Ein Filter-Trust-Region-Verfahren für nichtlineare Gleichungssysteme

Markus Kaiser

Institut für Angewandte Mathematik
Universität Erlangen–Nürnberg

30. Juli 2008

Übersicht

Problemstellung

Lösen eines nichtlinearen Gleichungssystems der Form

$$c(x) = 0,$$

wobei $x \in \mathbb{R}^n$ und $c : \mathbb{R}^n \to \mathbb{R}^m$ (zweimal stetig differenzierbar).

Äquivalente Umformung des Problems:

- Man betrachtet den Funktionsvektor c als Menge $C := \{c_i(x)\}\,,\; i \in \{1, ..., m\}$ einzelner Funktionen.

Äquivalente Umformung des Problems:

- Man betrachtet den Funktionsvektor c als Menge $C := \{c_i(x)\}, \; i \in \{1, ..., m\}$ einzelner Funktionen.
- Die Menge C wird in p nicht notwendigerweise disjunkte Teilmengen $I_1 \cup I_2 \cup ... \cup I_p = \{1, ..., m\}$ unterteilt.

Äquivalente Umformung des Problems:

- Man betrachtet den Funktionsvektor c als Menge $C := \{c_i(x)\}, \, i \in \{1, ..., m\}$ einzelner Funktionen.
- Die Menge C wird in p nicht notwendigerweise disjunkte Teilmengen $I_1 \cup I_2 \cup ... \cup I_p = \{1, ..., m\}$ unterteilt.
- Man definiert einen Gesamtfehler für jede Teilmenge von Funktionen aus C als $\theta_j(x) := \|c_{I_j}(x)\|_2$ für $j = 1, ..., p$.

Äquivalente Umformung des Problems:

- Man betrachtet den Funktionsvektor c als Menge $C := \{c_i(x)\}$, $i \in \{1, \dots, m\}$ einzelner Funktionen.
- Die Menge C wird in p nicht notwendigerweise disjunkte Teilmengen $I_1 \cup I_2 \cup \dots \cup I_p = \{1, \dots, m\}$ unterteilt.
- Man definiert einen Gesamtfehler für jede Teilmenge von Funktionen aus C als $\theta_j(x) := \|c_{I_j}(x)\|_2$ für $j = 1, \dots, p$.
- Damit lässt sich das nichtlineare Gleichungssystem als Minimierungsaufgabe

$$\min_{x \in \mathbb{R}} f(x) = \frac{1}{2} \|\theta(x)\|_2^2$$

des Fehlers $\theta(x) = (\theta_1(x), \dots, \theta_p(x))^T$ formulieren.

Benötigte Hilfsmittel

Dieses Minimierungsproblem soll nun unter Verwendung eines Trust-Region Verfahrens mit Hilfe eines mehrdimensionalen Filters gelöst werden.

Benötigte Hilfsmittel

Dieses Minimierungsproblem soll nun unter Verwendung eines Trust-Region Verfahrens mit Hilfe eines mehrdimensionalen Filters gelöst werden.
Die Vorteile, die man dadurch erwartet, sind:

- Trust-Region $\longrightarrow$ Konvergenz des Algorithmus von jedem beliebigen Startpunkt, d.h. globale Konvergenz gegen ein lokales Minimum

Benötigte Hilfsmittel

Dieses Minimierungsproblem soll nun unter Verwendung eines Trust-Region Verfahrens mit Hilfe eines mehrdimensionalen Filters gelöst werden.
Die Vorteile, die man dadurch erwartet, sind:

- Trust-Region $\longrightarrow$ Konvergenz des Algorithmus von jedem beliebigen Startpunkt, d.h. globale Konvergenz gegen ein lokales Minimum
- Filter $\longrightarrow$ Sicherung der Zulässigkeit und "Beschleunigung" des Verfahrens, d.h. Vermeiden von überflüssigen Iterationsschritten.

Trust-Region Verfahren – Einführung

- Problemformulierung: Minimiere eine reellwertige Zielfunktion ohne Nebenbedingungen: $\min_{x \in \mathbb{R}^n} f(x)$

Trust-Region Verfahren – Einführung

- Problemformulierung: Minimiere eine reellwertige Zielfunktion ohne Nebenbedingungen: $\min_{x \in \mathbb{R}^n} f(x)$
- Prinzipielles Vorgehen:
  - Die Zielfunktion wird in einer Umgebung um den Iterationswert x_k mit Hilfe einer einfach zu minimierenden Modellfunktion m_k approximiert.
  - Das Minimum von m_k innerhalb dieser Umgebung liefert einen Kandidaten für den neuen Iterationswert x_k^* mit dazugehörigem Iterationsschritt $s_k := x_k^* - x_k$.
  - Es wird überprüft, ob der Kandidat x_k^* als nächster Iterationswert geeignet ist. Falls nicht, wird die Umgebung verkleinert und die Modellfunktion angepasst.
  - Diese Schritte werden mit jedem neuen Iterationswert wiederholt, bis man ein lokales Minimum der Zielfunktion erreicht.

- **Definition:** *Trust-Region* wird das Gebiet um den aktuellen Iterationspunkt x_k genannt, in dem man von der Modellfunktion erwartet, dass sie die Zielfunktion hinreichend gut approximiert. Definiert wird die Trust-Region durch den Trust-Region-Radius $\Delta_k \in \mathbb{R}_+$:

$$\mathcal{B}_k = \{x \in \mathbb{R}^n : \|x - x_k\|_2 \leq \Delta_k\}$$

- Akzeptanzbedingung für einen Kandidaten x_k^*:
  - Die Modellfunktion muss dazu mit der Zielfunktion hinreichend gut übereinstimmmen
  - Vergleich der vorhergesagten Verbesserung (durch die Modellfunkion) mit der tatsächlichen Verbesserung (der Zielfunktion):

$$\rho_k := \frac{f(x_k) - f(x_k^*)}{m_k(x_k) - m_k(x_k^*)}$$

- Aktualisierung des Trust-Region-Radius:
  - Die Modellfunktion approximiert die Zielfunktion umso besser für einen Wert $x_k + s_k$, $s_k \in \mathbb{R}^n$, je kleiner $\|s_k\|_2$ ist.
  - Falls x_k^* nicht als neuer Iterationspunkt übernommen werden kann, wird Δ_k verkleinert, um so bei der nächsten Iteration einen Punkt zu finden, an dem Modell- und Zielfunktion besser übereinstimmen.
  - Δ_k kann auch vergrößert werden, wenn die Übereinstimmung zwischen Modell- und Zielfunktion ausreichend ist.

Trust-Region Algorithmus

Schritt 1 (Initialisierung): Ein Startwert x_0 und ein Trust-Region Radius Δ_0 sind gegeben. Außerdem sind die Konstanten η_1, η_2, γ_1 und γ_2 gegeben, für die $0 < \eta_1 \leq \eta_2 < 1$ und $0 < \gamma_1 \leq \gamma_2 < 1$ gilt. Berechne $f(x_0)$ und setze $k = 0$.

Trust-Region Algorithmus

Schritt 1 (Initialisierung): Ein Startwert x_0 und ein Trust-Region Radius Δ_0 sind gegeben. Außerdem sind die Konstanten η_1, η_2, γ_1 und γ_2 gegeben, für die $0 < \eta_1 \leq \eta_2 < 1$ und $0 < \gamma_1 \leq \gamma_2 < 1$ gilt. Berechne $f(x_0)$ und setze $k = 0$.

Schritt 2 (Wahl der Modellfunktion): Wähle eine Modellfunktion m_k innerhalb der Trust-Region $\mathcal{B}_k$.

Trust-Region Algorithmus

Schritt 1 (Initialisierung): Ein Startwert x_0 und ein Trust-Region Radius Δ_0 sind gegeben. Außerdem sind die Konstanten η_1, η_2, γ_1 und γ_2 gegeben, für die $0 < \eta_1 \leq \eta_2 < 1$ und $0 < \gamma_1 \leq \gamma_2 < 1$ gilt. Berechne $f(x_0)$ und setze $k = 0$.

Schritt 2 (Wahl der Modellfunktion): Wähle eine Modellfunktion m_k innerhalb der Trust-Region $\mathcal{B}_k$.

Schritt 3 (Berechnung der Schrittweite): Berechne einen Schritt s_k, der den Wert der Modellfunktion m_k ausreichend verringert und der die Bedingung $x_k + s_k \in \mathcal{B}_k$ erfüllt.

Schritt 4 (Akzeptanzkriterium): Berechne $f(x_k + s_k)$ und bestimme $\rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$. Ist $\rho_k \geq \eta_1$, dann ist $x_{k+1} = x_k + s_k$, andernfalls ist $x_{k+1} = x_k$.

Schritt 4 (Akzeptanzkriterium): Berechne $f(x_k + s_k)$ und bestimme $\rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$. Ist $\rho_k \geq \eta_1$, dann ist $x_{k+1} = x_k + s_k$, andernfalls ist $x_{k+1} = x_k$.

Schritt 5 (Aktualisierung des Trust-Region-Radius): Setzte

$$\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \textit{falls} \quad \rho_k < \eta_1 \\ [\gamma_2 \Delta_k, \Delta_k] & \textit{falls} \quad \rho_k \in [\eta_1, \eta_2) \\ [\Delta_k, \infty) & \textit{falls} \quad \rho_k \geq \eta_2 \end{cases}$$

Erhöhe k um 1 und gehe zu Schritt 2.

Mehrdimensionaler Filter – Definitionen

- Erinnerung: Das Ziel ist es ein Minimierungsproblem der Form

$$\min_{x \in \mathbb{R}} f(x) = \frac{1}{2} \|\theta(x)\|_2^2$$

zu lösen. Dabei sind die Komponenten von θ die Fehler der einzelnen Funktionsteilmengen des Funktionsvektors c.

Mehrdimensionaler Filter – Definitionen

- Erinnerung: Das Ziel ist es ein Minimierungsproblem der Form

$$\min_{x \in \mathbb{R}} f\left(x\right) = \frac{1}{2} \|\theta\left(x\right)\|_2^2$$

  zu lösen. Dabei sind die Komponenten von θ die Fehler der einzelnen Funktionsteilmengen des Funktionsvektors c.
- **Definition:** Ein Punkt x_1 *dominiert* einen Punkt x_2, falls gilt:

$$\theta_j\left(x_1\right) \leq \theta_j\left(x_2\right)$$

  für alle $j = 1, ..., p$

Mehrdimensionaler Filter – Definitionen

- Wird x_2 von x_1 dominiert, so ist er vernachlässigbar, da x_1 für alle Funktionsteilmengen einen mindestens ebenso guten Wert liefert wie x_2.

Mehrdimensionaler Filter – Definitionen

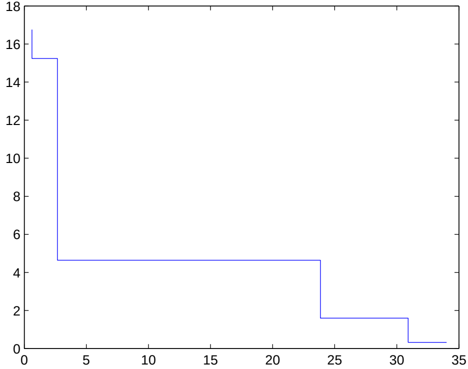
- Wird x_2 von x_1 dominiert, so ist er vernachlässigbar, da x_1 für alle Funktionsteilmengen einen mindestens ebenso guten Wert liefert wie x_2.
- **Definition:** Ein *Filter* ist eine Menge F von Tupeln der Gestalt $(\theta_{1,k}, ..., \theta_{p,k})$, so dass $\theta_{j,k} < \theta_{j,l}$ für mindestens ein $j \in \{1, ..., p\}$ $k \neq l$ gilt.

Mehrdimensionaler Filter – Definitionen

- Wird x_2 von x_1 dominiert, so ist er vernachlässigbar, da x_1 für alle Funktionsteilmengen einen mindestens ebenso guten Wert liefert wie x_2.
- **Definition:** Ein *Filter* ist eine Menge F von Tupeln der Gestalt $(\theta_{1,k}, \dots, \theta_{p,k})$, so dass $\theta_{j,k} < \theta_{j,l}$ für mindestens ein $j \in \{1, \dots, p\}$ $k \neq l$ gilt.
- Ein Filter besteht also nur aus den Fehlervektoren von Iterationswerten, die von keinem anderen Fehlervektor dominiert werden.

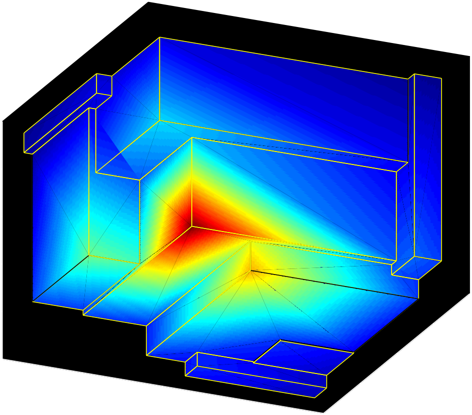
Beispiel

Beispiel eines zweidimensionalen Filters:



Beispiel

Beispiel eines dreidimensionalen Filters:



Akzeptanzbedingung des Filters

- Um ein Mindestmaß an Verbesserung zu gewährleisten, führt man folgende Akzeptanzbedingung ein:
  Ein Testpunkt x_k^* wird genau dann vom Filter F akzeptiert, wenn

$$\forall \theta\left(x_l\right) \in F \quad \exists j \in \{1, ..., p\} \quad \theta_j\left(x_k^+\right) < \theta_j\left(x_l\right) - \gamma$$

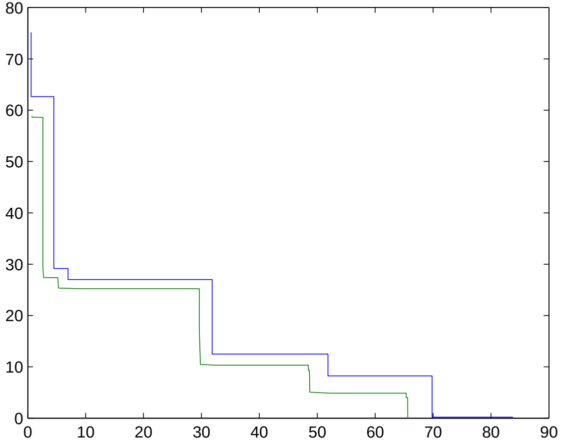
Akzeptanzbedingung des Filters

- Um ein Mindestmaß an Verbesserung zu gewährleisten, führt man folgende Akzeptanzbedingung ein:
  Ein Testpunkt x_k^* wird genau dann vom Filter F akzeptiert, wenn

$$\forall \theta(x_l) \in F \quad \exists j \in \{1, ..., p\} \quad \theta_j\left(x_k^+\right) < \theta_j(x_l) - \gamma$$

- Ein solcher Filter wird bei iterativen Optimierungsverfahren verwendet, um überflüssige Iterationen zu vermeiden und die Konvergenz gegen einen zulässigen Punkt zu gewährleisten.

Beispiel eines zweidimensionalen Filters mit envelope:



FTR-Algorithmus

Schritt 1 (Initialisierung): Es sind ein Startpunkt x_0, ein Trust-Region Radius $\Delta_0 > 0$, die Konstanten $0 < \gamma_0 \leq \gamma_1 < 1 \leq \gamma_2$, $\gamma > 0$, $0 < \eta_1 < \eta_2 < 1$ sowie die Mengen $\{\mathcal{I}_j\}_{j=1}^{p}$ gegeben.
Berechne $c_0 = c\,(x_0)$ und θ_0, setzte $k = 0$ und initialisiere einen boolschen Wert RESTRICT mit 0 sowie den Filter als leere Menge.

FTR-Algorithmus

Schritt 1 (Initialisierung): Es sind ein Startpunkt x_0, ein Trust-Region Radius $\Delta_0 > 0$, die Konstanten $0 < \gamma_0 \leq \gamma_1 < 1 \leq \gamma_2$, $\gamma > 0$, $0 < \eta_1 < \eta_2 < 1$ sowie die Mengen $\{\mathcal{I}_j\}_{j=1}^{p}$ gegeben.
Berechne $c_0 = c\left(x_0\right)$ und θ_0, setzte $k = 0$ und initialisiere einen boolschen Wert RESTRICT mit 0 sowie den Filter als leere Menge.

Schritt 2 (Test auf Optimalität): Falls $\theta_k = 0$ oder $\|\nabla f\left(x_k\right)\| = 0$, STOP

FTR-Algorithmus

Schritt 1 (Initialisierung): Es sind ein Startpunkt x_0, ein Trust-Region Radius $\Delta_0 > 0$, die Konstanten $0 < \gamma_0 \leq \gamma_1 < 1 \leq \gamma_2$, $\gamma > 0$, $0 < \eta_1 < \eta_2 < 1$ sowie die Mengen $\{\mathcal{I}_j\}_{j=1}^{p}$ gegeben.
Berechne $c_0 = c(x_0)$ und θ_0, setzte $k = 0$ und initialisiere einen boolschen Wert RESTRICT mit 0 sowie den Filter als leere Menge.

Schritt 2 (Test auf Optimalität): Falls $\theta_k = 0$ oder $\|\nabla f(x_k)\| = 0$, STOP

Schritt 3 (Bestimmen des Testpunkts): Berechne einen Schritt s_k, der, falls RESTRICT$=1$ ist, die Bedingung $\|s_k\|_2 \leq \Delta_k$ erfüllen muss. Berechne den Testpunkt $x_k^+ = x_k + s_k$.

FTR-Algorithmus

Schritt 4 (Bestimmen der Abweichung im Testpunkt):

Berechne $c\left(x_k^+\right)$ und $\theta_k^+ = \theta\left(x_k^+\right)$. Bestimme $\rho_k = \frac{f(x_k) - f\left(x_k^+\right)}{m_k(x_k) - m_k\left(x_k^+\right)}$.

FTR-Algorithmus

Schritt 4 (Bestimmen der Abweichung im Testpunkt):
Berechne $c\left(x_k^+\right)$ und $\theta_k^+ = \theta\left(x_k^+\right)$. Bestimme $\rho_k = \frac{f(x_k) - f\left(x_k^+\right)}{m_k(x_k) - m_k\left(x_k^+\right)}$.

Schritt 5 (Überprüfen, ob der Testschritt akzeptiert wird):

- Falls x_k^+ vom Filter akzeptiert wird:
  Setzte $x_{k+1} = x_k^+$, setzte RESTRICT$= 0$, und füge θ_k^+ zum Filter hinzu, wenn entweder $\rho_k < \eta_1$ gilt oder $\|s_k\|_2 \leq \Delta_k$ nicht erfüllt wird.
- Falls x_k^+ nicht vom Filter akzeptiert wird:
  Ist $\|s_k\|_2 \leq \Delta_k$ erfüllt und gilt $\rho_k \geq \eta_1$, setzte $x_{k+1} = x_k^+$ und RESTRICT$= 0$, andernfalls setzte $x_{k+1} = x_k$ und RESTRICT$= 1$.

FTR-Algorithmus

Schritt 6 (Aktualisierung des Trust-Region Radius): Ist $\|s_k\|_2 \leq \Delta_k$, wird der Trust-Region Radius folgendermaßen angepasst:

$$\Delta_{k+1} \in \begin{cases} [\gamma_0 \Delta_k, \gamma_1 \Delta_k] & \textit{falls} \quad \rho_k < \eta_1 \\ [\gamma_1 \Delta_k, \Delta_k] & \textit{falls} \quad \rho_k \in [\eta_1, \eta_2) \\ [\Delta_k, \gamma_2 \Delta_k] & \textit{falls} \quad \rho_k \geq \eta_2 \end{cases}$$

andernfalls setzte $\Delta_{k+1} = \Delta_k$, $k = k + 1$ und gehe zu Schritt 1.

Bestimmen des Testschrittes s_k

Die Geschwindigkeit und die Anzahl der benötigten Iterationen des Algorithmus hängt sehr stark von der verwendeten Modellfunktion und der Methode deren Minimierung ab.
Wünschenswerte Eigenschaften der Modellfunktionen:

- genaue Approximation der Zielfunktion
- einfach, d.h. schnell, zu minimieren

Modellfunktionen

Verwendete Modellfunktionen, wobei g der Gradient, J die Jaccobi-Matrix und H die Hessematrix der Funktion $f(x)$ im Punkt x_k ist:

- Lineare Approximation der Zielfunktion:

$$m_1(x_k + s) = f(x_k) + \langle g, s \rangle$$

Modellfunktionen

Verwendete Modellfunktionen, wobei g der Gradient, J die Jaccobi-Matrix und H die Hessematrix der Funktion $f(x)$ im Punkt x_k ist:

- Lineare Approximation der Zielfunktion:

$$m_1(x_k + s) = f(x_k) + \langle g, s \rangle$$

- Quadratische Approximation der Zielfunktion:

$$m_2(x_k + s) = f(x_k) + \langle g, s \rangle + \frac{1}{2}\langle s, Hs \rangle$$

Modellfunktionen

- Lineare Approximation der einzelnen Funktionen $c_i(x)$:

$$m_3(x_k + s) = \frac{1}{2} \sum_{i=1}^{p} \|c_{I_i}(x_k) + J_{I_i} s\|_2^2$$

Modellfunktionen

- Lineare Approximation der einzelnen Funktionen $c_i(x)$:

$$m_3\left(x_k + s\right) = \frac{1}{2} \sum_{i=1}^{p} \left\| c_{I_i}\left(x_k\right) + J_{I_i} s \right\|_2^2$$

- Quadratische Approximation der einzelnen Funktionen $c_i(x)$:

$$m_4\left(x_k + s\right) = m_3\left(x_k\right) + \frac{1}{2} \sum_{i=1}^{p} \sum_{j \in I_i} c_{I_i}\left(x_k\right) \left\langle s, H_{c_j} s \right\rangle$$

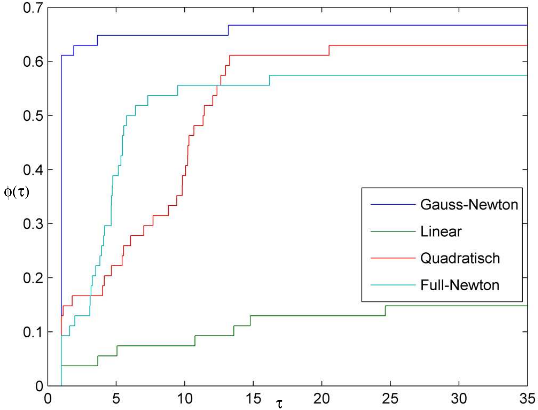
Vergleich der Modellfunktionen

- Tabellarische Übersicht:

Modell-funktion	gelöste Beispiele	am schnellsten gelöst	Abbruch wegen		
			Zeit	Iterationen	Fehler
Gauss-Newton	38	33	3	4	9
linear	15	2	2	27	10
Full-Newton	32	5	0	0	22
quadratisch	37	7	3	2	12

Vergleich der Modellfunktionen

- Leistungsprofil:



Aktualisierung des TR-Radius

- Einfache Möglichkeit für den Aktualisierungsschritt des Trust-Region-Radius ($0 < \gamma_1 < 1 < \gamma_2$):

$$\Delta_{k+1}^1 = \begin{cases} \gamma_1 \cdot \Delta_k & \textit{falls} \quad \rho_k < \eta_1 \\ \Delta_k & \textit{falls} \quad \rho_k \in [\eta_1, \eta_2) \\ \gamma_2 \cdot \Delta_k & \textit{falls} \quad \rho_k \geq \eta_2 \end{cases}$$

Aktualisierung des TR-Radius

- Einfache Möglichkeit für den Aktualisierungsschritt des Trust-Region-Radius ($0 < \gamma_1 < 1 < \gamma_2$):

$$\Delta_{k+1}^1 = \begin{cases} \gamma_1 \cdot \Delta_k & \textit{falls} \quad \rho_k < \eta_1 \\ \Delta_k & \textit{falls} \quad \rho_k \in [\eta_1, \eta_2) \\ \gamma_2 \cdot \Delta_k & \textit{falls} \quad \rho_k \geq \eta_2 \end{cases}$$

- Aktualisierung des Trust-Region-Radius unter Berücksichtigung der tatsächlichen Länge des Testschrittes:

$$\Delta_{k+1}^2 = \begin{cases} \max\left(\gamma_1 \left\| s_k \right\|, \Delta_k\right) & \textit{falls} \quad \rho_k < \eta_1 \\ \Delta_k & \textit{falls} \quad \rho_k \in [\eta_1, \eta_2) \\ \gamma_2 \left\| s_k \right\| & \textit{falls} \quad \rho_k \geq \eta_2 \end{cases}$$

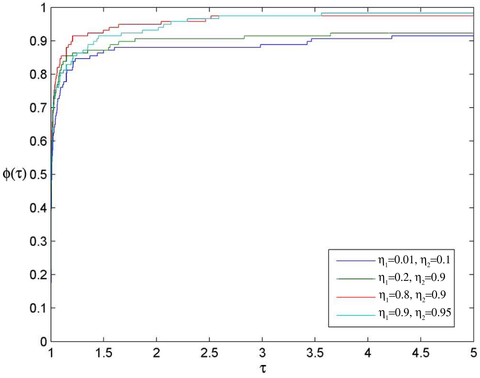
Parametertest für η_1 und η_2

- Auszug aus der tabellarischen Übersicht:

Parameterwerte		gelöste	am schnellsten	Abbruch wegen		
η_1	η_2	Beispiele	gelöst	Zeit	Iterat.	Fehler
0.01	0.1	109	32	0	6	2
0.2	0.9	110	38	0	5	2
0.8	0.9	114	42	0	1	2
0.9	0.95	115	37	0	0	2

Parametertest für η_1 und η_2

- Leistungsprofil:



Parametertest für γ_1, γ_2 und Δ_{k+1}

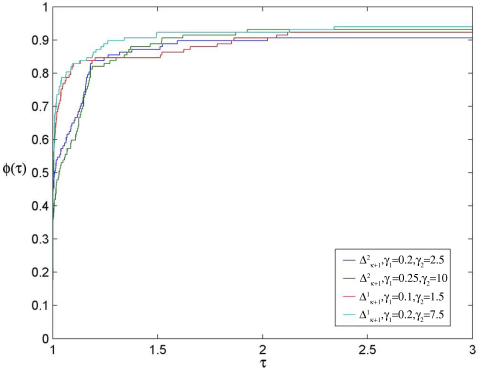
- Tabellarische Übersicht:

Parameter-einstellung	gelöste Beispiele	am schnellsten gelöst	Abbruch wegen Zeit	Abbruch wegen Iterationen	Abbruch wegen Fehler
1)	106	51	0	6	5
2)	110	37	0	5	2
3)	109	58	0	6	2
4)	110	45	0	5	2

Dabei entspricht 1): $\Delta_{k+1} = \Delta^2_{k+1}$, $\gamma_1 = 0.2$ und $\gamma_2 = 2.5$;
2): $\Delta_{k+1} = \Delta^2_{k+1}$, $\gamma_1 = 0.25$ und $\gamma_2 = 10$;
3): $\Delta_{k+1} = \Delta^1_{k+1}$, $\gamma_1 = 0.1$ und $\gamma_2 = 1.5$;
4): $\Delta_{k+1} = \Delta^1_{k+1}$, $\gamma_1 = 0.2$ und $\gamma_2 = 7.5$.

Parametertest für γ_1, γ_2 und Δ_{k+1}

- Leistungsprofil:



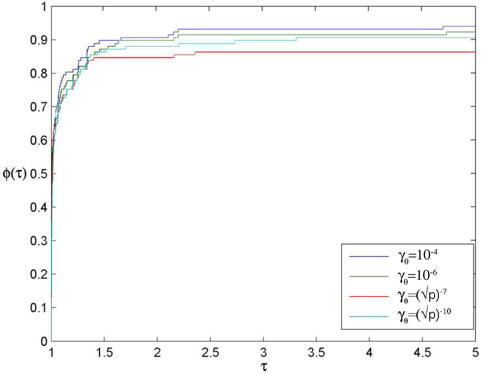
Parametertest für γ_θ

- Tabellarische Übersicht:

Parameter-einstellung	gelöste Beispiele	am schnellsten gelöst	Abbruch wegen Zeit	Abbruch wegen Iterationen	Abbruch wegen Fehler
$\gamma_\theta = 10^{-4}$	110	34	0	5	2
$\gamma_\theta = 10^{-6}$	108	35	0	7	2
$\gamma_\theta = \sqrt{p}^{-7}$	102	27	0	11	4
$\gamma_\theta = \sqrt{p}^{-10}$	106	20	0	9	2

Parametertest für γ_θ

- Leistungsprofil:



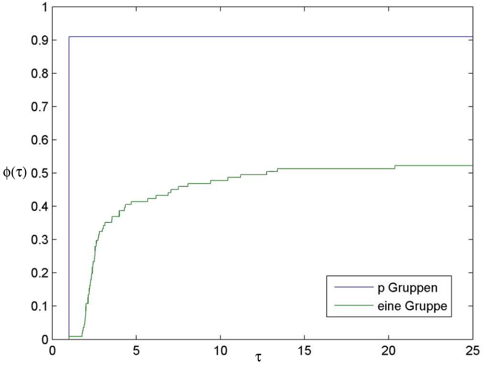
Gruppeneinteilung

- Tabellarische Übersicht:

Einteilung der Funktionen	gelöste Beispiele	am schnellsten gelöst	Abbruch wegen Zeit	Abbruch wegen Iterationen	Abbruch wegen Fehler
in p Gruppen	102	101	0	7	2
in eine Gruppe	58	1	0	15	38

Gruppeneinteilung

- Leistungsprofil:



Exemplarische Ergebnisse

- Keine Modellfunktion ist für alle Beispiele geeignet; im Durchschnitt erzielt jedoch das Gauss-Newton-Modell die besten Ergebnisse.

Exemplarische Ergebnisse

- Keine Modellfunktion ist für alle Beispiele geeignet; im Durchschnitt erzielt jedoch das Gauss-Newton-Modell die besten Ergebnisse.
- Aufgrund der durchgeführten Test erscheinen folgende Parametereinstellungen als geeignet: $\eta_1 = 0.9$, $\eta_2 = 0.95$, $\gamma_1 = 0.2$, $\gamma_2 = 7.5$ bei Verwendung des einfachen Aktualisierungsschrittes $\Delta_{k+1} = \Delta^1_{k+1}$ sowie $\gamma_\theta = 10^{-4}$.