

Vergleich mit linearem Gleichungssystem

$$\underline{A} \cdot \underline{x} = \underline{b}$$

mit dem Residual $\underline{r}(\underline{x}) = \underline{A} \cdot \underline{x} - \underline{b}$

Das ist ein Spezialfall mit dem das nichtlineare Problem approximiert wird.

Eine Lösungsstrategie ist die Herste (Lineare Fall):

$$\underline{x}_{i+1} = \underline{x}_i - \underline{P}^{-1} \cdot (\underline{A} \cdot \underline{x}_i - \underline{b})$$

Man kann Preconditioning (vgl. Eigenwert) im Prinzip Approximation des Inversen von \underline{A} (Wird z.T. f. jeden Schritt aktualisiert).

Beispiel für Verfahren dieser Art: Jacobi, Gauss-Seidel ...

Bsp. Gauss-Seidel :

$$\underline{A} = \underbrace{\underline{L}}_{\text{untere Dreiecksmatrix}} + \underline{D} + \underbrace{\underline{U}}_{\text{obere Dreiecksmatrix}}$$

Ausgangspunkt $\underline{L} \cdot \underline{x} + \underline{D} \cdot \underline{x} + \underline{U} \cdot \underline{x} = \underline{b}$

$$\rightarrow \underline{L} \cdot \underline{x}_{n+1} + \underline{D} \cdot \underline{x}_{n+1} + \underline{U} \cdot \underline{x}_n = \underline{b}$$

$$\parallel \underline{D} \cdot \underline{x}_{n+1} = \underline{b} - \underline{L} \cdot \underline{x}_{n+1} - \underline{U} \cdot \underline{x}_n \parallel \leftarrow \text{wird in jedem Schritt schon}$$

$$(\underline{D} + \underline{L}) \cdot \underline{x}_{n+1} = \underline{b} - \underline{U} \cdot \underline{x}_n$$

$$\underline{x}_{n+1} = -(\underline{D} + \underline{L})^{-1} (\underline{U} \cdot \underline{x}_n - \underline{b})$$

Problem: sequenzielles Verfahren schlecht parallelisierbar.

Jacobi :

Ausgangspunkt $\underline{L} \cdot \underline{x} + \underline{D} \cdot \underline{x} + \underline{U} \cdot \underline{x} = \underline{b}$

$$\Rightarrow \text{umform} \quad \underline{D} \cdot \underline{x}_{n+1} = \underline{b} - \underline{L} \cdot \underline{x}_n - \underline{U} \cdot \underline{x}_n$$

$$\Rightarrow \underline{x}_{n+1} = \underline{D}^{-1} \cdot (\underline{b} - (\underline{L} + \underline{U}) \cdot \underline{x}_n)$$

Richtverfahren

$$\underline{A} \cdot \underline{x} + \underline{x} = \underline{b} + \underline{x}$$

$$\underline{x}_{n+1} = \underline{b} - (\underline{A} - \underline{I}) \cdot \underline{x}_n$$

Nur auf den ersten Blick! man ist zu klein, wie aufgeschrieben, hängt die i-te Zeile nur von den i-1 Zeilen ab!

Nichtlineare Solver

Übersicht über verschiedene Verfahren! Adaption oft auch die Methode
and kombiniert.

Line Search

Die Idee ist die folgende: Wir haben ein Ausgangspunkt x_i

$$x_i \xrightarrow{\mu d_i}$$

Wir suchen $\lambda \approx \min_{\mu > 0} f(x_i + \mu d_i)$

Wobei f z. B. $\|v(\cdot)\|_2^2$ ist oder
eine andere an das Problem angepasste Funktion

- Viele Solver mit Problem in Konvergenzverhalten können durch Line Search zur Konvergenz gebracht werden.
- Die Richtung d wird in der Regel in jedem Schritt angepasst.
- Soloviel einfacher als die komplette Eliminierung denn $\|v\|_2^2$ ist nicht v , sondern $\sum_{j=1}^n v_j^2(x)$, daher ist eine Möglichkeit λ nach verschoben ist Schema zum Beispiel

$$\mu_j = \tau \mu_{j-1}$$

Wie kann das Residuum minimiert werden?

Störte Methode

$$g \cdot v(x + \lambda g) = \frac{d f(x + \lambda g)}{d \lambda}$$

z. B. $\lambda_{i-1} = 0$ wäre Startwert
 λ_0

$$\text{für } (i=0; i < n-1; i++)$$

$$\left\{ \begin{array}{l} \lambda_{i+1} = \lambda_i - \frac{g \cdot v(x + \lambda_i g) (\lambda_i - \lambda_{i-1})}{g \cdot v(x + \lambda_i g) - g \cdot v(x + \lambda_{i-1} g)} \end{array} \right.$$



Das ist ein critical point line search. Zur Erreichung $r(x) = F(x) - b$
Mittleres Richardson Verfahren

Ein alter Bekannter im Prinzip ist es steepest Descent!
 Idee für den ersten Schritt

$$d = -r(x_i) \leftarrow \text{Ein Schritt in die negative Richtung des Residuos. (Annahme } \nabla \approx \text{Id)}$$

$$x_{i+1} = x_i + \lambda d$$

Bei langsamem Konvergenz wird Preconditioning eingesetzt:
 z.B. $M()$

$$d = M(r, x_i) - x_i$$

$$x_{i+1} = x_i + \lambda d$$

λ mit Line Search best.!

Newton-Krylov Methode

$$\otimes d = J(x_i)^{-1} \cdot r(x_i)$$

(Der Jacobian ist nur durch Approximation gelöst, durch Krylov Methode)

$$x_{i+1} = x_i + \lambda d$$

λ wird über Line Search best.

Ein der Arbeitspunkte im Bereich der Lösung der Mittelwert Problem mit vielen Variablen. \otimes Folgt aus $d^T \cdot r(x+\lambda d) = \frac{d^T r(x+\lambda d)}{d^T d} \approx d^T J(x+\lambda d) d$

Quasi Newton

Ähnlich verwendet Änderung im Residuo und der Lösung um näherungsweise Lsg. des Jacobian zu bestimmen

$$K \approx J^{-1} \text{ (Problem } K \text{ dicht besetzte Matrix } \Rightarrow \text{ viel Speicher)}$$

$$x_{i+1} = x_i - \lambda K_i \cdot (K_{i-1} s_{i-1} \dots s_{i-m}, y_{i-1} \dots y_{i-m}) \cdot r(x_i) \quad \lambda \text{ als Line Search}$$

$$s_i = x_{i+1} - x_i$$

$$y_i = r(x_{i+1}) - r(x_i)$$

Beispiel für K_i : Die L-BFGS Methode

L- BFGS Methode zur Berechnung von \underline{k}_i :

$$\underline{d}_1 = \underline{r}(\underline{x}_i)$$

for $(k=i-1, k \leq i-m) \text{ (m)}$

$$\alpha_{1k} = \frac{\underline{s}_k \cdot \underline{d}_1}{\underline{y}_k \cdot \underline{s}_k} \quad \left| \leftarrow \text{in wie weit setzt } \underline{d}_1 \text{ nicht in Richtung von } \underline{z}_k, \text{ Abweichung von Diagonalität.}$$

$$\underline{d}_1 = \underline{d}_1 - \alpha_{1k} \underline{y}_k$$

$$\underline{d}_2 = \underline{K}_0 \cdot \underline{d}_1$$

for $(k=i-m, k \leq i-1, k+1)$

$$\beta_k = \frac{\underline{y}_k^T \cdot \underline{d}_2}{\underline{y}_k \cdot \underline{s}_k} \quad \leftarrow \text{in wie weit setzt } \underline{d}_2 \text{ nicht in Richtung von } \underline{z}_k$$

$$\underline{d}_2 = \underline{d}_2 + (\underline{z}_k - \beta_k) \underline{s}_k$$

Ergebnis \underline{d}_2

Nonlinear Conjugate Gradient

- 1) Auch in alten Bekant Optimierungsproblemen, Ideen von conjugate Gradient (langsame Änderung der Richtung...)
- 2) Benötigt in Vektoren und in Spalten als NR, konvergiert aber schneller
- 3) Problem mit nicht symmetrischen Jacobian

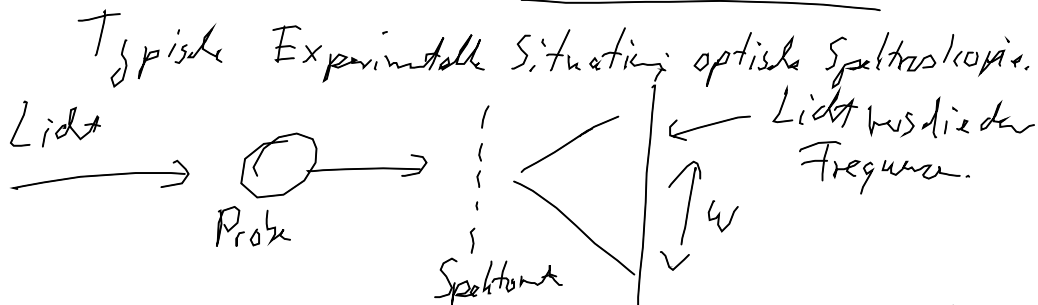
$$\underline{r}_i = \underline{r}(\underline{x}_i)$$

$$\beta_i = \frac{\underline{r}_i \cdot (\underline{r}_i - \underline{r}_{i-1})}{\underline{r}_{i-1} \cdot \underline{r}_{i-1}}$$

$$\underline{s}_i = -\underline{r}(\underline{x}_i) + \beta_i \underline{s}_{i-1} \quad \leftarrow \text{Langsame Änderung der Richtung.}$$

$$x_{i+n} = x_{i+1} \dots$$

VI.14 Berechnung von Spektrum und Fourier Transform (DFT und FFT)



typische vereinfachte Formel $\alpha(\omega) = \frac{\text{Im}(P(\omega))}{E(\omega)}$

OH liegt P , die Polarisation im Sample im Zentrum vor, als Resultat einer komplizierten Simulation!

Dabei ist die schnelle Fourier-Transformierte sehr wichtig.

Allg. Formel

$$P(\omega) = \int_0^{\infty} dt P(t) e^{i\omega t}$$

↑ Einsatz Fourier Transform wegen Kausalität

Problem:

- 1) $P(t)$ meist nicht bis ∞ vorhanden. Die Halbwertzeit ist das $P(t)$ schnell abfällt. (teilw. auf Problem im Exp.)

2) Bereich von $P(t)$ ist oft sehr teuer bei komplexen Systemen, Kompromiss zwischen t_{max} und Artefakt.

3.) Die Zeitdiskretisierung von $P(t)$ gibt maximal mögliche Frequenz von (Nyquistrate)

Verschiedene Algorithmen existieren.

- 1) FFT (Fast Fourier Transformation) (superschnell, limitierte Anwendbar)
- 2) DFT (Discrete Fourier Transformation) (langsam, flexibel)

Alle diese Algorithmen existieren auch für höhere Dimensionen.

Beachte:

$$\begin{aligned} f(k_x, k_y) &= \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy f(x, y) e^{ik_x x + ik_y y} \\ &= \int_{-\infty}^{\infty} dx e^{ik_x x} \underbrace{\int_{-\infty}^{\infty} dy f(x, y) e^{ik_y y}}_{f(x, k_y)} = \int_{-\infty}^{\infty} dx e^{ik_x x} f(x, k_y) \end{aligned}$$