

Inhaltsübersicht für heute:

Das Konjugierte-Gradienten-Verfahren

Inexakte Newton-Verfahren

Nichtlineare kleinste Quadrate

Das Gauss-Newton-Verfahren

Inhaltsübersicht für heute:

Das Konjugierte-Gradienten-Verfahren

Inexakte Newton-Verfahren

Nichtlineare kleinste Quadrate

Das Gauss-Newton-Verfahren

Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

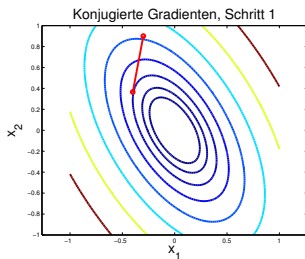
Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.



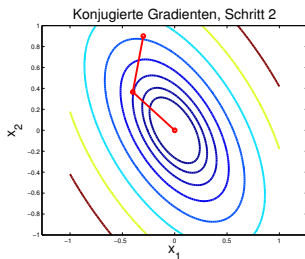
Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.



Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

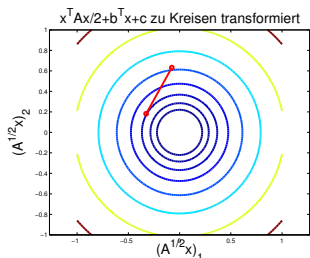
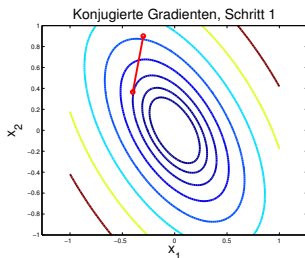
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen **konjugiert** bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.

Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.



Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T A x - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

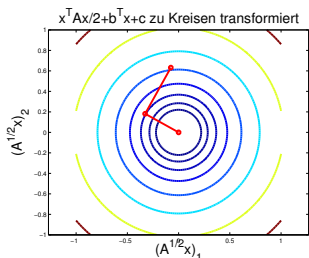
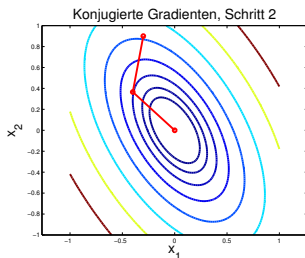
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen **konjugiert** bzgl. $A \succ 0$, falls $d^T A h = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.

Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.



Konsequenz:
Das Optimum ist
nach höchstens
 n Schritten
gefunden.

[Nur theoretisch!]

Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T A x - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

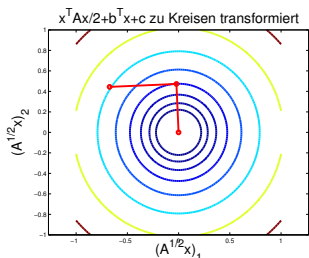
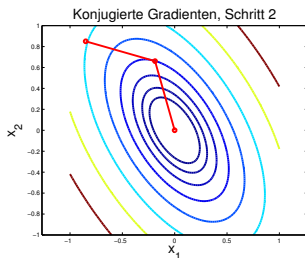
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen **konjugiert** bzgl. $A \succ 0$, falls $d^T A h = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.

Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.



Konsequenz:
Das Optimum ist
nach höchstens
 n Schritten
gefunden.

[Nur theoretisch!]

Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T A x - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

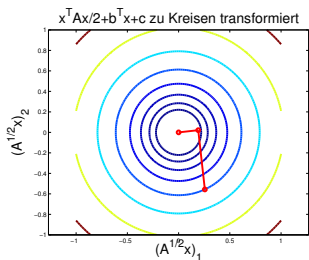
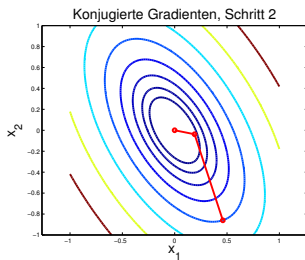
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen **konjugiert** bzgl. $A \succ 0$, falls $d^T A h = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.

Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.



Konsequenz:
Das Optimum ist
nach höchstens
 n Schritten
gefunden.
[Nur theoretisch!]

Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T A x - b^T x + c$ mit $A \succ 0$.

In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

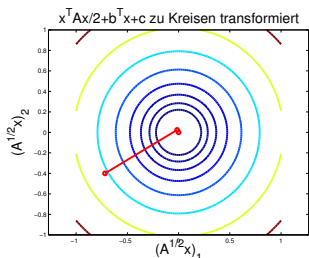
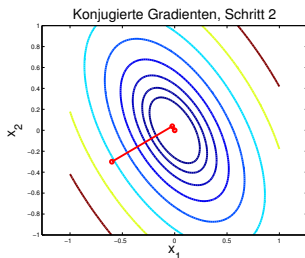
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen \rightarrow konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen **konjugiert** bzgl. $A \succ 0$, falls $d^T A h = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.

Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.



Konsequenz:
Das Optimum ist
nach höchstens
 n Schritten
gefunden.

[Nur theoretisch!]

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
 Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.

Aus exaktem Line-Search folgt $(r^{(k+1)})^T h^{(k)} = 0$

und für jedes $0 \leq i < k$, wegen $x^{(k+1)} = x^{(i+1)} + \sum_{j=i+1}^k \alpha_j h^{(j)}$,

$$(h^{(i)})^T (r^{(k+1)}) = \underbrace{(h^{(i)})^T [Ax^{(i+1)} - b]}_{=(h^{(i)})^T r^{(i+1)}=0} + \sum_{j=i+1}^k \alpha_j \underbrace{(h^{(i)})^T A h^{(j)}}_{=0} = 0$$

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T Ah^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k Ah^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
 Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T Ah^{(k)}}{(h^{(k)})^T Ah^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T Ah^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.

Wähle β_{k+1} so, dass $h^{(k+1)}$ konjugiert zu $h^{(k)}$: $0 = (h^{(k+1)})^T Ah^{(k)}$, also
 $0 = -(r^{(k+1)})^T Ah^{(k)} + \beta_{k+1} (h^{(k)})^T Ah^{(k)} \Rightarrow \beta_k = \frac{(r^{(k+1)})^T Ah^{(k)}}{(h^{(k)})^T Ah^{(k)}}$.

Für jedes $0 \leq i \leq k$ gilt (mit $\beta_0 = 0$) wegen $r^{(i)} = \beta_i h^{(i-1)} - h^{(i)}$
 $(r^{(i)})^T r^{(k+1)} = \beta_i (h^{(i-1)})^T r^{(k+1)} - (h^{(i)})^T r^{(k+1)} = 0$

und jedes $0 \leq i < k$ wegen $Ah^{(i)} = (r^{(i+1)} - r^{(i)})/\alpha_i$

$$(h^{(k+1)})^T Ah^{(i)} = -(r^{(k+1)})^T Ah^{(i)} + \beta_{k+1} \underbrace{(h^{(k)})^T Ah^{(i)}}_{=0} = -\frac{(r^{(k+1)})^T (r^{(i+1)} - r^{(i)})}{\alpha_i} = 0$$

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
 1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
 2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T Ah^{(k)}}$
 3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
 4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k Ah^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
 5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T Ah^{(k)}}{(h^{(k)})^T Ah^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T Ah^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.
 6. $k \leftarrow k + 1$, gehe zu 1.
-

Die Schritte des linearen CG-Verfahrens

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T Ah^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k Ah^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T Ah^{(k)}}{(h^{(k)})^T Ah^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T Ah^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.
6. $k \leftarrow k + 1$, gehe zu 1.

Satz (Lineares CG-Verfahren)

Ist in Iteration k der Gradient $r_k \neq 0$, so gilt

$$\operatorname{span}\{r^{(0)}, \dots, A^k r^{(0)}\} = \operatorname{span}\{r^{(0)}, \dots, r^{(k)}\} = \operatorname{span}\{h^{(0)}, \dots, h^{(k)}\}$$

und $(r^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T Ah^{(k)} = 0$ für $0 \leq i < k$.

Insbesondere ist $x^{(k)} = \operatorname{argmin} \{f(x) : x \in x^{(0)} + \operatorname{span}\{r^{(0)}, \dots, A^{k-1}r^{(0)}\}\}$

und nach höchstens n Iterationen endet das Verfahren mit $r^{(k)} = 0$.

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
[Besonders effizient für dünn besetzte Matrix A oder Hessian-free]

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
[Besonders effizient für dünn besetzte Matrix A oder Hessian-free]
- Stellenauslöschung verursacht numerische Probleme, $k \leq n$ reicht nicht.
→ In der Praxis ist nur mit Prädiktionierung gute Genauigkeit erzielbar.
[Prädikond. \approx Koordinatentrafo Richtung Kreis, stark problemabhängig]

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß \rightarrow Korrekturversuche über Wahl von β :

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß \rightarrow Korrekturversuche über Wahl von β :

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. \rightarrow Abstiegsrichtung

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß \rightarrow Korrekturversuche über Wahl von β :

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. \rightarrow Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß \rightarrow Korrekturversuche über Wahl von β :

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. \rightarrow Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k -Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k + 1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß \rightarrow Korrekturversuche über Wahl von β :

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. \rightarrow Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz
- Regelm. Neustart oder $\beta_k = 0$ ist jeweils ein steilster Abstiegschritt \rightarrow global konvergent

Inhaltsübersicht für heute:

Das Konjugierte-Gradienten-Verfahren

Inexakte Newton-Verfahren

Nichtlineare kleinste Quadrate

Das Gauss-Newton-Verfahren

Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

$$(INC) \quad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

und geben dafür eine **forcing sequence** $\eta_k \in (0, 1)$ vor.

Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

$$(INC) \quad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

und geben dafür eine **forcing sequence** $\eta_k \in (0, 1)$ vor.

Satz (Inexakte Newton-Verfahren)

Sei f hinr. glatt und x^ erfülle die hinr. Opt.-Bed. Ist $x^{(0)}$ nahe genug bei x^* , so gilt für jede Punktfolge $x^{(k+1)} = x^{(k)} + h^{(k)}$ mit $h^{(k)}$ erfüllt (INC):*

Falls $\eta_k \leq \eta < 1$, konvergiert $x^{(k)}$ linear gegen x^ .*

Falls $\eta_k \rightarrow 0$, konvergiert $x^{(k)}$ superlinear gegen x^ .*

Falls $\eta_k \leq \gamma \|\nabla f_k\|$ mit festem $\gamma > 0$, konvergieren die $x^{(k)}$ quadratisch.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
 - a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \neq 0$]
Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.

1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf

$\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald

a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \neq 0$]

Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.

b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\} \|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
 - a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \neq 0$]
Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
 - b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\} \|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, [„Newton“-Schritt]
sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

- Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
 - Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
 - $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \neq 0$]
Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
 - $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\} \|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
 - Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, [„Newton“-Schritt]
sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
 - $k \leftarrow k + 1$, GOTO 1.
-

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
 $p^{(i)}$ minimiert $m(p) = \frac{1}{2} p^T \nabla^2 f_k p + \nabla f_k^T p$ über $\text{span}\{d^{(0)}, \dots, d^{(i-1)}\}$.
 Weil $p^{(i)} = \sum_{j=0}^{i-1} \xi_j d^{(j)}$, $(d^{(j)})^T \nabla^2 f_k d^{(j)} > 0$ und $(d^{(j)})^T \nabla^2 f_k d^{(l)} = 0$ ($j \neq l$)
 folgt $0 = m(0) \geq m(p^{(i)}) = \nabla f_k^T p^{(i)} + \frac{1}{2} \sum_{j=0}^{i-1} \xi_j^2 (d^{(j)})^T \nabla^2 f_k d^{(j)}$

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
 1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
 - a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \neq 0$]
Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
 - b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\} \|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
 2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, [„Newton“-Schritt]
sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
 3. $k \leftarrow k + 1$, GOTO 1.
-

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
- CG benötigt nur Hessematrix mal Vektor \rightarrow *Hessian-free*
- Immer zuerst Schrittlänge 1 für superlineare Konvergenz versuchen!
- CG braucht problemspezifische Prädiktionierer, gefährdet gute Konv.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]

finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$

9. $i \leftarrow i + 1$, GOTO 1.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h

[$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu
 $\min_{\alpha} \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
 finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$

9. $i \leftarrow i + 1$, GOTO 1.

-
- $p^{(1)}$ ist Cauchy-Punkt, CG-Verfahren verbessert monoton \Rightarrow global konv.
 - Benötigt auch nur Hessematrix mal Vektor \rightarrow Hessian-free
 - Es gilt $\|p^{(0)}\| < \dots < \|p^{(i)}\| < \dots \leq \Delta \Rightarrow$ lokal inexakter Newton

Inhaltsübersicht für heute:

Das Konjugierte-Gradienten-Verfahren

Inexakte Newton-Verfahren

Nichtlineare kleinste Quadrate

Das Gauss-Newton-Verfahren

Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$ mit $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$ mit $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$ -normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x .

Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$ mit $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$ -normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x . Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte \tilde{y} mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^m \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{m}{2}} \exp \left(-\frac{1}{2\sigma^2} \sum_{j=1}^m [\tilde{y}_j - \Phi(t_j; x)]^2 \right)$$

aufzutreten.

Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$ mit $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$ -normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x . Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte \tilde{y} mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^m \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^m [\tilde{y}_j - \Phi(t_j; x)]^2\right)$$

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für \tilde{y} und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x)$ mit $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$ -normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x . Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte \tilde{y} mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^m \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{m}{2}} \exp \left(-\frac{1}{2\sigma^2} \sum_{j=1}^m [\tilde{y}_j - \Phi(t_j; x)]^2 \right)$$

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für \tilde{y} und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

$$\longrightarrow \text{Finde } \bar{x} \in \operatorname{Argmin} f(x) := \frac{1}{2} \sum_{j=1}^m r_j^2(x) \text{ mit } r_j(x) := \tilde{y}_j - \Phi(t_j; x).$$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Wegen $\frac{1}{2} \|Ax - b\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - x^T A^T b + \frac{1}{2} b^T b$
und $A^T A \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist
hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow x^*$ erfüllt die **Normalgleichungen** $A^T A x = A^T b$.

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Wegen $\frac{1}{2} \|Ax - b\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - x^T A^T b + \frac{1}{2} b^T b$
und $A^T A \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist
hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow x^*$ erfüllt die **Normalgleichungen** $A^T A x = A^T b$.

Da $A^T A \succeq 0$, bieten sich viele numerische Verfahren zur Lösung an:

- Cholesky-Faktorisierung von $A^T A$ (mit pivotisieren),
- QR-Faktorisierung von A (recht stabil, wird aber dicht besetzt),
- SVD-Dekomposition (am stabilsten und teuersten)
- Präkonditionierte CG-Verfahren (PCG)
(für Näherungslösung im *large scale*-Bereich, falls Av und $A^T w$ billig)

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

$$\nabla f(x) = \sum_{j=1}^m r_j(x) \nabla r_j(x) = J_r(x)^T r(x)$$

Für eine Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $g(x) = (g_1(x), \dots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x . So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.

$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h) \longrightarrow$ das lineare Modell von g in x .

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

$$\begin{aligned} \nabla f(x) &= \sum_{j=1}^m r_j(x) \nabla r_j(x) = J_r(x)^T r(x) \\ \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J_r(x)^T J_r(x) + \underbrace{\sum_{j=1}^m r_j(x) \nabla^2 r_j(x)}_{\text{klein}} \approx J_r(x)^T J_r(x) \end{aligned}$$

→ die Jacobimatrix liefert eine gute Näherung für die Hessematrix.

Für eine Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $g(x) = (g_1(x), \dots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x . So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.

$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h) \longrightarrow$ das lineare Modell von g in x .

Inhaltsübersicht für heute:

Das Konjugierte-Gradienten-Verfahren

Inexakte Newton-Verfahren

Nichtlineare kleinste Quadrate

Das Gauss-Newton-Verfahren

Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$

Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung:
 $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$,
 denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.

Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung:
 $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$,
 denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.
- Die Gleichung für h_k^{GN} hat die Form $A^T A x = A^T b$ (Normalgleichungen),
 h_k^{GN} ist Lösung des linearen kleinste Quadrate Problems

$$\min_h \|J_k^T h - r_k\|^2$$

\Rightarrow alle numerischen Verfahren des linearen Falls sind verwendbar.

Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung:
 $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$,
 denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.

- Die Gleichung für h_k^{GN} hat die Form $A^T A x = A^T b$ (Normalgleichungen),
 h_k^{GN} ist Lösung des linearen kleinste Quadrate Problems

$$\min_h \|J_k^T h - r_k\|^2$$

\Rightarrow alle numerischen Verfahren des linearen Falls sind verwendbar.

- Ist $\text{Rang}(J_k) < n$, verwendet man im **Levenberg-Marquardt-Verfahren**

$$(J_k^T J_k + \lambda I) h = -J_k r_k,$$

mit ähnlichen Anpassungsregeln für λ wie für die Trust-Region.

[Interpretiere λ als Lagrange-Multiplikator für die Trust-Region-Constraint.] 