Step Size: Interpolation

Oscar Dalmau
dalmau@cimat.mx

Centro de Investigación en Matemáticas
CIMAT A.C. Mexico

March 2018

Outline

1. Backtracking and Bisection
2. Interpolation
3. Initial step length
4. Barzilai–Borwein

Sufficient decrease and Backtracking

- Choose $\hat{\alpha} > 0$, $\rho \in (0, 1)$, $c_1 \in (0, 1)$, set $\alpha = \hat{\alpha}$
- **Repeat** until $f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha \nabla f_k^T \boldsymbol{d}_k$

  $\alpha = \rho\alpha$
- **end** (repeat)
- Terminate with $\alpha_k = \alpha$

Bisection with Wolfe conditions

- Choose $0 < c_1 < c_2 < 1$
- Set $\alpha = 0$, $\beta = \infty$, $\alpha^i = \alpha_0$, $i = 0$
- **Repeat**

  if $f(\boldsymbol{x}_k + \alpha^i \boldsymbol{d}_k) > f(\boldsymbol{x}_k) + c_1 \alpha^{\nabla} f_k^T \boldsymbol{d}_k$

  $\beta = \alpha^i$ and $\alpha^{i+1} = \frac{1}{2}(\alpha + \beta)$

  else if $\nabla f(\boldsymbol{x}_k + \alpha^i \boldsymbol{d}_k)^T \boldsymbol{d}_k < c_2 \nabla f_k^T \boldsymbol{d}_k$

  $\alpha = \alpha^i$ and $\alpha^{i+1} = \begin{cases} 2\alpha & \text{if } \beta = \infty \\ \frac{1}{2}(\alpha + \beta) & \text{otherwise} \end{cases}$

  otherwise

  break

  $i = i + 1$
- **end** (repeat)
- Terminate with $\alpha_k = \alpha^i$

Steepest decent with backtracking

1. Given $\boldsymbol{x}_0$, $\tau > 0$, $\rho \in (0, 1)$ and $0 < c_1 < 1$
2. Set $k = 0$
3. Compute $\boldsymbol{d}_k = -\boldsymbol{g}_k$
4. **While** (not converge), i.e. $\|\boldsymbol{g}_k\| \geq \tau$
    - Find α_k in the direction $\boldsymbol{d}_k$ with ρ, c_1 using backtracking
    - Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
    - Compute $\boldsymbol{d}_{k+1} = -\boldsymbol{g}_{k+1}$
    - $k = k + 1$
5. Return $\boldsymbol{x}^* = \boldsymbol{x}_k$

Quadratic Interpolation

Suppose that a given $\alpha_0 > 0$ does not satisfies the Armijo's or sufficient descent condition, i.e., $f(\boldsymbol{x}_k + \alpha_0 \boldsymbol{d}_k) > f(\boldsymbol{x}_k) + c_1 \alpha_0 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k$ with $\boldsymbol{d}_k = -\nabla f(\boldsymbol{x}_k)$ and $c_1 = 10^{-4}$, or equivalently

$$\begin{aligned} \phi(\alpha_0) &> \phi(0) + c_1 \alpha_0 \phi'(0) = \ell(\alpha_0) \\ \phi(\alpha) &= f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \\ \ell(\alpha) &= \phi(0) + c_1 \alpha \phi'(0) \end{aligned}$$

Quadratic Interpolation

The next candidate α_1 is computed as the minimum for the quadratic interpolation

$$\phi_q(\alpha) = a\alpha^2 + b\alpha + c$$

satisfying

$$\begin{aligned} \phi_q(0) &= \phi(0) \\ \phi_q'(0) &= \phi'(0) \\ \phi_q(\alpha_0) &= \phi(\alpha_0) \end{aligned}$$

Quadratic Interpolation

$$\phi_q'(\alpha) = 2a\alpha + b$$

Then

$$\begin{aligned}
\phi_q(0) = c &= \phi(0) = f(\boldsymbol{x}_k) \\
\phi_q'(0) = b &= \phi'(0) = \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k \\
\phi_q(\alpha_0) = a\alpha_0^2 + b\alpha_0 + c &= \phi(\alpha_0)
\end{aligned}$$

Quadratic Interpolation

Then

$$
\begin{aligned}
c &= \phi(0) = f(\boldsymbol{x}_k) \\
b &= \phi'(0) = \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k \\
a &= \frac{\phi(\alpha_0) - b\alpha_0 - c}{\alpha_0^2}
\end{aligned}
$$

and

$$
\begin{aligned}
2a\alpha + b &= 0 \\
\alpha^* &= \frac{-b}{2a}
\end{aligned}
$$

$$
\boxed{\alpha^* = \frac{-\alpha_0^2 \phi'(0)}{2(\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0))}}
$$

Quadratic Interpolation

Note that $\alpha^* \in (0, \alpha_0)$ due to $a > 0$ and $b < 0$

$$a = \frac{\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0)}{\alpha_0^2}$$

Let $\tau(\alpha) := \phi'(0)\alpha + \phi(0)$ the tangent to $\phi(\alpha)$ at $\alpha = 0$. as $c_1 = 10^{-4} < 1$ then $c_1\phi'(0)\alpha > \phi'(0)\alpha$ for $\alpha > 0$ due to $\phi'(0) < 0$. Therefore

$$\tau(\alpha) < \ell(\alpha), \text{ for } \alpha > 0$$

then

$$\ell(\alpha_0) > \tau(\alpha_0)$$

Quadratic Interpolation

As α_0 does not satisfies Armijo's Condition and the previous conclusion, we obtain:

$$\begin{aligned} \phi(\alpha_0) &> \phi(0) + c_1\alpha\phi'(0) = \ell(\alpha_0) \\ \ell(\alpha_0) &> \tau(\alpha_0) \end{aligned}$$

then $\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0) = \phi(\alpha_0) - \tau(\alpha_0) > 0$. And

$$a = \frac{\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0)}{\alpha_0^2} > 0$$

i.e., $\phi_q(\alpha)$ is a convex parabola (an open upward parabola).

Quadratic Interpolation

In order to proof that $\alpha^* < \alpha_0$ it is sufficient to proof that the slope of $\phi_q(\alpha)$ and α_0 is positive, i.e., α_0 is at the right side of the minimum α^* or it is located at the increasing region of $\phi_q(\alpha)$.

$$
\begin{aligned}
\phi_q'(\alpha_0) &= 2a\alpha_0 + b \\
&= 2\frac{\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0)}{\alpha_0^2}\alpha_0 + \phi(0) \\
&= 2\frac{\phi(\alpha_0) - \delta\phi'(0)\alpha_0 - \phi(0)}{\alpha_0}
\end{aligned}
$$

with $\delta = 0.5$.

Quadratic Interpolation

As $\delta = 0.5 > c_1$ then

$$\phi(0) + c_1\alpha\phi'(0) > \delta\phi'(0)\alpha_0 + \phi(0)$$

and due to α_0 does not satisfies Armijo's Condition, i.e.

$$\phi(\alpha_0) \quad > \quad \phi(0) + c_1\alpha\phi'(0)$$

we obtain

$$\phi(\alpha_0) \quad > \quad \delta\phi'(0)\alpha_0 + \phi(0)$$

Quadratic Interpolation

Therefore

$$\phi(\alpha_0) - \delta\phi'(0)\alpha_0 - \phi(0) \quad > \quad 0$$

and

$$\phi_q'(\alpha_0) > 0$$

due to $\alpha_0 > 0$. We conclude that $\alpha^* < \alpha_0$. And finally $\alpha^* \in (0, \alpha_0)$ and we can select $\alpha_1 = \alpha^*$

Quadratic Interpolation: Algorithm

Given $\alpha_0 > 0$, $\phi(\cdot), \phi'(\cdot)$. ($\alpha_0$ does no satisfy Armijo's condition)

$\alpha_1 = \frac{-\alpha_0^2 \phi'(0)}{2(\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0))}$

while $\phi(\alpha_1) > \phi(0) + c_1 \alpha_1 \phi'(0)$:

$\alpha_0 = \alpha_1$

$\alpha_1 = \frac{-\alpha_0^2 \phi'(0)}{2(\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0))}$

return α_1

Note: In practice we simply set $\alpha_0 = \frac{-\alpha_0^2 \phi'(0)}{2(\phi(\alpha_0) - \phi'(0)\alpha_0 - \phi(0))}$

Cubic Interpolation

- Suppose that $\alpha_0 > 0$ does not satisfies the Armijo's or sufficient descent condition and, α_1, obtained by the quadratic interpolation algorithm, does not satisfies the Armijo's condition.
- We can compute α_2 by using cubic interpolation.

Cubic Interpolation

The next candidate α_2 can be computed as the minimum of the Cubic interpolation

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d$$

satisfying

$$\begin{aligned} \phi_c(0) &= \phi(0) \\ \phi_c'(0) &= \phi'(0) \\ \phi_c(\alpha_0) &= \phi(\alpha_0) \\ \phi_c(\alpha_1) &= \phi(\alpha_1) \end{aligned}$$

Cubic Interpolation

From

$$
\begin{aligned}
\phi_c(\alpha) &= a\alpha^3 + b\alpha^2 + c\alpha + d \\
\phi_c'(\alpha) &= 3a\alpha^2 + b\alpha + c
\end{aligned}
$$

and the conditions

$$
\begin{aligned}
\phi_c(0) &= \phi(0) \\
\phi_c'(0) &= \phi'(0)
\end{aligned}
$$

we obtain that

$$
\begin{aligned}
d &= \phi(0) \\
c &= \phi'(0) < 0
\end{aligned}
$$

Cubic Interpolation

From the conditions

$$\begin{aligned} \phi_q(\alpha_0) &= \phi(\alpha_0) \\ \phi_q(\alpha_1) &= \phi(\alpha_1) \end{aligned}$$

we obtain that

$$\begin{aligned} \alpha_1^3 a + \alpha_1^2 b &= \phi_c(\alpha_1) - \phi'(0)\alpha_1 - \phi(0) \\ \alpha_0^3 a + \alpha_0^2 b &= \phi_c(\alpha_0) - \phi'(0)\alpha_0 - \phi(0) \end{aligned}$$

Note: The right side of both equations is positive due to α_0, α_1 do not satisfy Armijo (see proof in quadratic interpolation case)

Cubic Interpolation

Solving

$$\begin{array}{rcl}\alpha_1^3 a + \alpha_1^2 b & = & \phi_c(\alpha_1) - \phi'(0)\alpha_1 - \phi(0) \\ \alpha_0^3 a + \alpha_0^2 b & = & \phi_c(\alpha_0) - \phi'(0)\alpha_0 - \phi(0)\end{array}$$

for a, b

$$\left[\begin{array}{c} a \\ b \end{array}\right] = \frac{1}{\alpha_1^2\alpha_0^2(\alpha_1 - \alpha_0)}\left[\begin{array}{cc} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{array}\right]\left[\begin{array}{c} \phi_c(\alpha_1) - \phi'(0)\alpha_1 - \phi(0) \\ \phi_c(\alpha_0) - \phi'(0)\alpha_0 - \phi(0) \end{array}\right]$$

Note: Since $0 < \alpha_1 < \alpha_0$ and $\phi_c(\alpha_i) - \phi'(0)\alpha_i - \phi(0) > 0$, $i = 0, 1$, it can be proof that a and b have different sign.

Cubic Interpolation

We already know a, b, c and d then we can compute the minimum of $\phi_c(\cdot)$.

$$\phi_c'(\alpha) = 3a\alpha^2 + 2b\alpha + c = 0$$

and

$$\alpha^* = \frac{-b + \sqrt{b^2 - 3ac}}{3a}$$

is the minimum, i.e., it can be proof that $\phi_c''(\alpha^*) > 0$ and $\alpha^* > 0$, due to a, b have different sign, and we set $\alpha_2 = \alpha^*$

Cubic Interpolation: Algorithm

Given $\alpha_0, \alpha_1 > 0$, $\phi(\cdot), \phi'(\cdot)$. ($\alpha_0, \alpha_1$ do no satisfy Armijo's condition and α_1 is computed with the quadratic interpolation algorithm)

$\alpha_2 = \frac{-b+\sqrt{b^2-3ac}}{3a}$

while $\phi(\alpha_2) > \phi(0) + c_1 \alpha_2 \phi'(0)$:

$\alpha_0 = \alpha_1$, $\alpha_1 = \alpha_2$

$\alpha_2 = \frac{-b+\sqrt{b^2-3ac}}{3a}$

return α_2

Initial step length

1. For Newton and quasi Newton methods, the step $\alpha_0 = 1$ should always be used as the initial trial step length.
2. Another option is to assume that the first-order change in the function at iterate $\boldsymbol{x}_k$ will be the same as that obtained at the previous step, i.e., select the initial guess α_0 that satisfies

$$\alpha_0 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k \quad = \quad \alpha_{k-1} \nabla f(\boldsymbol{x}_{k-1})^T \boldsymbol{d}_{k-1}$$

Initial step length: $\boxed{\alpha_0 = \dfrac{\alpha_{k-1} \nabla f(\boldsymbol{x}_{k-1})^T \boldsymbol{d}_{k-1}}{\nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k}}$

Initial step length

Another strategy is to compute the quadratic interpolation of $f(\boldsymbol{x}_{k-1})$, $\nabla f(\boldsymbol{x}_{k-1})^T \boldsymbol{d}_{k-1}$ and $f(\boldsymbol{x}_k)$ where α_{k-1} is the minimun, i.e, $\phi(0)$, $\phi'(0)$ and $\phi(\alpha_{k-1}) = f(\boldsymbol{x}_k)$. Then,

$$
\begin{aligned}
\phi_q(\alpha) &= a\alpha^2 + \alpha b + \phi(0) \\
a &= \frac{\phi(\alpha_{k-1}) - \phi'(0)\alpha_{k-1} - \phi(0)}{\alpha_{k-1}^2} \\
b &= \phi'(0)
\end{aligned}
$$

Initial step length

If α_{k-1} is the minimum and $f(\boldsymbol{x}_k)$ its corresponding minimum value, then

$$\begin{aligned} 2a\alpha_{k-1} + b = 0; \quad \alpha_{k-1} &= -\frac{b}{2a} \\ f(\boldsymbol{x}_k) &= \phi(\alpha_{k-1}) \end{aligned}$$

$$\begin{aligned} f(\boldsymbol{x}_k) &= a\alpha_{k-1}^2 + \alpha_{k-1}b + f(\boldsymbol{x}_{k-1}) = \alpha_{k-1}(a\alpha_{k-1} + b) + f(\boldsymbol{x}_{k-1}) \\ &= \alpha_{k-1}(-ab/(2a) + b) + f(\boldsymbol{x}_k) = b\alpha_{k-1}/2 + f(\boldsymbol{x}_{k-1}) \\ \alpha_{k-1} &= 2\frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1})}{\phi'(0)} \end{aligned}$$

Initial step length: $\boxed{\alpha_0 = 2\frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1})}{\nabla f(\boldsymbol{x}_{k-1})^T \boldsymbol{d}_{k-1}}}$

Barzilai–Borwein

The Barzilai–Borwein method (BB) chooses α_k so that $\alpha_k \boldsymbol{g}_k$ approximates $\mathbf{H}_k^{-1}\boldsymbol{g}_k$ without computing $\mathbf{H}_k$.

Barzilai–Borwein: quadratic case

Let the quadratic problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T \mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

with $\mathbf{Q} \succ \mathbf{0}$ and symmetric. Then

- $\boldsymbol{g}_k = \mathbf{Q}\boldsymbol{x}_k - \boldsymbol{b}$
- $\mathbf{H}_k = \mathbf{Q}$
- Newton step: $\boldsymbol{d}_k = -\mathbf{Q}^{-1}\boldsymbol{g}_k$

Barzilai–Borwein: quadratic case

Let the quadratic problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T \mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

with $\mathbf{Q} \succ \mathbf{0}$ and symmetric.

- Goal: choose α_k so that $-\alpha_k \boldsymbol{g}_k$ approximates the Newton step, i.e., $-\alpha_k \boldsymbol{g}_k \approx \boldsymbol{d}_k = -\mathbf{Q}^{-1}\boldsymbol{g}_k$, then $(\alpha_k^{-1}\mathbf{I})\boldsymbol{g}_k \approx \mathbf{Q}\boldsymbol{g}_k$

Note that

- $\boldsymbol{g}_k - \boldsymbol{g}_{k-1} = \mathbf{Q}\boldsymbol{x}_k - \boldsymbol{b} - (\mathbf{Q}\boldsymbol{x}_{k-1} - \boldsymbol{b}) = \mathbf{Q}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$

Then Note that

- $\mathbf{Q}\boldsymbol{s}_{k-1} = \boldsymbol{y}_{k-1}$

with $\boldsymbol{s}_{k-1} \overset{def}{=} \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\boldsymbol{y}_{k-1} \overset{def}{=} \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$

Barzilai–Borwein: quadratic case

As $(\alpha_k^{-1}\mathbf{I})\boldsymbol{g}_k \approx \mathbf{Q}\boldsymbol{g}_k$ and from

- $\mathbf{Q}\boldsymbol{s}_{k-1} = \boldsymbol{y}_{k-1}$

one obtains (changing $\mathbf{Q}$ by $\alpha_k^{-1}\mathbf{I}$)

- $\alpha_k^{-1}\boldsymbol{s}_{k-1} = \boldsymbol{y}_{k-1}$ or $\boldsymbol{s}_{k-1} = \boldsymbol{y}_{k-1}\alpha_k$

then α_k can be estimated using least square with both alternatives:

$$
\begin{aligned}
\alpha_k^{-1} &= \arg\min_{\alpha} \|\alpha\boldsymbol{s}_k - \boldsymbol{y}_k\|^2 \;\Rightarrow\; \alpha_k^1 = \frac{\boldsymbol{s}_{k-1}^T\boldsymbol{s}_{k-1}}{\boldsymbol{s}_{k-1}^T\boldsymbol{y}_{k-1}} = \frac{1}{\alpha_k^{BB1}} \\
\alpha_k &= \arg\min_{\alpha} \|\boldsymbol{s}_k - \boldsymbol{y}_k\alpha\|^2 \;\Rightarrow\; \alpha_k^2 = \frac{\boldsymbol{s}_{k-1}^T\boldsymbol{y}_{k-1}}{\boldsymbol{y}_{k-1}^T\boldsymbol{y}_{k-1}} = \frac{1}{\alpha_k^{BB2}}
\end{aligned}
$$

We call α_k^1 (see R. Fletcher paper) and α_k^2 the **Barzilai–Borwein step sizes**.

1. Since $\boldsymbol{x}_{k-1}$ and $\boldsymbol{g}_{k-1}$ and thus $\boldsymbol{s}_{k-1} = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\boldsymbol{y}_{k-1} = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$ are unavailable at $k = 0$, we apply the standard gradient descent at $k = 0$ and start BB at $k = 1$
2. We can use either α_k^1 or α_k^2 or alternate between them
3. We can fix $\alpha_k = \alpha_k^1$ or $\alpha_k = \alpha_k^2$ for a few consecutive steps.
4. It performs very well on minimizing quadratic and many other functions.

Barzilai–Borwein descent Algorithm

Given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $0 < \epsilon << 1$

$k = 0$

while $\|\nabla f(\boldsymbol{x}_k)\| > \epsilon$:

  If $k == 0$:

    Compute α_0 using a line search algorithm

  else :

    Compute the **step size** α_k using BB (α_k^1 or α_k^2)

  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k$

  $k = k + 1$

return $\boldsymbol{x}_k$

Non-monotone Barzilai–Borwein

1. We can combine BB with non-monotone line search
2. See: Marcos Raydan Algorithm (Non-Monotone Barzilai and Borwein Gradient Method)
3. Reference: Marcos Raydan."The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem". 1997. **see next slide**.

Global Barzilai and Borwein (GBB) Algorithm

Given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $0 < \epsilon << 1$, $M \in N$, $c_1 \in (0,1)$, $\delta > 0$
$0 < \rho_1 < \rho_2 < 1$, α^l, α^u and set $k = 0$
while $\|\nabla f(\boldsymbol{x}_k)\| > \epsilon$:
    If $\alpha_k \leq \alpha^l$ or $\alpha_k \geq \alpha^u$: $\alpha_k = \delta$
    $\alpha = \alpha_k$ and $\boldsymbol{d}_k = -\boldsymbol{g}_k$
    while $f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) > \max_{0 \leq j \leq \min(k,M)} f(\boldsymbol{x}_{k-j}) + c_1 \alpha \boldsymbol{g}_k^T \boldsymbol{d}_k$:
        $\alpha = \rho\alpha$, $\rho \in [\rho_1, \rho_2]$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{d}_k$
    $\alpha_k = \alpha_k^1$ using BB ($\alpha_k^1 = \frac{1}{\alpha_k^{BB1}}$ or $\alpha_k^2 = \frac{1}{\alpha_k^{BB2}}$)
    $k = k + 1$
return $\boldsymbol{x}_k$

Non-monotone Barzilai–Borwein

1. We can combine BB with non-monotone line search, **see next slide**. Hongchao Zhang And William W. Hager. "A Nonmonotone Line Search Technique And Its Application To Unconstrained Optimization". SIAM J. OPTIM. Society for Industrial and Applied Mathematics Vol. 14, No. 4, pp. 1043-1056, 2004.

Zhang-Hager nonmonotone line search

Given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $0 < \epsilon << 1$, $c_1, \eta \in (0,1)$, $C_0 = f(\boldsymbol{x}_0)$ and set $k = 0$

while $\|\nabla f(\boldsymbol{x}_k)\| > \epsilon$:

  Compute α_k that satisfies Armijo's condition:

    sufficient descent $f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \le C_k + c_1 \alpha \boldsymbol{g}_k^T \boldsymbol{d}_k$

  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

  $Q_{k+1} = \eta Q_k + 1$, $C_{k+1} = (\eta Q_k C_k + f(\boldsymbol{x}_{k+1}))/Q_{k+1}$

  $k = k + 1$

return $\boldsymbol{x}_k$

Zhang-Hager nonmonotone line search.

1. If $\eta = 1$ then $C_k = \frac{1}{k+1} \sum_{j=0}^{k} f(\boldsymbol{x}_j)$
2. If $\eta < 1$ then C_k is the weighted sum of the past $f(\boldsymbol{x}_j)$ assigning greater weights to the recents $f(\boldsymbol{x}_j)$
3. See more details in the paper: Hongchao Zhang And William W. Hager. "A Nonmonotone Line Search Technique And Its Application To Unconstrained Optimization". SIAM J. OPTIM. Society for Industrial and Applied Mathematics Vol. 14, No. 4, pp. 1043-1056, 2004.

1847 A. Cauchy,'Me'thodes ge'ne'rales pour la re'solution des syste'mes d'e'quations simultane'es,' C. R. Acad. Sci. Par., vol. 25, pp. 536-538, 1847.

1959 H.Akaike,'On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method,' Ann. Inst. Statist. Math. Tokyo, vol. 11, pp. 1-16, 1959.

1986 Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. A nonmonotone line search technique for Newton's method. SIAM Journal on Numerical Analysis, 23(4): 707-716, 1986.

1988 J. Barzilai and J.M. Borwein, 'Two point step size gradient methods,' IMA J. Numer. Anal., vol. 8, pp. 141-148, 1988.

1993 M.Raydan,'On the Barzilai and Borwein choice of step length for the gradient method,'IMAJ.Numer.Anal., vol. 13, pp. 321-326, 1993.

1997 Marcos Raydan. 'The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem'

1999 Y.H. Dai and L.Z. Liao,'R-linear convergence of the Barzilai and Borwein gradient method,' Technical Report AMSS 1999-081, Academy of Mathematics and Systems Sciences, Beijing, China, 1999.

2001 Roger Fletcher. On the Barzilai and Borwein Method.

2001 M. Raydan and B. F. Svaiter. Relaxed Steepest Descent and Cauchy-Barzilai- Borwein Method

2004 Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. SIAM Journal on Optimization, 14(4): 1043-1056, 2004.

2013 Yu-Hong Dai. A new analysis on the barzilai-borwein gradient method. Journal of the Operations Research Society of China, pages 1-12, 2013.

2015 Shuai Huang and Zhong Wan and Xiaohong Chen. 'A new nonmonotone line search technique for unconstrained optimization'

2016 Frank E. Curtis and Wei Guo. R-Linear Convergence of Limited Memory Steepest Descent. Note: The Limited Memory Steepest Descent was proposed by Roger Fletcher

2017 Yutao Zheng, Bing Zheng . A New Modified Barzilai-Borwein Gradient Method for the Quadratic Minimization Problem. Journal of Optimization Theory and Applications (2017)