

Trust-Region- und Quasi-Newton-Verfahren

Laslo Hunhold

Seminar über industrielle Anwendungen
Mathematisches Institut
Universität zu Köln

bei

PD Dr. Thomas MRZIGLOD
Bayer AG, Leverkusen

8. Januar 2018



Einführung

Einführung

„Programmierung“

Einführung

„Programmierung“

- ▶ Formulierung des Problems

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch
- ▶ Lokal

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch
- ▶ Lokal

Anforderung an Algorithmen

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch
- ▶ Lokal

Anforderung an Algorithmen

- ▶ Robust

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch
- ▶ Lokal

Anforderung an Algorithmen

- ▶ Robust
- ▶ Effizient

Einführung

„Programmierung“

- ▶ Formulierung des Problems
- ▶ Entwicklung von Algorithmen
- ▶ Analyse der Algorithmen

Problemeigenschaften

- ▶ Unbeschränkt
- ▶ Stetig
- ▶ Deterministisch
- ▶ Lokal

Anforderung an Algorithmen

- ▶ Robust
- ▶ Effizient
- ▶ Genau

Mathematische Formulierung

Problem

Mathematische Formulierung

Problem

$$\begin{array}{ll} \min_x & f(x) \\ \text{gemäß} & x \in \Omega \end{array}$$

Mathematische Formulierung

Problem

$$\begin{array}{ll} \min_x & f(x) \\ \text{gemäß} & x \in \Omega \end{array}$$

Parameter

Mathematische Formulierung

Problem

$$\begin{array}{ll} \min_x & f(x) \\ \text{gemäß} & x \in \Omega \end{array}$$

Parameter

- ▶ Lösungsraum Ω (überabzählbar)

Mathematische Formulierung

Problem

$$\min_x f(x)$$

gemäß $x \in \Omega$

Parameter

- ▶ Lösungsraum Ω (überabzählbar)
- ▶ Bewertungsfunktion $f: \Omega \rightarrow \mathbb{R}$ („glatt“)

Mathematische Formulierung

Problem

$$\min_x f(x)$$

gemäß $x \in \Omega$

Parameter

- ▶ Lösungsraum Ω (überabzählbar)
- ▶ Bewertungsfunktion $f: \Omega \rightarrow \mathbb{R}$ („glatt“)
- ▶ Variable x

Mathematische Formulierung

Lösungsbegriff und Strategie

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*) : f(x^*) \leq f(x)$

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*) : f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*) : f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Taylorformel $\forall x, p \in \Omega: x + p \in \Omega:$

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*): f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Taylorformel $\forall x, p \in \Omega: x + p \in \Omega:$

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

→ NB 1. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^1(\mathcal{N}(x^*)) \Rightarrow$
 $\nabla f(x^*) = 0$

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*): f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Taylorformel $\forall x, p \in \Omega: x + p \in \Omega:$

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

→ NB 1. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^1(\mathcal{N}(x^*)) \Rightarrow$
 $\nabla f(x^*) = 0$

→ NB 2. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^2(\mathcal{N}(x^*)) \Rightarrow$
 $\nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ positiv semidefinit

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*): f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Taylorformel $\forall x, p \in \Omega: x + p \in \Omega:$

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

→ NB 1. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^1(\mathcal{N}(x^*)) \Rightarrow \nabla f(x^*) = 0$

→ NB 2. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^2(\mathcal{N}(x^*)) \Rightarrow \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ positiv semidefinit

→ HB 2. Ordnung $f \in \mathcal{C}^2(\mathcal{N}(x^*)) \wedge \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ pos. def. $\Rightarrow x^*$ streng lokaler Minimierer

Mathematische Formulierung

Lösungsbegriff und Strategie

Lokaler Minimierer $x^* \exists \mathcal{N}(x^*) \subseteq \Omega: \forall x \in \mathcal{N}(x^*): f(x^*) \leq f(x)$

Stationärer Punkt $x^* \nabla f(x^*) = 0$

Taylorformel $\forall x, p \in \Omega: x + p \in \Omega:$

$$f(x + p) \approx f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

→ NB 1. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^1(\mathcal{N}(x^*)) \Rightarrow \nabla f(x^*) = 0$

→ NB 2. Ordnung x^* lokaler Minimierer $\wedge f \in \mathcal{C}^2(\mathcal{N}(x^*)) \Rightarrow \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ positiv semidefinit

→ HB 2. Ordnung $f \in \mathcal{C}^2(\mathcal{N}(x^*)) \wedge \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ pos. def. $\Rightarrow x^*$ streng lokaler Minimierer

→ Finde x^* mit $\nabla f(x^*) = 0$

Algorithmen

Überblick

Algorithmen

Überblick

Parameter

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

- ▶ $x_{k+1} \leftarrow \{x_k, \dots, x_0, \text{Verhalten von } f \text{ in } x_k\}$

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

- ▶ $x_{k+1} \leftarrow \{x_k, \dots, x_0, \text{Verhalten von } f \text{ in } x_k\}$
- ▶ 2 Strategien Liniensuchverfahren und Trust-Region-Verfahren

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

- ▶ $x_{k+1} \leftarrow \{x_k, \dots, x_0, \text{Verhalten von } f \text{ in } x_k\}$
- ▶ 2 Strategien Liniensuchverfahren und Trust-Region-Verfahren

Ziel

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

- ▶ $x_{k+1} \leftarrow \{x_k, \dots, x_0, \text{ Verhalten von } f \text{ in } x_k\}$
- ▶ 2 Strategien Liniensuchverfahren und Trust-Region-Verfahren

Ziel

- ▶ monotoner Algorithmus $f(x_{k+1}) < f(x_k)$

Algorithmen

Überblick

Parameter

- ▶ Startpunkt $x_0 \in \Omega$
- ▶ Iterierte $\{x_k\}_{k=0}^{\infty}$

Schritt

- ▶ $x_{k+1} \leftarrow \{x_k, \dots, x_0, \text{Verhalten von } f \text{ in } x_k\}$
- ▶ 2 Strategien Liniensuchverfahren und Trust-Region-Verfahren

Ziel

- ▶ monotoner Algorithmus $f(x_{k+1}) < f(x_k)$
- ▶ nichtmonotoner Algorithmus $\exists m > 0: f(x_{k+m}) < f(x_k)$

Algorithmen

Liniensuchverfahren

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ *Wähle* Suchrichtung p_k

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ Wähle Suchrichtung p_k
- ▶ Suche Schrittweite $\alpha_k = \operatorname{argmin}_{\alpha > 0} (f(x_k + \alpha \cdot p_k))$
 - ▶ „Liniensuche“

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ Wähle Suchrichtung p_k
- ▶ Suche Schrittweite $\alpha_k = \operatorname{argmin}_{\alpha > 0} (f(x_k + \alpha_k \cdot p_k))$
 - ▶ „Liniensuche“
 - ▶ Approximativ

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ Wähle Suchrichtung p_k
- ▶ Suche Schrittweite $\alpha_k = \operatorname{argmin}_{\alpha > 0} (f(x_k + \alpha_k \cdot p_k))$
 - ▶ „Liniensuche“
 - ▶ Approximativ
 - ▶ Sample-Bewertung mit Wolfe- bzw. Goldstein-Bedingungen

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ Wähle Suchrichtung p_k
- ▶ Suche Schrittweite $\alpha_k = \operatorname{argmin}_{\alpha > 0} (f(x_k + \alpha_k \cdot p_k))$
 - ▶ „Liniensuche“
 - ▶ Approximativ
 - ▶ Sample-Bewertung mit Wolfe- bzw. Goldstein-Bedingungen
 - ▶ Nicht näher betrachtet

Algorithmen

Liniensuchverfahren

Schritt $k \rightarrow k + 1$

- ▶ Wähle Suchrichtung p_k
- ▶ Suche Schrittweite $\alpha_k = \operatorname{argmin}_{\alpha > 0} (f(x_k + \alpha_k \cdot p_k))$
 - ▶ „Liniensuche“
 - ▶ Approximativ
 - ▶ Sample-Bewertung mit Wolfe- bzw. Goldstein-Bedingungen
 - ▶ Nicht näher betrachtet
- ▶ Setze $x_{k+1} = x_k + \alpha_k \cdot p_k$

Algorithmen

Trust-Region-Verfahren

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
 - ▶ $\nabla f_k := \nabla f(x_k)$
 - ▶ B_k symmetrisch
- ▶ *Wiederhole*

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

- ▶ *Bestimme Trust-Region-Radius* Δ_{k+1} aus ρ_k

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

- ▶ *Bestimme Trust-Region-Radius* Δ_{k+1} aus ρ_k
- ▶ *Wenn* $\rho_k > \eta$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

- ▶ *Bestimme Trust-Region-Radius* Δ_{k+1} aus ρ_k
- ▶ *Wenn* $\rho_k > \eta$
 - ▶ $x_{k+1} = x_k + p_k$

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

- ▶ *Bestimme Trust-Region-Radius* Δ_{k+1} aus ρ_k
- ▶ *Wenn* $\rho_k > \eta$
 - ▶ $x_{k+1} = x_k + p_k$

Sonst

Algorithmen

Trust-Region-Verfahren

Schritt $k \rightarrow k + 1$

- ▶ *Formuliere Modellfunktion*

$$m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p \approx f(x_k + p)$$

- ▶ $f_k := f(x_k)$
- ▶ $\nabla f_k := \nabla f(x_k)$
- ▶ B_k symmetrisch

- ▶ *Wiederhole*

- ▶ *Suche Suchrichtung* $p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$
- ▶ *Berechne Vereinbarkeit* $\rho_k(p_k) = \frac{f(x_k) - f(x_k + p_k)}{m(0) - m(p_k)}$

solange $\rho_k < 0$

- ▶ *Bestimme Trust-Region-Radius* Δ_{k+1} aus ρ_k
- ▶ *Wenn* $\rho_k > \eta$
 - ▶ $x_{k+1} = x_k + p_k$

Sonst

- ▶ $x_{k+1} = x_k$

Algorithmen

Vergleich

Algorithmen

Vergleich

Unterschied im Ansatz

Algorithmen

Vergleich

Unterschied im Ansatz

- ▶ Jeweils umgekehrte *Reihenfolge*

Algorithmen

Vergleich

Unterschied im Ansatz

- ▶ Jeweils umgekehrte *Reihenfolge*

- ▶ **Liniensuchverfahren**

Wähle Suchrichtung p_k und suche Schrittweite α_k

Algorithmen

Vergleich

Unterschied im Ansatz

- ▶ Jeweils umgekehrte *Reihenfolge*
- ▶ **Liniensuchverfahren**
Wähle Suchrichtung p_k und suche Schrittweite α_k
- ▶ **Trust-Region-Verfahren**
Wähle maximale Schrittweite $\Delta_k \geq \alpha_k$ und suche Suchrichtung p_k

Algorithmen

Vergleich

Unterschied im Ansatz

- ▶ Jeweils umgekehrte *Reihenfolge*
- ▶ Liniensuchverfahren
Wähle Suchrichtung p_k und suche Schrittweite α_k
- ▶ Trust-Region-Verfahren
Wähle maximale Schrittweite $\Delta_k \geq \alpha_k$ und suche Suchrichtung p_k

Gemeinsames Problem

- ▶ Wahl bzw. Suche der Suchrichtung

Liniensuchverfahren

Wahl der Suchrichtung

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow \boxed{p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow \boxed{p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow \boxed{p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow \boxed{p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$
- ▶ Löse $0 = \nabla m_k(p) = \nabla f_k + (\nabla^2 f_k)^T p = \nabla f_k + \nabla^2 f_k p$

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$
- ▶ Löse $0 = \nabla m_k(p) = \nabla f_k + (\nabla^2 f_k)^T p = \nabla f_k + \nabla^2 f_k p$

$$\rightarrow p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$$

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$
- ▶ Löse $0 = \nabla m_k(p) = \nabla f_k + (\nabla^2 f_k)^T p = \nabla f_k + \nabla^2 f_k p$

$$\rightarrow p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$$

- ▶ 2. Ordnung \rightarrow Schnelle Konvergenz, langsame Berechnung

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$
- ▶ Löse $0 = \nabla m_k(p) = \nabla f_k + (\nabla^2 f_k)^T p = \nabla f_k + \nabla^2 f_k p$

$$\rightarrow p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$$

- ▶ 2. Ordnung \rightarrow Schnelle Konvergenz, langsame Berechnung
- ▶ Finde Kompromiß

Liniensuchverfahren

Wahl der Suchrichtung

Steilster Abstieg

$$\rightarrow p_k^S = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

- ▶ 1. Ordnung \rightarrow Einfache Berechnung, langsame Konvergenz
- ▶ $\forall \tilde{p}_k: |\angle(p_k^S, \tilde{p}_k)| < \frac{\pi}{2}: \forall \tilde{\alpha}_k \text{ klein: } f(x_k + \tilde{\alpha}_k \cdot \tilde{p}_k) < f(x_k)$

Newton-Richtung

- ▶ Setze $m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T \nabla^2 f_k p \approx f(x_k + p)$
- ▶ Löse $0 = \nabla m_k(p) = \nabla f_k + (\nabla^2 f_k)^T p = \nabla f_k + \nabla^2 f_k p$

$$\rightarrow p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$$

- ▶ 2. Ordnung \rightarrow Schnelle Konvergenz, langsame Berechnung
- ▶ Finde Kompromiß \rightarrow Quasi-Newton-Verfahren

Liniensuchverfahren

Quasi-Newton-Verfahren

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow p_k^{QN} = -B_k^{-1} \nabla f_k$

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow p_k^{QN} = -B_k^{-1} \nabla f_k$

Bedingungen an B_k

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k$

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow \boxed{B_{k+1} s_k = y_k}$ mit $s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow$

$$\boxed{B_{k+1} s_k = y_k} \text{ mit } s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$$

- ▶ Kurvaturbedingung

$$B_k \text{ positiv definit} \rightarrow \boxed{s_k^T y_k > 0}$$

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow \boxed{B_{k+1} s_k = y_k}$ mit $s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$
- ▶ Kurvaturbedingung
 B_k positiv definit $\rightarrow \boxed{s_k^T y_k > 0}$
- ▶ Symmetrisch wie $\nabla^2 f_k$

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow p_k^{QN} = -B_k^{-1} \nabla f_k$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow$

$$B_{k+1} s_k = y_k \quad \text{mit } s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$$

- ▶ Kurvaturbedingung

$$B_k \text{ positiv definit} \rightarrow s_k^T y_k > 0$$

- ▶ Symmetrisch wie $\nabla^2 f_k$

- ▶ $\text{rank}(B_{k+1} - B_k)$ klein

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow$

$$\boxed{B_{k+1} s_k = y_k} \text{ mit } s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$$

- ▶ Kurvaturbedingung

$$B_k \text{ positiv definit} \rightarrow \boxed{s_k^T y_k > 0}$$

- ▶ $\boxed{\text{Symmetrisch}}$ wie $\nabla^2 f_k$

- ▶ $\boxed{\text{rank}(B_{k+1} - B_k) \text{ klein}}$

Verfahren

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

- ▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow$

$$\boxed{B_{k+1} s_k = y_k} \text{ mit } s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$$

- ▶ Kurvaturbedingung

$$B_k \text{ positiv definit} \rightarrow \boxed{s_k^T y_k > 0}$$

- ▶ $\boxed{\text{Symmetrisch}}$ wie $\nabla^2 f_k$

- ▶ $\boxed{\text{rank}(B_{k+1} - B_k) \text{ klein}}$

Verfahren

- ▶ BFGS („Broyden-Fletcher-Goldfarb-Shanno“)

Liniensuchverfahren

Quasi-Newton-Verfahren

Approximiere $B_k \approx \nabla^2 f_k \rightarrow \boxed{p_k^{QN} = -B_k^{-1} \nabla f_k}$

Bedingungen an B_k

▶ Sekantengleichung $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k \rightarrow$

$$\boxed{B_{k+1} s_k = y_k} \text{ mit } s_k := x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$$

▶ Kurvaturbedingung

$$B_k \text{ positiv definit} \rightarrow \boxed{s_k^T y_k > 0}$$

▶ Symmetrisch wie $\nabla^2 f_k$

▶ $\boxed{\text{rank}(B_{k+1} - B_k) \text{ klein}}$

Verfahren

- ▶ BFGS („Broyden-Fletcher-Goldfarb-Shanno“)
- ▶ SR1 („Symmetric Rank-One“)

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

$$\rightarrow \boxed{B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}}$$

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

$$\rightarrow \boxed{B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}}$$

Eigenschaften

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

$$\rightarrow \boxed{B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}}$$

Eigenschaften

- ▶ B_k positiv definit

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

$$\rightarrow \boxed{B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}}$$

Eigenschaften

- ▶ B_k positiv definit
- ▶ $\text{rank}(B_{k+1} - B_k) = \text{rank}(U_k + V_k) = \text{rank}(U_k) + \text{rank}(V_k) = 2$

Liniensuchverfahren

Quasi-Newton-Verfahren (BFGS)

Herleitung

- ▶ $B_{k+1} = B_k + U_k + V_k$ mit U_k, V_k symmetrisch-Rang-1
- ▶ B_{k+1} symmetrisch \rightarrow Setze $U_k = \gamma uu^T$, $V_k = \mu vv^T$ mit $\gamma, \mu \in \mathbb{R}$, $u, v \in \mathbb{R}^n$
- ▶ Wähle $u = B_k s_k$, $v = y_k$
- ▶ Sekantengleichung $\rightarrow \gamma = -\frac{1}{s_k^T B_k s_k}$, $\mu = \frac{1}{y_k^T s_k}$

$$\rightarrow \boxed{B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}}$$

Eigenschaften

- ▶ B_k positiv definit
- ▶ $\text{rank}(B_{k+1} - B_k) = \text{rank}(U_k + V_k) = \text{rank}(U_k) + \text{rank}(V_k) = 2$
- ▶ Global superlinear konvergent

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

- ▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

- ▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$
- ▶ Sekantengleichung

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

- ▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$
- ▶ Sekantengleichung \rightarrow
 $B_{k+1}s_k = y_k =$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

- ▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$
- ▶ Sekantengleichung \rightarrow
 $B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k =$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

$$\rightarrow \sigma = \operatorname{sgn} \left([s_k^T (y_k - B_k s_k)] \right), \quad \alpha = \pm \left| s_k^T (y_k - B_k s_k) \right|^{-\frac{1}{2}}$$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

► $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

► Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

► $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

$$\rightarrow \sigma = \operatorname{sgn} \left([s_k^T (y_k - B_k s_k)] \right), \quad \alpha = \pm |s_k^T (y_k - B_k s_k)|^{-\frac{1}{2}}$$

$$\rightarrow \boxed{B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}}$$

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

$$\rightarrow \sigma = \operatorname{sgn} \left([s_k^T (y_k - B_k s_k)] \right), \quad \alpha = \pm |s_k^T (y_k - B_k s_k)|^{-\frac{1}{2}}$$

$$\rightarrow \boxed{B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}}$$

Eigenschaften

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

$$\rightarrow \sigma = \operatorname{sgn} \left([s_k^T (y_k - B_k s_k)] \right), \quad \alpha = \pm |s_k^T (y_k - B_k s_k)|^{-\frac{1}{2}}$$

$$\rightarrow \boxed{B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}}$$

Eigenschaften

▶ B_k i.A. nicht positiv definit

Liniensuchverfahren

Quasi-Newton-Verfahren (SR1)

Herleitung

▶ $B_{k+1} = B_k + \sigma uu^T$ mit $\sigma \in \{-1, 1\}$, $u \in \mathbb{R}^n$

▶ Sekantengleichung \rightarrow

$$B_{k+1}s_k = y_k = B_k s_k + \sigma uu^T s_k = B_k s_k + [\sigma u^T s_k] u$$

$$\rightarrow u = \alpha(y_k - B_k s_k)$$

▶ $y_k - B_k s_k = \sigma \alpha^2 (y_k^T - s_k^T B_k) s_k (y_k - B_k s_k)$
 $= \left\{ \sigma \alpha^2 [s_k^T (y_k - B_k s_k)] \right\} (y_k - B_k s_k)$

$$\rightarrow \sigma = \operatorname{sgn} \left([s_k^T (y_k - B_k s_k)] \right), \quad \alpha = \pm |s_k^T (y_k - B_k s_k)|^{-\frac{1}{2}}$$

$$\rightarrow \boxed{B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}}$$

Eigenschaften

▶ B_k i.A. nicht positiv definit

▶ $\operatorname{rank}(B_{k+1} - B_k) = \operatorname{rank}(\sigma uu^T) = 1$

Trust-Region-Verfahren

Suche der Suchrichtung

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p))$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

$$\rightarrow m_k(p) = f_k + (\nabla f_k)^T p$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

$$\rightarrow m_k(p) = f_k + (\nabla f_k)^T p$$

$$\rightarrow p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

$$\rightarrow m_k(p) = f_k + (\nabla f_k)^T p$$

$$\rightarrow p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|} = -\Delta_k p_k^S$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

$$\rightarrow m_k(p) = f_k + (\nabla f_k)^T p$$

$$\rightarrow p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|} = -\Delta_k p_k^S \rightarrow \text{Steilster Abstieg}$$

Trust-Region-Verfahren

Suche der Suchrichtung

Erinnerung

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} (m_k(p)) \text{ mit } m_k(p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p$$

Lineares Problem $B_k = 0$

$$\rightarrow m_k(p) = f_k + (\nabla f_k)^T p$$

$$\rightarrow \boxed{p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}} = -\Delta_k p_k^S \rightarrow \text{Steilster Abstieg}$$

Trust-Region-Newton-Verfahren $B_k = \nabla^2 f_k$

► $\|p\| \leq \Delta_k \rightarrow \nabla^2 f_k$ muß nicht positiv definit sein

→ Trust-Region-Quasi-Newton-Verfahren mit SR1

Trust-Region-Verfahren

Cauchy-Punkt

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur ausreichende Reduktion

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur ausreichende Reduktion
- ▶ Quantifiziere Reduktion mit Cauchy-Punkt p_k^C

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur ausreichende Reduktion
- ▶ Quantifiziere Reduktion mit Cauchy-Punkt p_k^C

Vorgehen

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur **ausreichende Reduktion**
- ▶ Quantifiziere Reduktion mit **Cauchy-Punkt** p_k^C

Vorgehen

- ▶ *Berechne* Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur ausreichende Reduktion
- ▶ Quantifiziere Reduktion mit Cauchy-Punkt p_k^C

Vorgehen

- ▶ Berechne Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$
- ▶ Bestimme $\tau_k = \operatorname{argmin}_{\tau \in (0,1]} \left(m_k \left(\tau p_k^L \right) \right)$

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur **ausreichende Reduktion**
- ▶ Quantifiziere Reduktion mit **Cauchy-Punkt** p_k^C

Vorgehen

- ▶ *Berechne* Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$
- ▶ *Bestimme* $\tau_k = \operatorname{argmin}_{\tau \in (0,1]} \left(m_k \left(\tau p_k^L \right) \right)$

$$\rightarrow \boxed{p_k^C = \tau_k p_k^L}$$

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur **ausreichende Reduktion**
- ▶ Quantifiziere Reduktion mit **Cauchy-Punkt** p_k^C

Vorgehen

- ▶ *Berechne* Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$
- ▶ *Bestimme* $\tau_k = \operatorname{argmin}_{\tau \in (0,1]} \left(m_k \left(\tau p_k^L \right) \right)$

$$\rightarrow \boxed{p_k^C = \tau_k p_k^L}$$

Nutzen

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur **ausreichende Reduktion**
- ▶ Quantifiziere Reduktion mit **Cauchy-Punkt** p_k^C

Vorgehen

- ▶ *Berechne* Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$
- ▶ *Bestimme* $\tau_k = \operatorname{argmin}_{\tau \in (0,1]} \left(m_k \left(\tau p_k^L \right) \right)$

$$\rightarrow \boxed{p_k^C = \tau_k p_k^L}$$

Nutzen

- ▶ Einfach zu berechnen

Trust-Region-Verfahren

Cauchy-Punkt

Motivation

- ▶ Exakter Schritt p_k nicht nötig, nur ausreichende Reduktion
- ▶ Quantifiziere Reduktion mit Cauchy-Punkt p_k^C

Vorgehen

- ▶ Berechne Lösung $p_k^L = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}$ des linearen Problems mit $B_k = 0$
- ▶ Bestimme $\tau_k = \operatorname{argmin}_{\tau \in (0,1]} \left(m_k \left(\tau p_k^L \right) \right)$

$$\rightarrow \boxed{p_k^C = \tau_k p_k^L}$$

Nutzen

- ▶ Einfach zu berechnen
- ▶ Globale Konvergenz, falls Verfahren pro Schritt mindestens Cauchy-Reduktion erreicht

Trust-Region-Verfahren

Analyse

Trust-Region-Verfahren

Analyse

- ▶ Global konvergent für $\eta = 0$ und $\eta \in \left(0, \frac{1}{4}\right)$ unter jeweils schwachen Bedingungen

Trust-Region-Verfahren

Analyse

- ▶ Global konvergent für $\eta = 0$ und $\eta \in \left(0, \frac{1}{4}\right)$ unter jeweils schwachen Bedingungen
- ▶ Kreisförmige Trust-Regions nicht geeignet für schlecht-skalierende Funktionen

Ausblick

Ausblick

- ▶ Sherman-Morrison-Formel für analytischen Ausdruck

$$H_k := B_k^{-1}$$

Ausblick

- ▶ Sherman-Morrison-Formel für analytischen Ausdruck

$$H_k := B_k^{-1}$$

- ▶ Nicht-euklidische Trust-Regions

Ende

Vielen Dank für Ihre Aufmerksamkeit!