling Nonpositive Curvature in a Limited Memory Steepest Descent Method.
IMA Journal of Numerical Analysis, DOI: [10.1093/imanum/drv034](https://doi.org/10.1093/imanum/drv034), 2015.