

Das CG-Verfahren

Numerische Mathematik 1
WS 2011/12

Das Energiefunktional

Sei $0 \prec A = A^T \in \mathbb{R}^{n,n}$ eine symmetrische und positiv-definite Matrix. Sei $b \in \mathbb{R}^n$.

Dann ist das CG-Verfahren (angewendet auf A und b) eng verbunden mit dem Energiefunktional $f : \mathbb{R}^n \rightarrow \mathbb{R}$ gegeben durch

$$f(x) := \frac{1}{2}x^T Ax - b^T x \quad \text{für alle } x \in \mathbb{R}^n.$$

Hinreichende Optimalitätsbedingung

Analysis II: Eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ hat in x^* ein lokales Extremum, falls

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

und

$$D^2 f(x^*) \succ 0.$$

Bei obigem Energiefunktional ist

$$\nabla f(x) = Ax - b$$

und

$$D^2 f(x) = A \succ 0,$$

d.h. die Lösung x^* des linearen Gleichungssystems $Ax^* = b$ ist das eindeutige (sogar globale) Extremum von f .

Das Residuum

Somit ist der Gradient an der Stelle $x \in \mathbb{R}^n$ gleich dem Residuum:

$$\nabla f(x) = Ax - b =: r$$

Die Kondition

Ist $0 < A = A^T \in \mathbb{R}^{n,n}$ so gibt es orthogonales $U \in \mathbb{R}^{n,n}$, so dass

$$U^T A U = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

die Diagonalmatrix der Eigenwerte $\lambda_1 \geq \dots \geq \lambda_n > 0$ ist.

Dies ist gleichzeitig eine Singulärwertzerlegung.

Damit ist

$$\kappa_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \lambda_1 \cdot \frac{1}{\lambda_n} = \frac{\lambda_1}{\lambda_n}.$$

Matlab-Code

```
function plot_energy_functional(A, b)

H = 1.5; h = 0.04; x1 = -H:h:H; x2 = -H:h:H;

[X1, X2] = meshgrid(x1, x2);

for i=1:length(x1)
    for j=1:length(x2)
        x = [X1(i,j); X2(i,j)];

        Z(i,j) = (1/2)*(x'*A*x) - b'*x;

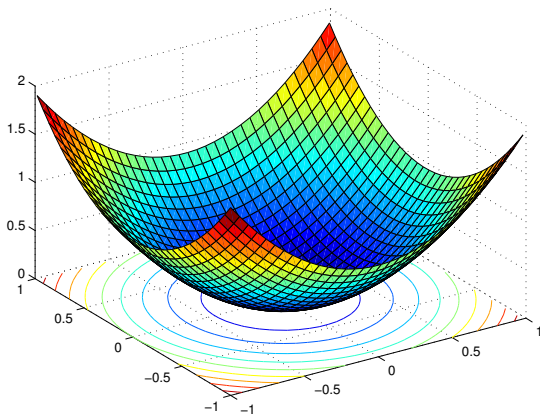
        if( Z(i,j)>3 )
            Z(i,j) = inf;
        end
    end
end

surfc(X1, X2, Z); % und dann contour statt surfc
```

gut konditioniertes Beispiel

>> plot_energy_functional(A, b) für

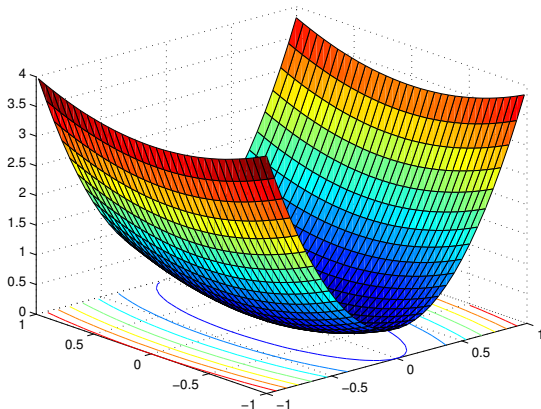
$$A = \begin{bmatrix} 2 & \\ & 1.9 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



schlecht konditioniertes Beispiel

>> plot_energy_functional(A, b) für

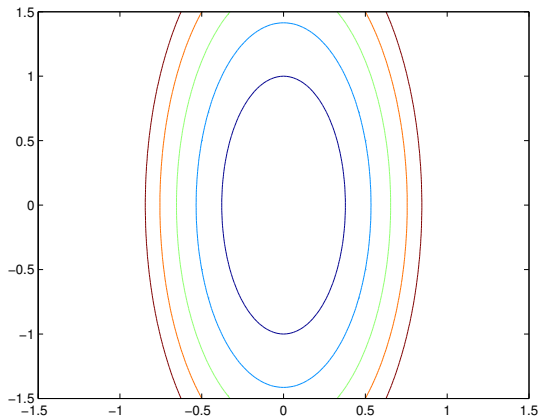
$$A = \begin{bmatrix} 7 & \\ & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Contour-Plot

>> plot_energy_functional(A, b) für

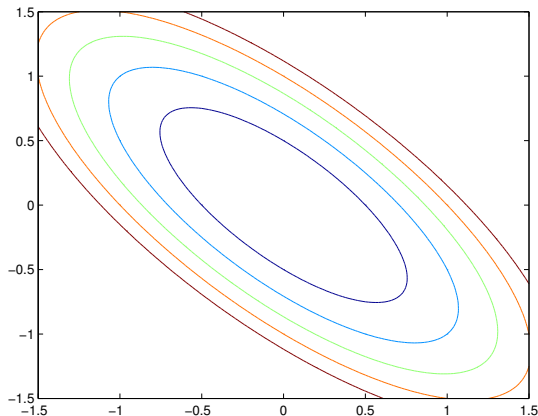
$$A = \begin{bmatrix} 7 & \\ & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Contour-Plot

>> plot_energy_functional(A, b) für

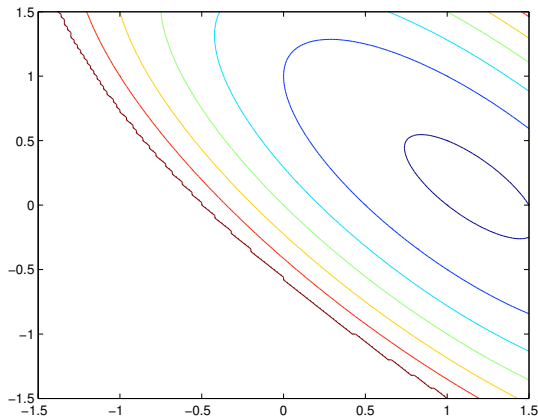
$$A = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix} \sim \begin{bmatrix} 7 & \\ & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Contour-Plot

>> plot_energy_functional(A, b) für

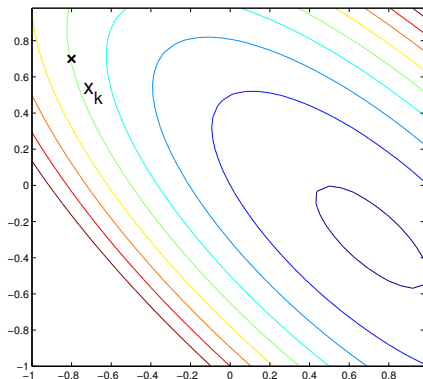
$$A = \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix} \sim \begin{bmatrix} 7 & \\ & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$



Verfahren des steilsten Abstiegs

Ausgehend von der aktuellen Position x_k bewegt man sich in die Richtung des steilsten Abstiegs:

$$-\nabla f(x_k) = -(Ax_k - b)$$



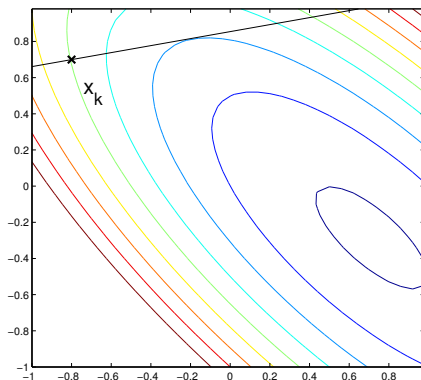
Verfahren des steilsten Abstiegs

Mit der Notation

$$r_k := \nabla f(x_k) = Ax_k - b$$

sucht man also die nächste Iterierte in der Form

$$x_{k+1} = x_k + \alpha_k r_k.$$



Verfahren des steilsten Abstiegs

Um ein x_{k+1} zu finden, sodass $f(x_{k+1})$ möglichst klein wird muss dann

$$\begin{aligned}0 &= r_{k+1}^T r_k \\ &= (Ax_{k+1} - b)^T r_k \\ &= (Ax_k + \alpha_k Ar_k - b)^T r_k \\ &= (r_k + \alpha_k Ar_k)^T r_k = \|r_k\|_2^2 + \alpha_k r_k^T Ar_k\end{aligned}$$

sein, d.h. es muss

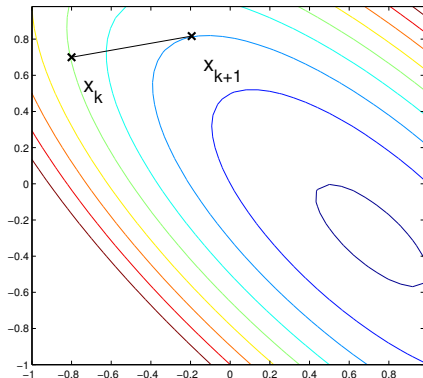
$$\alpha_k = -\frac{\|r_k\|_2^2}{r_k^T Ar_k}$$

sein.

Verfahren des steilsten Abstiegs

Man führt also den Schritt

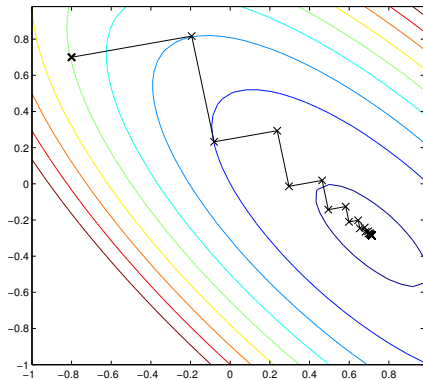
$$x_{k+1} = x_k + \alpha_k r_k, \quad \text{mit} \quad \alpha_k = -\frac{\|r_k\|_2^2}{r_k^T A r_k}.$$



Verfahren des steilsten Abstiegs

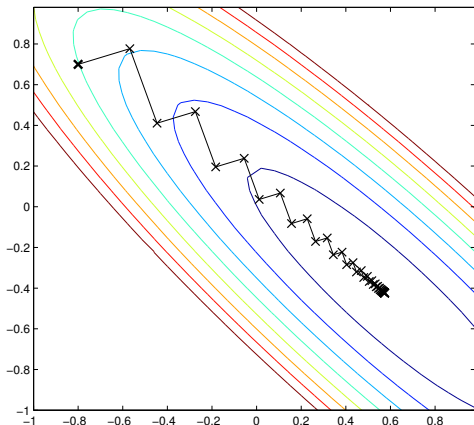
Das Verfahren des steilsten Abstiegs:

$$x_{k+1} = x_k + \alpha_k r_k, \quad \text{mit} \quad \alpha_k = -\frac{\|r_k\|_2^2}{r_k^T A r_k}.$$



Verfahren des steilsten Abstiegs

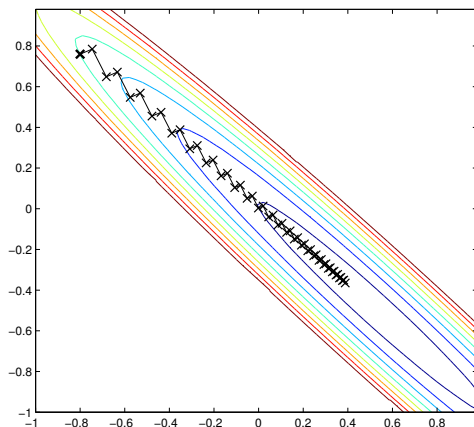
Mit größerer Konditionszahl von A



ungewolltes Zick-Zack-Verhalten

Verfahren des steilsten Abstiegs

Bei noch größerer Konditionszahl von A



ungewolltes Zick-Zack-Verhalten

Fehlerabschätzung

Für das Verfahren des steilsten Abstiegs gilt die Fehlerabschätzung:

$$\|x_k - x^*\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k \|x_0 - x^*\|_A$$

Abhilfe: CG-Verfahren

Wähle Suchrichtungen d_k für die Iteration

$$x_{k+1} = x_k + \alpha_k d_k,$$

die A -konjugiert sind, d.h.

$$d_k^T A d_j = 0,$$

für alle $k \neq j$.

Auch vom Fehler

$$e_k := x_k - x^*,$$

wobei $x^* := A^{-1}b$ die Lösung von $Ax = b$ (d.h. das Minimum von f) ist fordert man A -Konjugiertheit zur vorherigen Suchrichtung

$$d_k^T A e_{k+1} = 0.$$

Herleitung des CG-Verfahrens

... jede Menge komplizierte Formeln ...

wie in der Vorlesung, vgl. auch

[Jonathan Richard Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*]

CG-Verfahren

Der Algorithmus zu $0 \prec A = A^T \in \mathbb{R}^{n,n}$ und $b \in \mathbb{R}^n$:

Setze $r_0 := -b$, $x_0 := 0$ und $k := 0$

(a) Wenn $\|r_k\|_2 \leq \text{maxres}$, so Abbruch

(b) Wenn $\|r_k\|_2 > \text{maxres}$

$$\bullet d_k := \begin{cases} -r_k + \beta_{k-1} d_{k-1}, & \text{mit } \beta_{k-1} := \frac{\|r_k\|_2^2}{\|r_{k-1}\|_2^2} & \text{wenn } k \geq 1 \\ -r_0, & & \text{wenn } k = 0 \end{cases}$$

$$\bullet x_{k+1} := x_k + \alpha_k d_k \quad \text{mit } \alpha_k := \frac{\|r_k\|_2^2}{d_k^T A d_k}$$

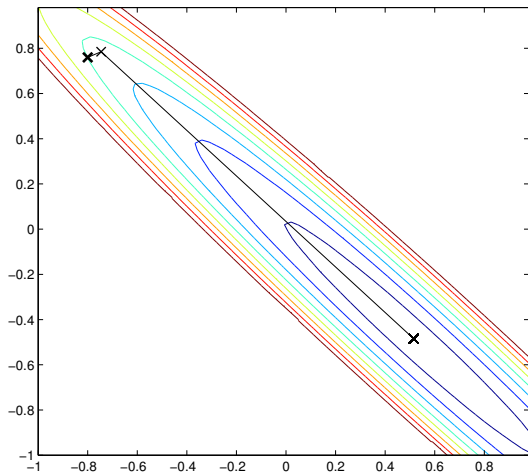
$$\bullet r_{k+1} := r_k + \alpha_k A d_k$$

$$\bullet k := k + 1$$

• Gehe zu (a)

aus [Plato, Numerische Mathematik kompakt]

CG-Verfahren



Fehlerabschätzung

Für das CG-Verfahren gilt die Fehlerabschätzung:

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k \|x_0 - x^*\|_A$$

Abschließende Bemerkungen

- Das CG-Verfahren ermöglicht eine sog. **kurze Rekursion**, d.h. jeder Schritt k ist gleich teuer. Bei GMRES ist das nicht der Fall.
- Die A -orthogonalität der Suchrichtungen geht durch Rundungsfehler verloren.