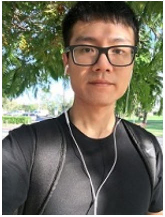
A Newton-MR Algorithm with Complexity Guarantee for Non-Convex Problems

Fred Roosta

School of Mathematics and Physics
University of Queensland



Yang Liu (UQ)

Michael Mahoney (Berkeley)

Peng Xu (Stanford)

Problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x})$$

$f : \mathbb{R}^d \to \mathbb{R}$ is

- twice differentiable
- (potentially) non-convex

Algorithm Generic 2^{nd}-order Method

```
Start from $\boldsymbol{x}_0$
```

Algorithm Generic 2$^{\text{nd}}$-order Method

```
Start from $\boldsymbol{x}_0$
for $k = 1, 2, \ldots$ do
end for
```

Algorithm Generic 2nd-order Method

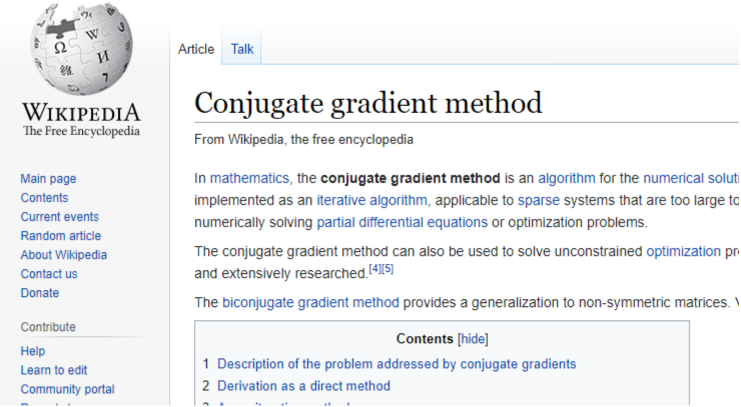
```
Start from $\boldsymbol{x}_0$
for $k = 1, 2, \ldots$ do
    $\boldsymbol{p}_k = \begin{cases} \alpha_k \boldsymbol{p} \quad \text{where} \quad \boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k & \text{(Line Search)} \\ & \\ & \end{cases}$
end for
```

Algorithm Generic 2nd-order Method

```
Start from $\boldsymbol{x}_0$
for $k = 1, 2, \ldots$ do
    $$\boldsymbol{p}_k = \begin{cases} \alpha_k \boldsymbol{p} \quad \text{where} \quad \boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k & \text{(Line Search)} \\ \underset{\|\boldsymbol{p}\| \leq \Delta}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle / 2 & \text{(Trust Region)} \end{cases}$$
end for
```

Algorithm Generic 2$^{\text{nd}}$-order Method

```
Start from $\boldsymbol{x}_0$
for $k = 1, 2, \ldots$ do
    $\boldsymbol{p}_k = \begin{cases} \alpha_k \boldsymbol{p} \quad \text{where} \quad \boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k & \text{(Line Search)} \\ \underset{\|\boldsymbol{p}\| \leq \Delta}{\arg\min} \; \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle / 2 & \text{(Trust Region)} \end{cases}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$
end for
```



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute

Help
Learn to edit
Community portal

Article Talk

Conjugate gradient method

From Wikipedia, the free encyclopedia

In mathematics, the **conjugate gradient method** is an algorithm for the numerical soluti
implemented as an iterative algorithm, applicable to sparse systems that are too large to
numerically solving partial differential equations or optimization problems.

The conjugate gradient method can also be used to solve unconstrained optimization pr
and extensively researched.[4][5]

The biconjugate gradient method provides a generalization to non-symmetric matrices. \

Contents [hide]

CG
Minres

Q: Why CG?

Why CG?

Why CG?

- Let's consider the simple case of $\mathbf{H}_k \mathbf{p} \approx -\mathbf{g}_k$

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow \boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow \boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t} \; \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

↓

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t} \; \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

$$\Big\downarrow$$

$$\langle \boldsymbol{p}, \boldsymbol{g}_k \rangle \leq -\frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t} \; \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

$$\Big\downarrow$$

$$\langle \boldsymbol{p}, \boldsymbol{g}_k \rangle \leq -\frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle \; < 0$$

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \; \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

$$\downarrow$$

$$\langle \boldsymbol{p}, \boldsymbol{g}_k \rangle \leq -\frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle \; < 0$$

$$\downarrow$$

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

$$\downarrow$$

$$\langle \boldsymbol{p}, \boldsymbol{g}_k \rangle \leq -\frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle < 0$$

$$\downarrow$$

$\boldsymbol{p}^{(t)}$ is a descent direction for $f(\boldsymbol{x})$ <u>for all t!</u>

Why CG?

- Let's consider the simple case of $\boldsymbol{H}_k \boldsymbol{p} \approx -\boldsymbol{g}_k$
- When f is strongly convex $\Longrightarrow$ $\boldsymbol{H}_k$ is SPD
- More subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g}_k \rangle + \frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle$$

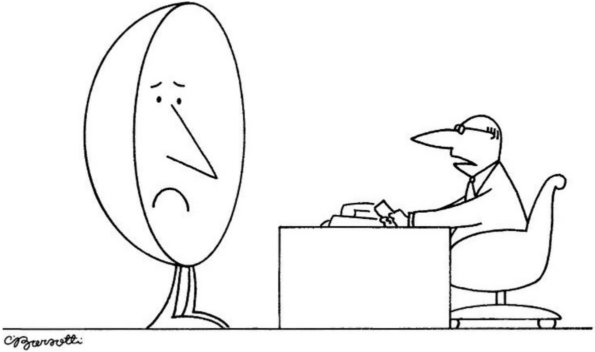
$$\Bigg\downarrow$$

$$\langle \boldsymbol{p}, \boldsymbol{g}_k \rangle \leq -\frac{1}{2} \langle \boldsymbol{p}, \boldsymbol{H}_k \boldsymbol{p} \rangle < 0$$

$$\Bigg\downarrow$$

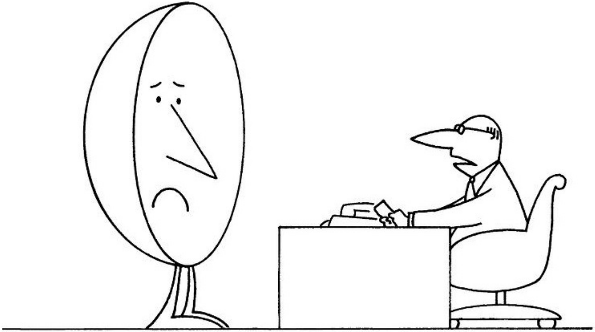
$\boldsymbol{p}^{(t)}$ is a descent direction for $f(\boldsymbol{x})$ for all t!

Classical Newton's Method



"Actually, the job calls for someone who is convex."

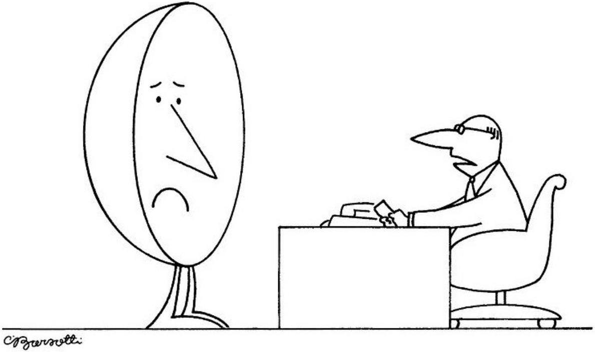
Classical Newton's Method



"Actually, the job calls for someone who is convex."

But...what if the Hessian is indefinite and/or singular?

Classical Newton's Method

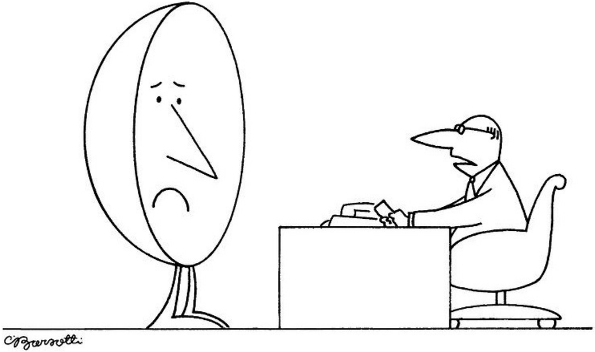


"Actually, the job calls for someone who is convex."

But...what if the Hessian is indefinite and/or singular?

- Indefinite Hessian $\Longrightarrow$ Unbounded sub-problem

Classical Newton's Method



"Actually, the job calls for someone who is convex."

But...what if the Hessian is indefinite and/or singular?

- Indefinite Hessian $\Longrightarrow$ Unbounded sub-problem
- $\boldsymbol{g} \notin \text{Range}(\boldsymbol{H}) \Longrightarrow$ Unbounded sub-problem

$$\min_{\mathbf{p}\in\mathbb{R}^d} \|\mathbf{H}_k \mathbf{p} + \mathbf{g}_k\|$$

The underlying matrix in OLS is

$$\min_{\boldsymbol{p}\in\mathbb{R}^d} \|\boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k\|$$

The underlying matrix in OLS is

- symmetric

$$\min_{\boldsymbol{p}\in\mathbb{R}^d} \|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|$$

The underlying matrix in OLS is

- symmetric
- (possibly) indefinite

$$\min_{\boldsymbol{p}\in\mathbb{R}^d} \|\boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k\|$$

The underlying matrix in OLS is

- symmetric
- (possibly) indefinite
- (possibly) singular

$$\min_{\boldsymbol{p}\in\mathbb{R}^d} \|\boldsymbol{H}_k\boldsymbol{p}+\boldsymbol{g}_k\|$$

The underlying matrix in OLS is

- symmetric
- (possibly) indefinite
- (possibly) singular
- (possibly) ill-conditioned

$$\min_{\boldsymbol{p}\in\mathbb{R}^d} \|\boldsymbol{H}_k\boldsymbol{p}+\boldsymbol{g}_k\|$$

The underlying matrix in OLS is

- symmetric
- (possibly) indefinite
- (possibly) singular
- (possibly) ill-conditioned

MINRES-type OLS Solvers

Newton-MR-type Algorithms

A class of Newton-type algorithms with MINRES as sub-problem solver

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \frac{1}{2} \| \boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k \|^2$$

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \arg\min_{\boldsymbol{p}\in\mathcal{K}_t} \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \frac{1}{2} \| \boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k \|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

Sub-problems of MINRES:

$$\mathbf{p}^{(t)} = \underset{\mathbf{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\mathbf{p}^{(t)} = \underset{\mathbf{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\mathbf{H}_k\mathbf{p} + \mathbf{g}_k\|^2$$

$$\Big\downarrow$$

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t} \frac{1}{2} \| \boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k \|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t} \frac{1}{2} \| \boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k \|^2$$

$$\Big\downarrow$$

$$\left\langle \boldsymbol{p}^{(t)}, \boldsymbol{H}_k \boldsymbol{g}_k \right\rangle \leq -\frac{1}{2} \| \nabla^2 f(\boldsymbol{x}_k) \boldsymbol{p}^{(t)} \|^2$$

Sub-problems of MINRES:

$$\mathbf{p}^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t}{\arg\min} \frac{1}{2} \| \mathbf{H}_k \mathbf{p} + \mathbf{g}_k \|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\mathbf{p}^{(t)} = \underset{\mathbf{p} \in \mathcal{K}_t}{\arg\min} \ \frac{1}{2} \| \mathbf{H}_k \mathbf{p} + \mathbf{g}_k \|^2$$

$$\Big\downarrow$$

$$\left\langle \mathbf{p}^{(t)}, \mathbf{H}_k \mathbf{g}_k \right\rangle \leq -\frac{1}{2} \| \nabla^2 f(\mathbf{x}_k) \mathbf{p}^{(t)} \|^2 < 0$$

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

$$\Big\downarrow$$

$$\left\langle \boldsymbol{p}^{(t)}, \boldsymbol{H}_k\boldsymbol{g}_k \right\rangle \leq -\frac{1}{2}\|\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}^{(t)}\|^2 < 0$$

$$\Big\downarrow$$

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \frac{1}{2} \|\boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \; \frac{1}{2} \|\boldsymbol{H}_k \boldsymbol{p} + \boldsymbol{g}_k\|^2$$

$$\Big\downarrow$$

$$\left\langle \boldsymbol{p}^{(t)}, \boldsymbol{H}_k \boldsymbol{g}_k \right\rangle \leq -\frac{1}{2} \|\nabla^2 f(\boldsymbol{x}_k) \boldsymbol{p}^{(t)}\|^2 < 0$$

$$\Big\downarrow$$

$\boldsymbol{p}^{(t)}$ is a descent direction for $\|\boldsymbol{g}\|^2$ for all t!

Sub-problems of MINRES:

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

- There is always a solution (sometimes infinitely many)
- But more subtly...

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min}\ \frac{1}{2}\|\boldsymbol{H}_k\boldsymbol{p} + \boldsymbol{g}_k\|^2$$

$$\Downarrow$$

$$\left\langle \boldsymbol{p}^{(t)}, \boldsymbol{H}_k\boldsymbol{g}_k \right\rangle \leq -\frac{1}{2}\|\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}^{(t)}\|^2 < 0$$

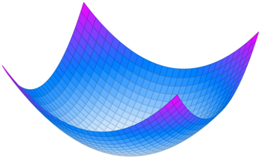
$$\Downarrow$$

$\boldsymbol{p}^{(t)}$ is a descent direction for $\|\boldsymbol{g}\|^2$ for all t!

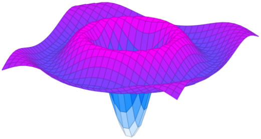
$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \quad f(\boldsymbol{x})$$

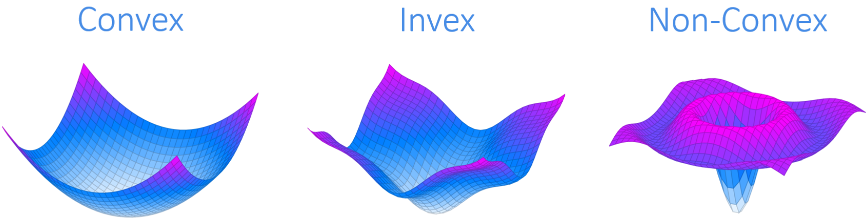
$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad \|\boldsymbol{g}\|$$

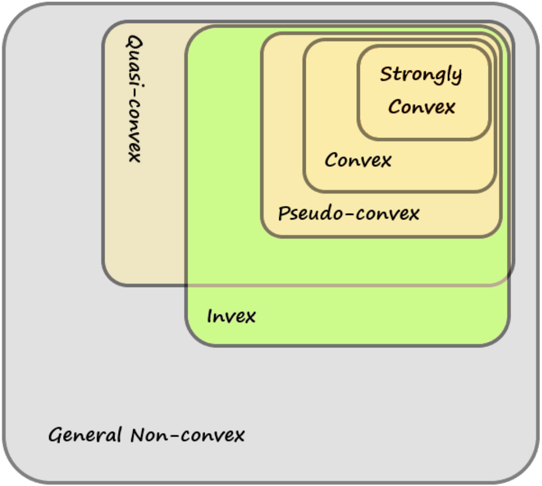
Convex



Non-Convex




Convex
Invex
Non-Convex


Quasi-convex
Strongly Convex
Convex
Pseudo-convex
Invex
General Non-convex

Algorithm Newton-MR (Invex)

```
Input: $\boldsymbol{x}_0$, $0 < \tau < 1$, $0 < \rho < 1$
for $k = 0, 1, 2, \ldots$ until $\|\boldsymbol{g}_k\| \leq \tau$ do
    $\boldsymbol{p}_k \approx -\boldsymbol{H}_k^{\dagger}\boldsymbol{g}_k$
    Find $\alpha_k$ such that $\|\boldsymbol{g}_{k+1}\|^2 \leq \|\boldsymbol{g}_k\|^2 + 2\rho\alpha_k \langle \boldsymbol{p}_k, \boldsymbol{H}_k\boldsymbol{g}_k \rangle$
    Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
end for
Output: $\boldsymbol{x}$ for which $\|\boldsymbol{g}_k\| \leq \tau$
```

Examples of Convergence Results

Global Linear Rate in "$\|\boldsymbol{g}\|$"

$$\left\|\boldsymbol{g}^{(k+1)}\right\|^2 \leq (1-\eta)\left\|\boldsymbol{g}_k\right\|^2, \quad 0 < \eta \leq 1.$$

Global Linear Rate in "$f(\boldsymbol{x}) - \min_{\boldsymbol{x}} f$" Under Polyak-Łojasiewicz

$$f(\boldsymbol{x}_k) - \min_{\boldsymbol{x}} f \leq C\zeta^k, \quad 0 < \zeta \leq 1.$$

Error Recursion with $\alpha_k = 1$

$$\min_{\boldsymbol{y}\in\mathcal{X}^\star} \|\boldsymbol{x}_{k+1} - \boldsymbol{y}\| \leq c_1 \min_{\boldsymbol{y}\in\mathcal{X}^\star} \|\boldsymbol{x}_k - \boldsymbol{y}\|^2 + \sqrt{(1-\nu)}c_2 \min_{\boldsymbol{y}\in\mathcal{X}^\star} \|\boldsymbol{x}_k - \boldsymbol{y}\|.$$

Inexact Hessian

$$\tilde{\boldsymbol{H}} \approx \boldsymbol{H}$$

Inexact Hessian

$$\left\| \tilde{\boldsymbol{H}} - \boldsymbol{H} \right\| \leq \epsilon$$

Finite-sum Optimization

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

Finite-sum Optimization

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

$$\boldsymbol{H} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(\boldsymbol{x}).$$

Finite-sum Optimization

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

$$\tilde{\boldsymbol{H}} = \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \nabla^2 f_j(\boldsymbol{x}),$$

Finite-sum Optimization

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

$$|\mathcal{S}| \in \mathcal{O}\left(\epsilon^{-2}\log\left(\frac{2d}{\delta}\right)\right)$$

Finite-sum Optimization

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

$$\mathbb{P}\left(\left\|\tilde{\mathbf{H}} - \mathbf{H}\right\| \leq \epsilon\right) \geq 1 - \delta$$

Newton-MR with Inexact Hessian

Algorithm Newton-MR With Inexact Hessian Information

```
**Input:** $\boldsymbol{x}_0$, $0 < \tau < 1$, $0 < \rho < 1$
**for** $k = 0, 1, 2, \ldots$ until $\|\boldsymbol{g}_k\| \leq \tau$ **do**
    $\boldsymbol{p}_k \approx -\tilde{\boldsymbol{H}}_k^{\dagger}\boldsymbol{g}_k$
    Find $\alpha_k$ such that $\|\boldsymbol{g}_{k+1}\|^2 \leq \|\boldsymbol{g}_k\|^2 + 2\rho\alpha_k \left\langle \boldsymbol{p}_k, \tilde{\boldsymbol{H}}_k\boldsymbol{g}_k \right\rangle$
    Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k$
**end for**
**Output:** $\boldsymbol{x}$ for which $\|\boldsymbol{g}_k\| \leq \tau$
```

Recall: Newton's Method

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{H}_k^{-1} \boldsymbol{g}_k$$

Recall: Newton's Method w. Inexact Hessian

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \tilde{\boldsymbol{H}}_k^{-1} \boldsymbol{g}_k$$

Recall: Newton's Method w. Inexact Hessian

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \tilde{\boldsymbol{H}}_k^{-1} \boldsymbol{g}_k$$

$$(1 - \tilde{\epsilon}_1)\boldsymbol{H} \preceq \tilde{\boldsymbol{H}} \preceq (1 + \tilde{\epsilon}_1)\boldsymbol{H}$$

Recall: Newton's Method w. Inexact Hessian

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \tilde{\boldsymbol{H}}_k^{-1} \boldsymbol{g}_k$$

$$(1 - \tilde{\epsilon}_2)\boldsymbol{H}^{-1} \preceq \tilde{\boldsymbol{H}}^{-1} \preceq (1 + \tilde{\epsilon}_1)\boldsymbol{H}^{-1}$$

Recall: Newton's Method w. Inexact Hessian

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \tilde{\mathbf{H}}_k^{-1} \mathbf{g}_k$$

$$\left\| \tilde{\mathbf{H}}^{-1} - \mathbf{H}^{-1} \right\| \leq \tilde{\epsilon}_3$$

Newton-MR Method

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{H}_k^{\dagger} \boldsymbol{g}_k$$

Newton-MR Method w. Inexact Hessian

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \tilde{\boldsymbol{H}}_k^{\dagger} \boldsymbol{g}_k$$

Newton-MR Method w. Inexact Hessian

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \tilde{\boldsymbol{H}}_k^{\dagger} \boldsymbol{g}_k$$

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \tilde{\epsilon}_3 \quad \textbf{(?)}$$

$$\lim_{\epsilon \to 0} \tilde{\boldsymbol{H}}^{\dagger} = \boldsymbol{H}^{\dagger}$$

$$\lim_{\epsilon \to 0} \tilde{\boldsymbol{H}}^{\dagger} = \boldsymbol{H}^{\dagger} \iff \text{Rank}(\tilde{\boldsymbol{H}}) = \text{Rank}(\boldsymbol{H})$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\lim_{\epsilon \to 0} \tilde{\boldsymbol{H}}^{\dagger} = \boldsymbol{H}^{\dagger} \iff \operatorname{Rank}(\tilde{\boldsymbol{H}}) = \operatorname{Rank}(\boldsymbol{H})$$

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \tilde{\epsilon}_3 \quad ✗$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \left(\frac{1+\sqrt{5}}{2} \right) \max \left\{ \left\| \boldsymbol{H}^{\dagger} \right\|^{2}, \left\| \tilde{\boldsymbol{H}}^{\dagger} \right\|^{2} \right\} \epsilon$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \left(\frac{1+\sqrt{5}}{2} \right) \max \left\{ \left\| \boldsymbol{H}^{\dagger} \right\|^{2}, \left\| \tilde{\boldsymbol{H}}^{\dagger} \right\|^{2} \right\} \epsilon$$

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} \right\| \in o\left(\frac{1}{\sqrt{\epsilon}} \right)$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \left(\frac{1+\sqrt{5}}{2} \right) \max \left\{ \left\| \boldsymbol{H}^{\dagger} \right\|^{2}, \left\| \tilde{\boldsymbol{H}}^{\dagger} \right\|^{2} \right\} \epsilon$$

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} \right\| \in o\left(\frac{1}{\sqrt{\epsilon}} \right) \qquad \text{✗}$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\left\| \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}^{\dagger} \right\| \leq \left(\frac{1+\sqrt{5}}{2} \right) \max \left\{ \left\| \boldsymbol{H}^{\dagger} \right\|^2, \left\| \tilde{\boldsymbol{H}}^{\dagger} \right\|^2 \right\} \epsilon$$

$$\| \tilde{\boldsymbol{H}}^{\dagger} \| \in \mathcal{O}\left(\frac{1}{\epsilon} \right)$$

[Matrix Perturbation Theory, Gilbert W. Stewart and Ji-guang Sun]

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle$$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = -\left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle$$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = -\left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle = -\left\| \boldsymbol{U}\boldsymbol{U}^{\intercal}\boldsymbol{g} \right\|$$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = -\left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle = -\left\| \boldsymbol{U}\boldsymbol{U}^{\intercal}\boldsymbol{g} \right\|$$

$$\left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{U}\boldsymbol{U}^{\intercal} \right\| \leq \tilde{\epsilon}_3 \quad \textbf{(?)}$$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = -\left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle = -\left\| \boldsymbol{U}\boldsymbol{U}^{\intercal}\boldsymbol{g} \right\|$$

$$\left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{U}\boldsymbol{U}^{\intercal} \right\| \leq \tilde{\epsilon}_3 \quad \textbf{(?)}$$

$\text{Rank}(\tilde{\boldsymbol{H}}) \neq \text{Rank}(\boldsymbol{H})$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = -\left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle = -\left\| \boldsymbol{U}\boldsymbol{U}^{\intercal}\boldsymbol{g} \right\|$$

$$\left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{U}\boldsymbol{U}^{\intercal} \right\| \leq \tilde{\epsilon}_3 \quad \textbf{(?)}$$

$$\text{Rank}(\tilde{\boldsymbol{H}}) \neq \text{Rank}(\boldsymbol{H}) \Longrightarrow \left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{U}\boldsymbol{U}^{\intercal} \right\| = 1$$

$$\langle \boldsymbol{Hg}, \boldsymbol{p} \rangle = - \left\langle \boldsymbol{Hg}, \boldsymbol{H}^{\dagger}\boldsymbol{g} \right\rangle = - \left\| \boldsymbol{UU}^{\intercal}\boldsymbol{g} \right\|$$

$$\left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{UU}^{\intercal} \right\| \leq \tilde{\epsilon}_3 \quad \text{✗}$$

$$\text{Rank}(\tilde{\boldsymbol{H}}) \neq \text{Rank}(\boldsymbol{H}) \Longrightarrow \left\| \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\intercal} - \boldsymbol{UU}^{\intercal} \right\| = 1$$

Instead of

$$\left\| \tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}\boldsymbol{H}^{\dagger} \right\| \leq \tilde{\epsilon}_3$$

which implies

$$\left\| \left(\tilde{\boldsymbol{H}} \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H} \boldsymbol{H}^{\dagger} \right) \boldsymbol{v} \right\| \leq \tilde{\epsilon}_3 \left\| \boldsymbol{v} \right\|, \quad \text{for all } \boldsymbol{v}$$

we only need

$$\left\| \left(\tilde{\boldsymbol{H}} \tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H} \boldsymbol{H}^{\dagger} \right) \boldsymbol{g} \right\| \leq \tilde{\epsilon}_3 \left\| \boldsymbol{g} \right\|$$

$$\left\| \left(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^{\dagger} - \boldsymbol{H}\boldsymbol{H}^{\dagger} \right) \boldsymbol{g} \right\| \leq \left(\mathcal{O}(\epsilon) + \sqrt{1-\nu} \right) \|\boldsymbol{g}\|$$

$$\boldsymbol{p}_k^{(t)} \approx \underset{\boldsymbol{p}\in\mathcal{K}_t}{\arg\min} \left\| \tilde{\boldsymbol{H}}_k \boldsymbol{p} + \boldsymbol{g}_k \right\|$$

$$\boldsymbol{p}_k^{(t)} \approx \underset{\boldsymbol{p} \in \mathcal{K}_t}{\arg\min} \left\| \tilde{\boldsymbol{H}}_k \boldsymbol{p} + \boldsymbol{g}_k \right\|$$

$$\left\| \boldsymbol{p}_k^{(t)} \right\| \leq \left(\mathcal{O}(1) + \frac{\sqrt{1-\nu}}{\epsilon} \right) \|\boldsymbol{g}_k\|, \quad t = 1, 2, \ldots, \text{Rank}(\tilde{\boldsymbol{H}}_k)$$

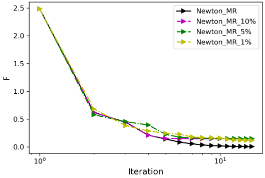
Global Convergence: Inherent Stability

$$\left\|\boldsymbol{g}_{k+1}\right\|^2 \leq \left(1-\eta+\mathcal{O}(\epsilon)\right)\left\|\boldsymbol{g}_k\right\|^2$$

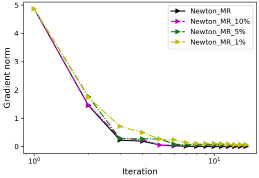
Local Convergence: Inherent Stability

$$\left\|\boldsymbol{g}(\boldsymbol{x}_{k+1})\right\| \leq c_1\left\|\boldsymbol{g}(\boldsymbol{x}_k)\right\|^2 + \left(c_2+\mathcal{O}(\epsilon)\right)\left\|\boldsymbol{g}(\boldsymbol{x}_k)\right\|$$

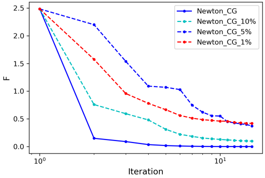
Softmax-Cross Entropy: HAPT



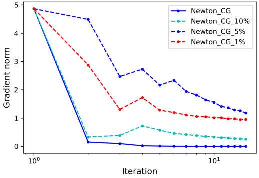
$f(\boldsymbol{x}_k)$ vs. Iterations



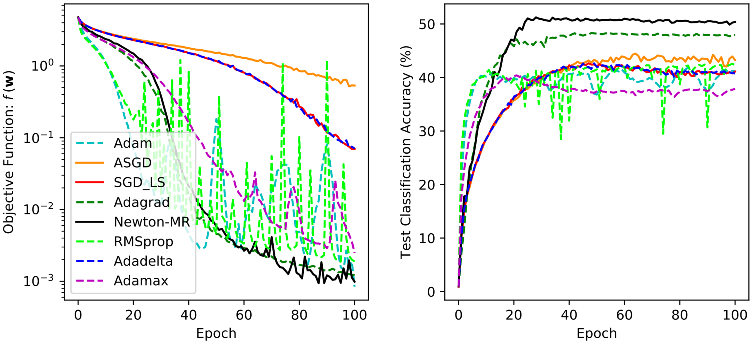
$\|\boldsymbol{g}_k\|$ vs. Iterations



$f(\boldsymbol{x}_k)$ vs. Iterations



$\|\boldsymbol{g}_k\|$ vs. Iterations



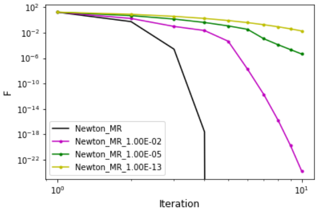
DenseNet-201 with SoftPlus activation and `CIFAR100` dataset

The factors involving $1 - \nu$ have real effect!

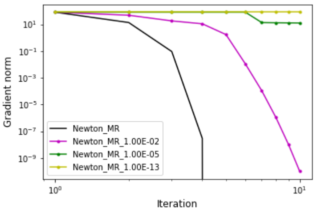
$$f(x_1, x_2) = \frac{ax_1^2}{b - x_2}, \quad x_1 \in \mathbb{R},\ x_2 \in (-\infty, b) \cup (b, \infty)$$

$$f(x_1, x_2) = \frac{a x_1^2}{b - x_2}, \quad x_1 \in \mathbb{R},\ x_2 \in (-\infty, b) \cup (b, \infty)$$

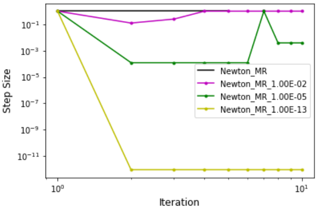
$$\nu = \frac{8}{9}$$



$f(\boldsymbol{x}_k)$ vs. Iterations



$\|\boldsymbol{g}_k\|$ vs. Iterations



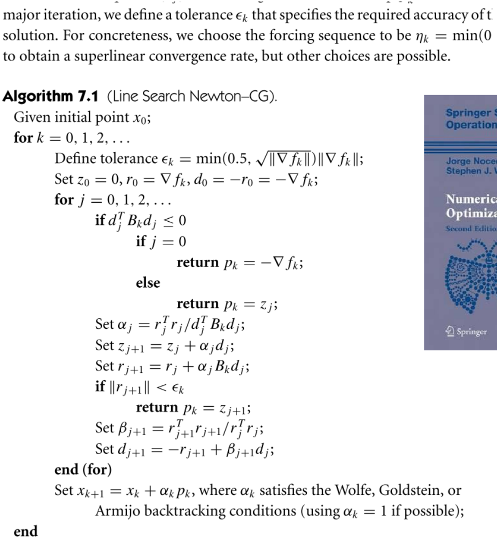
Step-size vs. Iterations

JEFFY, WHERE'S DADDY?
HA HA, YAY!
WHEEEERE'S MOMMY?
YAY!
OKAY, JEFFY –
WHERE'S the global optimum of a non-convex function?

Recall...

Algorithm Generic 2nd-order Method

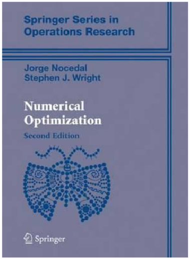
```
Start from $\mathbf{x}_0$
for $k = 1, 2, \ldots$ do
    $\mathbf{p}_k = \begin{cases} \alpha_k \mathbf{p} \quad \text{where} \quad \mathbf{H}_k \mathbf{p} \approx -\mathbf{g}_k & \text{(Line Search)} \\ \underset{\|\mathbf{p}\| \leq \Delta}{\arg\min} \; \langle \mathbf{p}, \mathbf{g}_k \rangle + \langle \mathbf{p}, \mathbf{H}_k \mathbf{p} \rangle / 2 & \text{(Trust Region)} \end{cases}$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$
end for
```

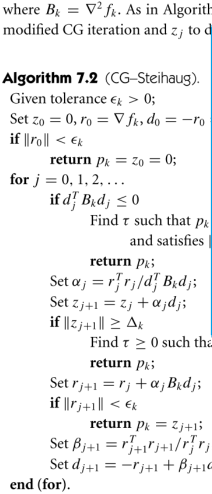


major iteration, we define a tolerance ϵ_k that specifies the required accuracy of t solution. For concreteness, we choose the forcing sequence to be $\eta_k = \min(0$ to obtain a superlinear convergence rate, but other choices are possible.

Algorithm 7.1 (Line Search Newton–CG).

```
Given initial point $x_0$;
for $k = 0, 1, 2, \ldots$
    Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|})\|\nabla f_k\|$;
    Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
    for $j = 0, 1, 2, \ldots$
        if $d_j^T B_k d_j \le 0$
            if $j = 0$
                return $p_k = -\nabla f_k$;
            else
                return $p_k = z_j$;
        Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
        Set $z_{j+1} = z_j + \alpha_j d_j$;
        Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
        if $\|r_{j+1}\| < \epsilon_k$
            return $p_k = z_{j+1}$;
        Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
        Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
    end (for)
    Set $x_{k+1} = x_k + \alpha_k p_k$, where $\alpha_k$ satisfies the Wolfe, Goldstein, or
        Armijo backtracking conditions (using $\alpha_k = 1$ if possible);
end
```

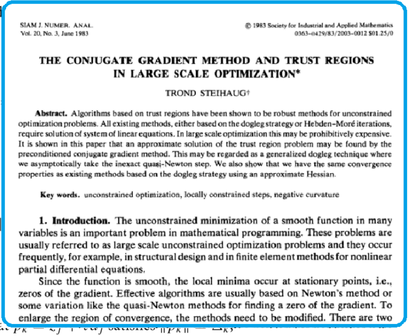




where $B_k = \nabla^2 f_k$. As in Algorithm 7.1, we use d_j to denote the search directic
modified CG iteration and z_j to d

Algorithm 7.2 (CG–Steihaug).

```
Given tolerance $\epsilon_k > 0$;
Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0$
if $\|r_0\| < \epsilon_k$
        return $p_k = z_0 = 0$;
for $j = 0, 1, 2, \ldots$
        if $d_j^T B_k d_j \le 0$
                Find $\tau$ such that $p_k$
                        and satisfies |
                return $p_k$;
        Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
        Set $z_{j+1} = z_j + \alpha_j d_j$;
        if $\|z_{j+1}\| \ge \Delta_k$
                Find $\tau \ge 0$ such that $p_k = z_j + \tau d_j$ satisfies $\|p_k\| = \Delta_k$;
                return $p_k$;
        Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
        if $\|r_{j+1}\| < \epsilon_k$
                return $p_k = z_{j+1}$;
        Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
        Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
end (for).
```



SIAM J. NUMER. ANAL.
Vol. 20, No. 3, June 1983

© 1983 Society for Industrial and Applied Mathematics
0363-0429/83/2003-0012 $01.25/0

THE CONJUGATE GRADIENT METHOD AND TRUST REGIONS IN LARGE SCALE OPTIMIZATION*

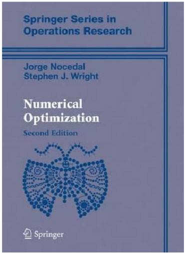
TROND STEIHAUG†

Abstract. Algorithms based on trust regions have been shown to be robust methods for unconstrained optimization problems. All existing methods, either based on the dogleg strategy or Hebden–Moré iterations, require solution of system of linear equations. In large scale optimization this may be prohibitively expensive. It is shown in this paper that an approximate solution of the trust region problem may be found by the preconditioned conjugate gradient method. This may be regarded as a generalized dogleg technique where we asymptotically take the inexact quasi-Newton step. We also show that we have the same convergence properties as existing methods based on the dogleg strategy using an approximate Hessian.

Key words. unconstrained optimization, locally constrained steps, negative curvature

1. Introduction. The unconstrained minimization of a smooth function in many variables is an important problem in mathematical programming. These problems are usually referred to as large scale unconstrained optimization problems and they occur frequently, for example, in structural design and in finite element methods for nonlinear partial differential equations.

Since the function is smooth, the local minima occur at stationary points, i.e., zeros of the gradient. Effective algorithms are usually based on Newton's method or some variation like the quasi-Newton methods for finding a zero of the gradient. To enlarge the region of convergence, the methods need to be modified. There are two



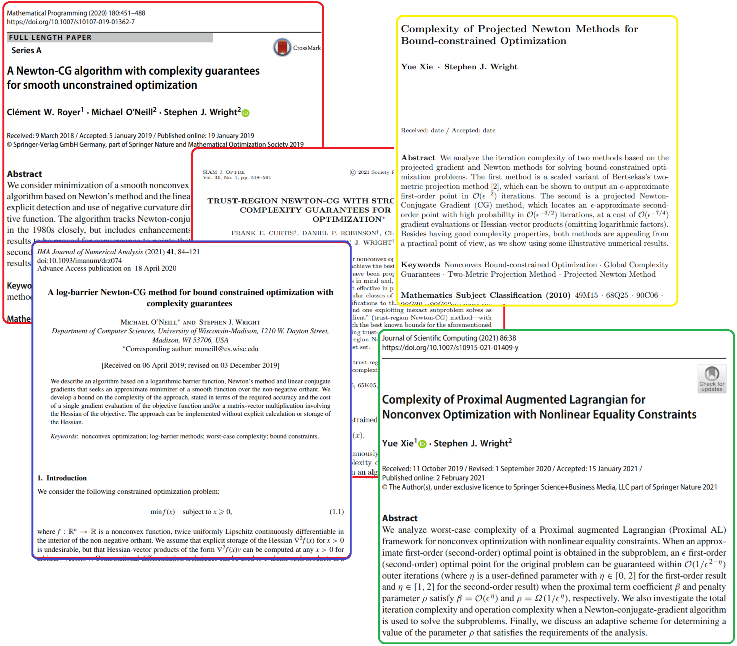
Springer Series in Operations Research

Jorge Nocedal
Stephen J. Wright

Numerical Optimization

Second Edition

Springer



Mathematical Programming (2020) 180:451–488
https://doi.org/10.1007/s10107-019-01362-7

FULL LENGTH PAPER
Series A

CrossMark

A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization

Clément W. Royer[1] · Michael O'Neill[2] · Stephen J. Wright[2]

Received: 9 March 2018 / Accepted: 5 January 2019 / Published online: 19 January 2019
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2019

Abstract

We consider minimization of a smooth nonconvex
algorithm based on Newton's method and the linea
explicit detection and use of negative curvature dir
tive function. The algorithm tracks Newton-conju
in the 1980s closely, but includes enhancements
results
second
results

Keywo
metho

Mathe

SIAM J. OPTIM.
Vol. 31, No. 1, pp. 518–544

© 2021 Society f

TRUST-REGION NEWTON-CG WITH STRC
COMPLEXITY GUARANTEES FOR
OPTIMIZATION*

FRANK E. CURTIS†, DANIEL P. ROBINSON†, CL
J. WRIGHT§

r nonconvex op
ıchieve the best
ıave been prop
s in mind and,
t effective in p
ular classes of
ifications to th
nd one exploiting inex
lient" (trust-region N
ch the best known bou
ing trust-
region Ne
st set.

trust-reg
complexi

5, 65K05,

strained

(x),

nuously
lexity c
n an alg

IMA Journal of Numerical Analysis (2021) **41**, 84–121
doi:10.1093/imanum/drz074
Advance Access publication on 18 April 2020

A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees

Michael O'Neill* and Stephen J. Wright
Department of Computer Sciences, University of Wisconsin-Madison, 1210 W. Dayton Street, Madison, WI 53706, USA
*Corresponding author: moneill@cs.wisc.edu

[Received on 06 April 2019; revised on 03 December 2019]

We describe an algorithm based on a logarithmic barrier function, Newton's method and linear conjugate gradients that seeks an approximate minimizer of a smooth function over the non-negative orthant. We develop a bound on the complexity of the approach, stated in terms of the required accuracy and the cost of a single gradient evaluation of the objective function and/or a matrix-vector multiplication involving the Hessian of the objective. The approach can be implemented without explicit calculation or storage of the Hessian.

Keywords: nonconvex optimization; log-barrier methods; worst-case complexity; bound constraints.

1. Introduction

We consider the following constrained optimization problem:

$$\min f(x) \quad \text{subject to } x \geqslant 0, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonconvex function, twice uniformly Lipschitz continuously differentiable in the interior of the non-negative orthant. We assume that explicit storage of the Hessian $\nabla^2 f(x)$ for $x > 0$ is undesirable, but that Hessian-vector products of the form $\nabla^2 f(x)v$ can be computed at any $x > 0$ for

Complexity of Projected Newton Methods for Bound-constrained Optimization

Yue Xie · Stephen J. Wright

Received: date / Accepted: date

Abstract We analyze the iteration complexity of two methods based on the projected gradient and Newton methods for solving bound-constrained optimization problems. The first method is a scaled variant of Bertsekas's two-metric projection method [2], which can be shown to output an ϵ-approximate first-order point in $\mathcal{O}(\epsilon^{-2})$ iterations. The second is a projected Newton-Conjugate Gradient (CG) method, which locates an ϵ-approximate second-order point with high probability in $\mathcal{O}(\epsilon^{-3/2})$ iterations, at a cost of $\mathcal{O}(\epsilon^{-7/4})$ gradient evaluations or Hessian-vector products (omitting logarithmic factors). Besides having good complexity properties, both methods are appealing from a practical point of view, as we show using some illustrative numerical results.

Keywords Nonconvex Bound-constrained Optimization · Global Complexity Guarantees · Two-Metric Projection Method · Projected Newton Method

Mathematics Subject Classification (2010) 49M15 · 68Q25 · 90C06 ·

Journal of Scientific Computing (2021) 86:38
https://doi.org/10.1007/s10915-021-01409-y

Check for updates

Complexity of Proximal Augmented Lagrangian for Nonconvex Optimization with Nonlinear Equality Constraints

Yue Xie[1] · Stephen J. Wright[2]

Received: 11 October 2019 / Revised: 1 September 2020 / Accepted: 15 January 2021 /
Published online: 2 February 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

We analyze worst-case complexity of a Proximal augmented Lagrangian (Proximal AL) framework for nonconvex optimization with nonlinear equality constraints. When an approximate first-order (second-order) optimal point is obtained in the subproblem, an ϵ first-order (second-order) optimal point for the original problem can be guaranteed within $\mathcal{O}(1/\epsilon^{2-\eta})$ outer iterations (where η is a user-defined parameter with $\eta \in [0, 2]$ for the first-order result and $\eta \in [1, 2]$ for the second-order result) when the proximal term coefficient β and penalty parameter ρ satisfy $\beta = \mathcal{O}(\epsilon^{\eta})$ and $\rho = \Omega(1/\epsilon^{\eta})$, respectively. We also investigate the total iteration complexity and operation complexity when a Newton-conjugate-gradient algorithm is used to solve the subproblems. Finally, we discuss an adaptive scheme for determining a value of the parameter ρ that satisfies the requirements of the analysis.

Conjugate Gradient

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})}{\arg\min} \; \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \; \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \; /2$$

Conjugate Gradient

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t(\boldsymbol{H}, -\boldsymbol{g})}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \ \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \ /2$$

- **Useful for trust region:**

Conjugate Gradient

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p} \in \mathcal{K}_t(\boldsymbol{H}, -\boldsymbol{g})} \quad \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \ \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \ /2$$

- **Useful for trust region:**
  - Similarity to TR's sub-problem: $\mathop{\arg\min}_{\|\boldsymbol{p}\| \leq \Delta} \ \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \ /2$

Conjugate Gradient

$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})} \quad \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \ \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \ /2$$

- **Useful for trust region:**
  - Similarity to TR's sub-problem: $\mathop{\arg\min}_{\|\boldsymbol{p}\|\leq\Delta} \ \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \ /2$
  - $\left\|\boldsymbol{p}^{(t)}\right\|$ increasing with t

Conjugate Gradient

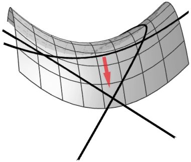
$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t(\boldsymbol{H}, -\boldsymbol{g})}{\arg\min} \quad \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle /2$$

- **Useful for trust region:**
  - Similarity to TR's sub-problem: $\underset{\|\boldsymbol{p}\| \leq \Delta}{\arg\min} \; \langle \boldsymbol{p}, \boldsymbol{g} \rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle /2$
  - $\left\| \boldsymbol{p}^{(t)} \right\|$ increasing with t
- **Useful for Newton-CG and trust-region:**

Conjugate Gradient

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})}{\arg\min} \quad \langle \boldsymbol{p}, \boldsymbol{g}\rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p}\rangle /2$$

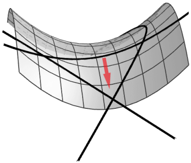
- **Useful for trust region:**
  - Similarity to TR's sub-problem: $\underset{\|\boldsymbol{p}\|\leq\Delta}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{g}\rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p}\rangle /2$
  - $\left\|\boldsymbol{p}^{(t)}\right\|$ increasing with t
- **Useful for Newton-CG and trust-region:**
  - Negative curvature direction



Conjugate Gradient

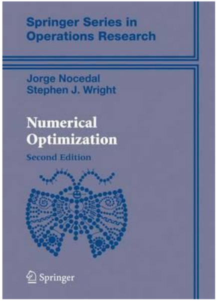
$$\boldsymbol{p}^{(t)} = \mathop{\arg\min}_{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})} \langle \boldsymbol{p}, \boldsymbol{g}\rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p}\rangle/2$$

- **Useful for trust region:**
  - Similarity to TR's sub-problem: $\mathop{\arg\min}_{\|\boldsymbol{p}\|\leq\Delta} \langle \boldsymbol{p}, \boldsymbol{g}\rangle + \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p}\rangle/2$
  - $\left\|\boldsymbol{p}^{(t)}\right\|$ increasing with t
- **Useful for Newton-CG and trust-region:**
  - Negative curvature direction



Algorithm 7.1 (Line Search Newton–CG).

```
Given initial point $x_0$;
for $k = 0, 1, 2, \dots$
    Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|})\|\nabla f_k\|$;
    Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
    for $j = 0, 1, 2, \dots$
        if $d_j^T B_k d_j \le 0$
            if $j = 0$
                return $p_k = -\nabla f_k$;
            else
                return $p_k = z_j$;
        Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
        Set $z_{j+1} = z_j + \alpha_j d_j$;
        Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
        if $\|r_{j+1}\| < \epsilon_k$
            return $p_k = z_{j+1}$;
        Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
        Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
    end (for)
    Set $x_{k+1} = x_k + \alpha_k p_k$, where $\alpha_k$ satisfies the Wolfe, Goldstein, or
        Armijo backtracking conditions (using $\alpha_k = 1$ if possible);
end
```



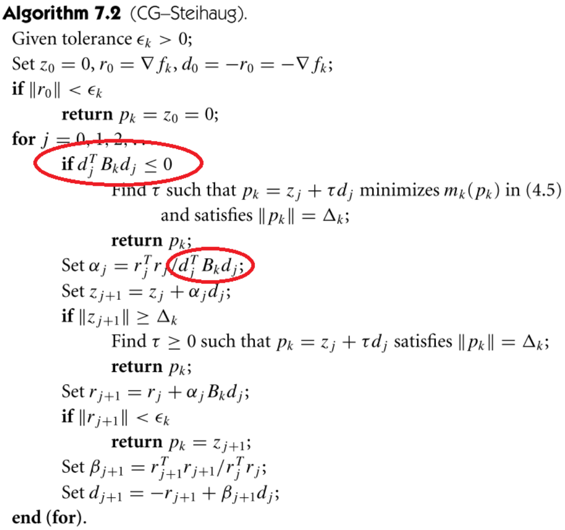
Springer Series in Operations Research

Jorge Nocedal
Stephen J. Wright

Numerical Optimization

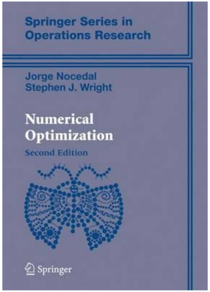
Second Edition

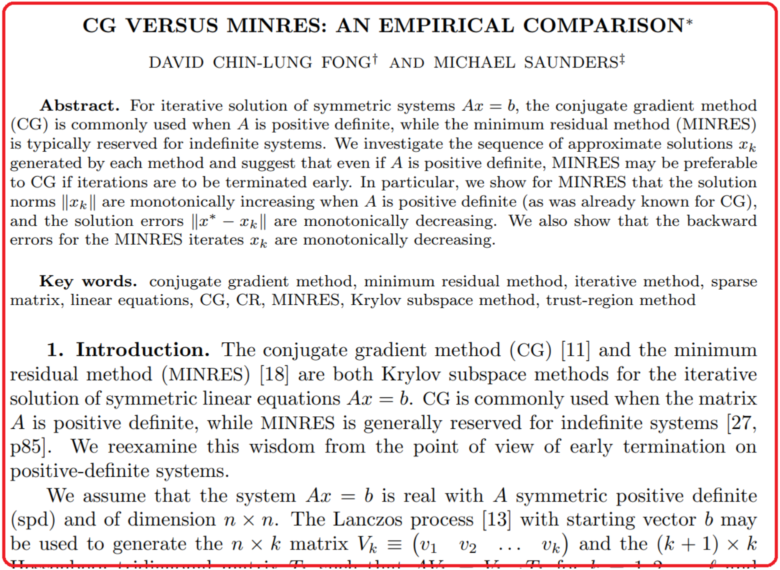
Springer



Algorithm 7.2 (CG–Steihaug).

```
Given tolerance $\epsilon_k > 0$;
Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
if $\|r_0\| < \epsilon_k$
    return $p_k = z_0 = 0$;
for $j = 0, 1, 2, \ldots$
    if $d_j^T B_k d_j \le 0$
        Find $\tau$ such that $p_k = z_j + \tau d_j$ minimizes $m_k(p_k)$ in (4.5)
            and satisfies $\|p_k\| = \Delta_k$;
        return $p_k$;
    Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
    Set $z_{j+1} = z_j + \alpha_j d_j$;
    if $\|z_{j+1}\| \ge \Delta_k$
        Find $\tau \ge 0$ such that $p_k = z_j + \tau d_j$ satisfies $\|p_k\| = \Delta_k$;
        return $p_k$;
    Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
    if $\|r_{j+1}\| < \epsilon_k$
        return $p_k = z_{j+1}$;
    Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
    Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
end (for).
```





CG VERSUS MINRES: AN EMPIRICAL COMPARISON*

DAVID CHIN-LUNG FONG[†] AND MICHAEL SAUNDERS[‡]

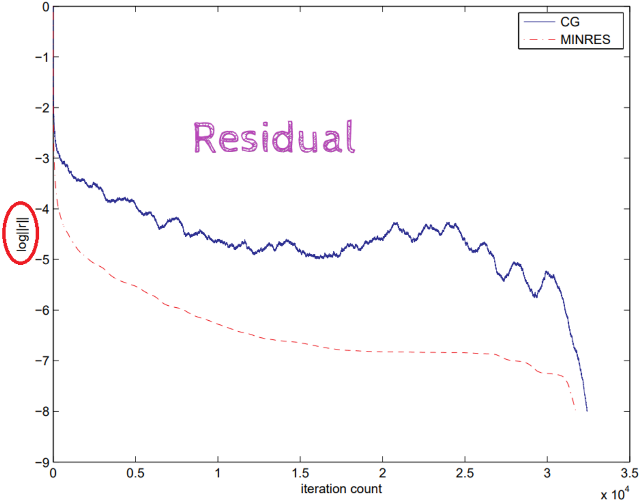
Abstract. For iterative solution of symmetric systems $Ax = b$, the conjugate gradient method (CG) is commonly used when A is positive definite, while the minimum residual method (MINRES) is typically reserved for indefinite systems. We investigate the sequence of approximate solutions x_k generated by each method and suggest that even if A is positive definite, MINRES may be preferable to CG if iterations are to be terminated early. In particular, we show for MINRES that the solution norms $\|x_k\|$ are monotonically increasing when A is positive definite (as was already known for CG), and the solution errors $\|x^* - x_k\|$ are monotonically decreasing. We also show that the backward errors for the MINRES iterates x_k are monotonically decreasing.

Key words. conjugate gradient method, minimum residual method, iterative method, sparse matrix, linear equations, CG, CR, MINRES, Krylov subspace method, trust-region method

1. Introduction.

The conjugate gradient method (CG) [11] and the minimum residual method (MINRES) [18] are both Krylov subspace methods for the iterative solution of symmetric linear equations $Ax = b$. CG is commonly used when the matrix A is positive definite, while MINRES is generally reserved for indefinite systems [27, p85]. We reexamine this wisdom from the point of view of early termination on positive-definite systems.

We assume that the system $Ax = b$ is real with A symmetric positive definite (spd) and of dimension $n \times n$. The Lanczos process [13] with starting vector b may be used to generate the $n \times k$ matrix $V_k \equiv \begin{pmatrix} v_1 & v_2 & \ldots & v_k \end{pmatrix}$ and the $(k+1) \times k$ Hessenberg tridiagonal matrix $\underline{T}_k$ such that $AV_k = V_{k+1}\underline{T}_k$ for $k = 1, 2, \ldots, \ell$ and



Fong, D.C., & Saunders, M. (2012). CG Versus MINRES: An Empirical Comparison. Sultan Qaboos University Journal for Science, 17, 44-62.

Minimum Residual

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})}{\arg\min} \left\|\boldsymbol{g} + \boldsymbol{H}\boldsymbol{p}\right\|^2/2$$

Minimum Residual

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p}\in\mathcal{K}_t(\boldsymbol{H},-\boldsymbol{g})}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{g}\rangle + \left\langle \boldsymbol{p}, \boldsymbol{H}^2\boldsymbol{p}\right\rangle /2$$

Minimum Residual

$$\boldsymbol{p}^{(t)} = \underset{\boldsymbol{p} \in \mathcal{K}_t(\boldsymbol{H}, -\boldsymbol{g})}{\arg\min} \ \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{g} \rangle + \underbrace{\left\langle \boldsymbol{p}, \boldsymbol{H}^2 \boldsymbol{p} \right\rangle}_{\text{☹}} /2$$

MINRES:

MINRES:

- **Negative Curvature** or **PSD Certificate**? ($\underline{\textbf{without any additional work}}$)

MINRES:

- **Negative Curvature** or **PSD Certificate**? (**without any additional work**)
- **Monotonicity Properties**?

$$\boldsymbol{A} \longleftarrow \boldsymbol{H}, \quad \boldsymbol{b} \longleftarrow -\boldsymbol{g}$$

Starting from $\boldsymbol{x}_0 = \boldsymbol{0}$, we have

$$\boldsymbol{x}_t = \underset{\boldsymbol{x} \in \mathcal{K}_t(\boldsymbol{A},\boldsymbol{b})}{\arg\min} \; \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$$

Lemma (Liu and Roosta, 2021)

As part of MINRES iterations, we can readily compute

$$\frac{\langle \boldsymbol{r}_{t-1}, \boldsymbol{A}\boldsymbol{r}_{t-1}\rangle}{\langle \boldsymbol{r}_{t-1}, \boldsymbol{r}_{t-1}\rangle} = \spadesuit_{t-1} \times \clubsuit_{t}$$

where $\boldsymbol{r}_{t-1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{t-1}$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^{\intercal} \end{bmatrix}$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^\intercal \end{bmatrix} \implies \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right)$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^\intercal \end{bmatrix} \Longrightarrow \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$\text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right)$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right)$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^{\intercal} \end{bmatrix} \Longrightarrow \boldsymbol{V}_t^{\intercal}\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right) \Longrightarrow \boldsymbol{r}_{t-1} = \boldsymbol{V}_t\boldsymbol{z}$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^{\intercal} \end{bmatrix} \implies \boldsymbol{V}_t^{\intercal}\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\begin{aligned} \text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b}) &\implies \boldsymbol{r}_{t-1} = \boldsymbol{V}_t\boldsymbol{z} \\ &\implies \boldsymbol{r}_{t-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{t-1} \end{aligned}$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^{\intercal} \end{bmatrix} \implies \boldsymbol{V}_t^{\intercal}\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\begin{aligned} \text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b}) &\implies \boldsymbol{r}_{t-1} = \boldsymbol{V}_t\boldsymbol{z} \\ &\implies \boldsymbol{r}_{t-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{t-1} = \boldsymbol{z}^{\intercal}\boldsymbol{V}_t^{\intercal}\boldsymbol{A}\boldsymbol{V}_t\boldsymbol{z} \end{aligned}$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right)$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^\intercal \end{bmatrix} \implies \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\begin{aligned} \text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t\left(\boldsymbol{A}, \boldsymbol{b}\right) &\implies \boldsymbol{r}_{t-1} = \boldsymbol{V}_t \boldsymbol{z} \\ &\implies \boldsymbol{r}_{t-1}^\intercal \boldsymbol{A}\boldsymbol{r}_{t-1} = \boldsymbol{z}^\intercal \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t \boldsymbol{z} = \boldsymbol{z}^\intercal \boldsymbol{T}_t \boldsymbol{z} \end{aligned}$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^\intercal \end{bmatrix} \implies \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\begin{aligned} \text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b}) &\implies \boldsymbol{r}_{t-1} = \boldsymbol{V}_t \boldsymbol{z} \\ &\implies \boldsymbol{r}_{t-1}^\intercal \boldsymbol{A} \boldsymbol{r}_{t-1} = \boldsymbol{z}^\intercal \boldsymbol{V}_t^\intercal \boldsymbol{A} \boldsymbol{V}_t \boldsymbol{z} = \boldsymbol{z}^\intercal \boldsymbol{T}_t \boldsymbol{z} \end{aligned}$$

$$\boldsymbol{T}_t \succ \boldsymbol{0} \implies \boldsymbol{r}_{i-1}^\intercal \boldsymbol{A} \boldsymbol{r}_{i-1} > 0, \; 1 \leq i \leq t$$

Since $\boldsymbol{r}_{t-1} \in \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b})$, from Lanczos Process, we get

$$\boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{V}_{t+1}\widetilde{\boldsymbol{T}}_t, \quad \widetilde{\boldsymbol{T}}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \beta_{t+1}\boldsymbol{e}_t^\intercal \end{bmatrix} \implies \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t = \boldsymbol{T}_t$$

$$\begin{aligned} \text{Range}(\boldsymbol{V}_t) = \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{b}) &\implies \boldsymbol{r}_{t-1} = \boldsymbol{V}_t \boldsymbol{z} \\ &\implies \boldsymbol{r}_{t-1}^\intercal \boldsymbol{A}\boldsymbol{r}_{t-1} = \boldsymbol{z}^\intercal \boldsymbol{V}_t^\intercal \boldsymbol{A}\boldsymbol{V}_t \boldsymbol{z} = \boldsymbol{z}^\intercal \boldsymbol{T}_t \boldsymbol{z} \end{aligned}$$

$$\boldsymbol{T}_t \succ \boldsymbol{0} \implies \boldsymbol{r}_{i-1}^\intercal \boldsymbol{A}\boldsymbol{r}_{i-1} > 0, \; 1 \leq i \leq t$$

How about the converse?

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$\boldsymbol{r}_{i-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{i-1} > 0,\ 1 \leq i \leq t \implies \begin{cases} \boldsymbol{T}_i \succ \mathbf{0}, & 1 \leq i \leq t \\ \\ \\ \\ \\ \end{cases}$$

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$\boldsymbol{r}_{i-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{i-1} > 0,\ 1 \leq i \leq t \implies \begin{cases} \boldsymbol{T}_i \succ \mathbf{0}, & 1 \leq i \leq t \\ \\ \boldsymbol{A} \succeq \mathbf{0}, & \text{if} \quad t = g(\boldsymbol{A}, \boldsymbol{b}) \geq \text{Rank}(\boldsymbol{A}) \\ \\ \\ \\ \\ \\ \end{cases}$$

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$\mathbf{r}_{i-1}^{\intercal}\mathbf{A}\mathbf{r}_{i-1} > 0,\ 1 \leq i \leq t \implies \begin{cases} \mathbf{T}_i \succ \mathbf{0}, & 1 \leq i \leq t \\ \mathbf{A} \succeq \mathbf{0}, & \text{if} \quad t = g(\mathbf{A}, \mathbf{b}) \geq \text{Rank}(\mathbf{A}) \end{cases}$$

E.g.: Picking $\mathbf{b}$ uniformly at random from unit sphere guarantees w.p.1 that $t = g(\mathbf{A}, \mathbf{b}) \geq \text{Rank}(\mathbf{A})$

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$\boldsymbol{r}_{i-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{i-1} > 0,\ 1 \leq i \leq t \implies \begin{cases} \boldsymbol{T}_i \succ \boldsymbol{0}, \quad 1 \leq i \leq t \\ \\ \boldsymbol{A} \succeq \boldsymbol{0}, \quad \text{if} \quad t = g(\boldsymbol{A}, \boldsymbol{b}) \geq \text{Rank}(\boldsymbol{A}) \\ \\ \boldsymbol{x}_i^{\intercal}\boldsymbol{b} > \boldsymbol{x}_i^{\intercal}\boldsymbol{A}\boldsymbol{x}_t, \quad 1 \leq i \leq t \\ \\ \\ \\ \\ \end{cases}$$

E.g.: Picking $\boldsymbol{b}$ uniformly at random from unit sphere guarantees w.p.1 that $t = g(\boldsymbol{A}, \boldsymbol{b}) \geq \text{Rank}(\boldsymbol{A})$

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$\boldsymbol{r}_{i-1}^\top \boldsymbol{A}\boldsymbol{r}_{i-1} > 0,\ 1 \le i \le t \implies \begin{cases} \boldsymbol{T}_i \succ \boldsymbol{0}, & 1 \le i \le t \\ \boldsymbol{A} \succeq \boldsymbol{0}, & \text{if} \quad t = g(\boldsymbol{A}, \boldsymbol{b}) \ge \text{Rank}(\boldsymbol{A}) \\ \boldsymbol{x}_i^\top \boldsymbol{b} > \boldsymbol{x}_i^\top \boldsymbol{A}\boldsymbol{x}_t, & 1 \le i \le t \\ \langle \boldsymbol{x}_i, \boldsymbol{A}\boldsymbol{x}_i \rangle /2 - \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle \downarrow, & 1 \le i \le t \end{cases}$$

E.g.: Picking $\boldsymbol{b}$ uniformly at random from unit sphere guarantees w.p.1 that $t = g(\boldsymbol{A}, \boldsymbol{b}) \ge \text{Rank}(\boldsymbol{A})$

MINRES: Non-positive Curvature Detection and Monotonicity Properties

$$
\boldsymbol{r}_{i-1}^{\intercal}\boldsymbol{A}\boldsymbol{r}_{i-1} > 0,\ 1 \leq i \leq t \implies \begin{cases} \boldsymbol{T}_i \succ \boldsymbol{0}, & 1 \leq i \leq t \\ \boldsymbol{A} \succeq \boldsymbol{0}, & \text{if} \quad t = g(\boldsymbol{A}, \boldsymbol{b}) \geq \text{Rank}(\boldsymbol{A}) \\ \boldsymbol{x}_i^{\intercal}\boldsymbol{b} > \boldsymbol{x}_i^{\intercal}\boldsymbol{A}\boldsymbol{x}_t, & 1 \leq i \leq t \\ \langle \boldsymbol{x}_i, \boldsymbol{A}\boldsymbol{x}_i \rangle /2 - \langle \boldsymbol{b}, \boldsymbol{x}_i \rangle \ \downarrow, & 1 \leq i \leq t \\ \|\boldsymbol{x}_i\| \ \uparrow, & 1 \leq i \leq t \end{cases}
$$

E.g.: Picking $\boldsymbol{b}$ uniformly at random from unit sphere guarantees w.p.1 that $t = g(\boldsymbol{A}, \boldsymbol{b}) \geq \text{Rank}(\boldsymbol{A})$

Approximate Optimality Conditions:

Approximate Optimality Conditions:

- **First-order**

$$\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$$

Approximate Optimality Conditions:

- **First-order**

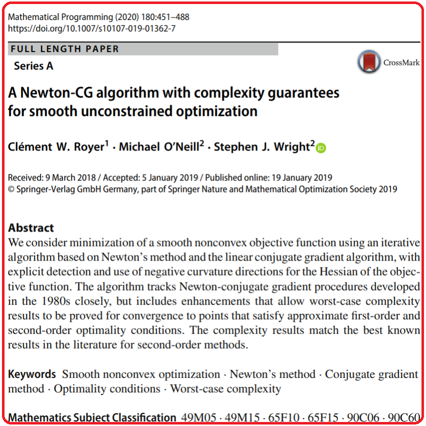
$$\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$$

- **Second-order**

$$\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}, \quad \text{and} \quad \lambda_{\min}\left(\boldsymbol{H}_k\right) \geq -\epsilon_{\boldsymbol{H}}\boldsymbol{I}$$

We use the perturbation approach by Royer, O'Neill, and Wright, 2020

$$\boldsymbol{H} \Longleftarrow \boldsymbol{H} + \epsilon_{\boldsymbol{H}} \boldsymbol{I}$$



Mathematical Programming (2020) 180:451–488
https://doi.org/10.1007/s10107-019-01362-7

FULL LENGTH PAPER

Series A

CrossMark

A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization

Clément W. Royer[1] · Michael O'Neill[2] · Stephen J. Wright[2]

Received: 9 March 2018 / Accepted: 5 January 2019 / Published online: 19 January 2019

Abstract
We consider minimization of a smooth nonconvex objective function using an iterative algorithm based on Newton's method and the linear conjugate gradient algorithm, with explicit detection and use of negative curvature directions for the Hessian of the objective function. The algorithm tracks Newton-conjugate gradient procedures developed in the 1980s closely, but includes enhancements that allow worst-case complexity results to be proved for convergence to points that satisfy approximate first-order and second-order optimality conditions. The complexity results match the best known results in the literature for second-order methods.

Keywords Smooth nonconvex optimization · Newton's method · Conjugate gradient method · Optimality conditions · Worst-case complexity

Mathematics Subject Classification 49M05 · 49M15 · 65F10 · 65F15 · 90C06 · 90C60

```
Algorithm Newton-MR (Non-convex)
for $k = 1, 2, \ldots$ do
end for
```

Algorithm Newton-MR (Non-convex)

```
for $k = 1, 2, \dots$ do
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ then
end for
```

Algorithm Newton-MR (Non-convex)

```
for $k = 1, 2, \dots$ do
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ then
        $\boldsymbol{g}_k \sim \mathcal{B}(\mathbf{0}, 1)$
    end if
end for
```

Algorithm Newton-MR (Non-convex)

```
for $k = 1, 2, \ldots$ do
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ then
        $\boldsymbol{g}_k \sim \mathcal{B}(\mathbf{0}, 1)$
    end if
    Run MINRES to obtain $\boldsymbol{p}_k \approx \arg\min_{\boldsymbol{p} \in \mathbb{R}^d} \|(\boldsymbol{H}_k + \epsilon_{\boldsymbol{H}} \boldsymbol{I})\, \boldsymbol{p} + \boldsymbol{g}_k\|$
end for
```

Algorithm Newton-MR (Non-convex)

```
for $k = 1, 2, \ldots$ do
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ then
        $\boldsymbol{g}_k \sim \mathcal{B}(\mathbf{0}, 1)$
    end if
    Run MINRES to obtain $\boldsymbol{p}_k \approx \arg\min_{\boldsymbol{p} \in \mathbb{R}^d} \|(\boldsymbol{H}_k + \epsilon_{\boldsymbol{H}} \boldsymbol{I})\, \boldsymbol{p} + \boldsymbol{g}_k\|$
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ and MINRES certifies $\boldsymbol{H}_k \geq -\epsilon_{\boldsymbol{H}} \boldsymbol{I}$ then
        Terminate
    end if
end for
```

Algorithm Newton-MR (Non-convex)

```
for  $k = 1, 2, \dots$ do
    if  $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$  then
        $\boldsymbol{g}_k \sim \mathcal{B}(\boldsymbol{0}, 1)$
    end if
    Run MINRES to obtain $\boldsymbol{p}_k \approx \underset{\boldsymbol{p} \in \mathbb{R}^d}{\arg\min} \|(\boldsymbol{H}_k + \epsilon_{\boldsymbol{H}} \boldsymbol{I}) \boldsymbol{p} + \boldsymbol{g}_k\|$
    if  $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ and MINRES certifies $\boldsymbol{H}_k \geq -\epsilon_{\boldsymbol{H}} \boldsymbol{I}$  then
        Terminate
    end if
    Find $\alpha_k$, with the initial trial step-size $\alpha_k = 2$, such that
        $f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f(\boldsymbol{x}_k) - \rho \alpha_k^3 \|\boldsymbol{p}_k\|^3$
end for
```

Algorithm Newton-MR (Non-convex)

```
for $k = 1, 2, \ldots$ do
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ then
        $\boldsymbol{g}_k \sim \mathcal{B}(\mathbf{0}, 1)$
    end if
    Run MINRES to obtain $\boldsymbol{p}_k \approx \underset{\boldsymbol{p} \in \mathbb{R}^d}{\arg\min} \|(\boldsymbol{H}_k + \epsilon_{\boldsymbol{H}} \boldsymbol{I})\, \boldsymbol{p} + \boldsymbol{g}_k\|$
    if $\|\boldsymbol{g}_k\| \leq \epsilon_{\boldsymbol{g}}$ and MINRES certifies $\boldsymbol{H}_k \succeq -\epsilon_{\boldsymbol{H}} \boldsymbol{I}$ then
        Terminate
    end if
    Find $\alpha_k$, with the initial trial step-size $\alpha_k = 2$, such that

        $$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f(\boldsymbol{x}_k) - \rho \alpha_k^3 \|\boldsymbol{p}_k\|^3$$

    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
end for
```

$$\overbrace{\langle \boldsymbol{p}, \boldsymbol{g} \rangle \leq - \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \,/2}^{\text{CG}} \qquad \overbrace{\langle \boldsymbol{p}, \boldsymbol{g} \rangle \leq - \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle}^{\text{MINRES}}$$

$$\overbrace{\langle \boldsymbol{p}, \boldsymbol{g} \rangle \leq - \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle \, /2}^{\text{CG}}$$

$$\Big\downarrow$$

$$\alpha = 1$$

$$\overbrace{\langle \boldsymbol{p}, \boldsymbol{g} \rangle \leq - \langle \boldsymbol{p}, \boldsymbol{H}\boldsymbol{p} \rangle}^{\text{MINRES}}$$

$$\Big\downarrow$$

$$\alpha = 2$$

Operation Complexity

Operation Complexity

- **First-order**

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon_{\boldsymbol{g}}^{-3}\epsilon_{\boldsymbol{H}}^{5/2}, \epsilon_{\boldsymbol{H}}^{-7/2}\right\}\right)$$

Operation Complexity

- **First-order**

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon_{\boldsymbol{g}}^{-3}\epsilon_{\boldsymbol{H}}^{5/2}, \epsilon_{\boldsymbol{H}}^{-7/2}\right\}\right) \overset{\epsilon_{\boldsymbol{H}}^2=\epsilon_{\boldsymbol{g}}=\epsilon}{\Longrightarrow} \tilde{\mathcal{O}}\left(\epsilon^{-7/4}\right)$$

Operation Complexity

- **First-order**

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon_{\boldsymbol{g}}^{-3}\epsilon_{\boldsymbol{H}}^{5/2}, \epsilon_{\boldsymbol{H}}^{-7/2}\right\}\right) \overset{\epsilon_{\boldsymbol{H}}^{2}=\epsilon_{\boldsymbol{g}}=\epsilon}{\Longrightarrow} \tilde{\mathcal{O}}\left(\epsilon^{-7/4}\right)$$

- **Second-order**

$$\tilde{\mathcal{O}}\left(\max\{\epsilon_{\boldsymbol{H}}^{-7/2}, \epsilon_{\boldsymbol{H}}^{-1/2}\epsilon_{\boldsymbol{g}}^{-3/2}\}\right)$$

Operation Complexity

- **First-order**

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon_{\boldsymbol{g}}^{-3}\epsilon_{\boldsymbol{H}}^{5/2}, \epsilon_{\boldsymbol{H}}^{-7/2}\right\}\right) \xRightarrow{\epsilon_{\boldsymbol{H}}^{2}=\epsilon_{\boldsymbol{g}}=\epsilon} \tilde{\mathcal{O}}\left(\epsilon^{-7/4}\right)$$

- **Second-order**

$$\tilde{\mathcal{O}}\left(\max\{\epsilon_{\boldsymbol{H}}^{-7/2}, \epsilon_{\boldsymbol{H}}^{-1/2}\epsilon_{\boldsymbol{g}}^{-3/2}\}\right) \xRightarrow{\epsilon_{\boldsymbol{H}}^{2}=\epsilon_{\boldsymbol{g}}=\epsilon} \tilde{\mathcal{O}}\left(\epsilon^{-7/4}\right)$$

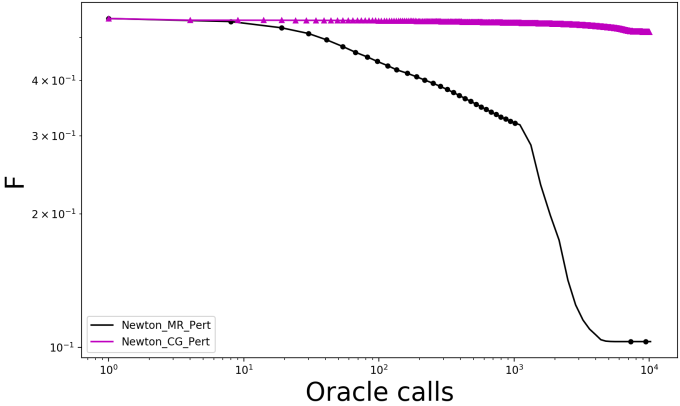


Figure: None-linear Least-square problem with CIFAR10 dataset.

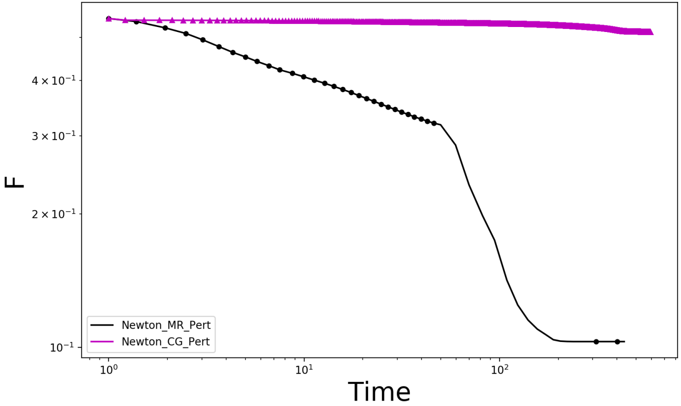


Figure: None-linear Least-square problem with CIFAR10 dataset.

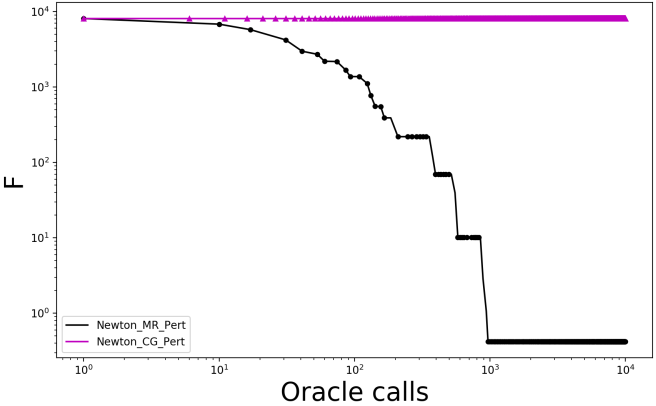


Figure: Auto-encoder with CIFAR10 dataset.

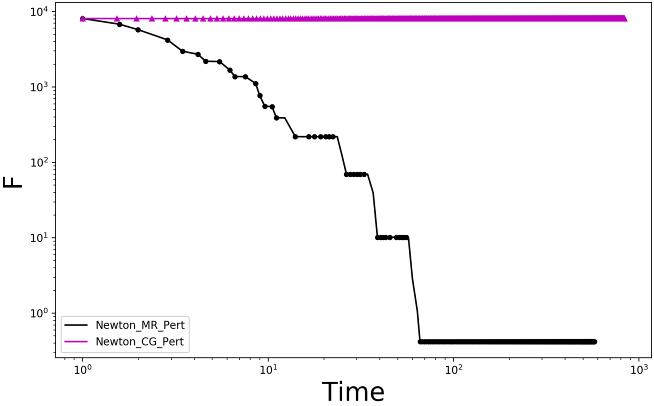


Figure: Auto-encoder with CIFAR10 dataset.

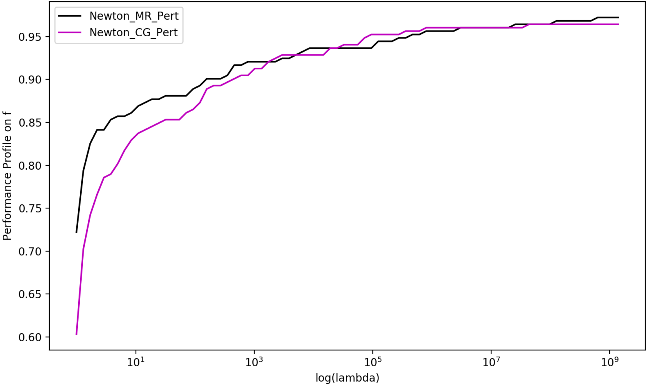


Figure: Performance Profile on 252 CUTEst Problems

THANK
YOU

Roosta, Fred et al. (2018). "Newton-MR: Inexact Newton Method With Minimum Residual Sub-problem Solver". In: *arXiv preprint arXiv:1810.00303*.

Royer, Clément W, Michael O'Neill, and Stephen J Wright (2020). "A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization". In: *Mathematical Programming* 180.1, pp. 451–488.

Liu, Yang and Fred Roosta (2021a). "A Newton-MR Algorithm With Complexity Guarantee Non-convex Optimization". In: In preparation.

— (2021b). "Convergence of Newton-MR under Inexact Hessian Information". In: *SIAM Journal on Optimization* 31.1, pp. 59–90.

— (2021c). "MINRES: From Negative Curvature Detection to Monotonicity Properties". In: Submitted.