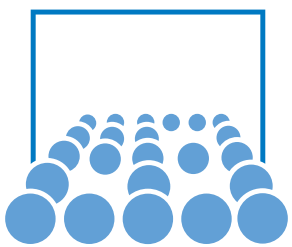
Numerisches Programmieren 2012/13

Christoph Riesinger, Jürgen Bräckle

21. - 24. Januar 2013



Vergleich direkte/iterative Löser

		direkt	iterativ
End-/ Zwischenergebnis		exakte Lösung des Gleichungssystems	Zwischenlösungen $x^{(i)}$
Laufzeit		$O(n^3)$	$O(n^2)$ pro Iteration (Matrix/Vektor-Produkt)
Anwendungsfälle		kleine, dicht besetzte Matrizen	große, dünn besetzte Matrizen
Beispielalgorithmen		Gauß-Elimination LR-Zerlegung Cholesky-Faktorisierung	**Relaxationsverfahren:** Richardson Jacobi Gauß-Seidel **Abstiegsverfahren:** steilster Abstieg konjugierte Gradienten

Begriffe

- **Näherung $x^{(i)}$:**
  Zwischenlösung im i-ten Iterationsschritt.
  Sollte zum Schluss nahe/exakt bei x liegen.
- **Fehler $e^{(i)}$:**
  $e^{(i)} = x - x^{(i)}$
  Abstand der aktuellen Lösung $x^{(i)}$ vom exakten Ergebnis x.

Da x unbekannt, ist auch $e^{(i)}$ unbekannt. Deshalb verwendet man das

- **Residuum $r^{(i)}$:**
  $r^{(i)} = b - Ax^{(i)}$
  b, A und $x^{(i)}$ sind in jedem Schritt bekannt.

Relaxationsverfahren

- Richardson Iteration:

```
for i = 0,1,...
   for k = 1,...,n:
```

$$x_k^{(i+1)} := x_k^{(i)} + r_k^{(i)}$$

- Jacobi Iteration:

```
for i = 0,1,...
   for k = 1,...,n:
   for k = 1,...,n:
```

$$y_k := \frac{1}{a_{kk}} \cdot r_k^{(i)}$$
$$x_k^{(i+1)} := x_k^{(i)} + y_k$$

- Gauss-Seidel Iteration:

```
for i = 0,1,...
   for k = 1,...,n:
```

$$r_k^{(i)} := b_k - \sum_{j=1}^{k-1} a_{kj} x_j^{(i+1)} - \sum_{j=k}^{n} a_{kj} x_j^{(i)}$$
$$y_k := \frac{1}{a_{kk}} \cdot r_k^{(i)}$$
$$x_k^{(i+1)} := x_k^{(i)} + y_k$$

- SOR method (successive over-relaxation):

$$x_k^{(i+1)} := x_k^{(i)} + \alpha y_k$$

Analyse der Relaxationsverfahren

Vorausstzung(en) für (schnelle) Konvergenz: siehe Foliensatz 6, Seite 11.
Alle vier vorgestellten Relaxationsverfahren basieren auf folgender Idee:

$$A = M + (A - M)$$

mit

- $Mx = b$ soll einfach zu lösen sein
  $\Rightarrow M$ sollte der Einheitsmatrix ähneln
- $A - M$ sollte sehr klein (bzgl. einer Matrixnorm) sein
  $\Rightarrow M$ sollte Ähnlichkeit zu A haben

$$A = L_A + D_A + U_A$$

Richardson	$M := I$
Jacobi	$M := D_A$
Gauß-Seidel	$M := D_A + L_A$
SOR	$M := \frac{1}{\alpha} D_A + L_A$

Abstiegsverfahren

Vorausstzung

$A \in \mathbb{R}^{n \times n}$ ist symmetrisch und positiv definit, d.h. $A = A^T$ und $x^T Ax > 0$.

Grundprinzip

Anstatt ein Lineares Gleichungssystem zu lösen, wird eine mehrdimensionale Funktion minimiert.

$$f(x) = \frac{1}{2} x^T Ax - b^T x + c$$

Das Minimum der Funktion f ist die Lösung des Linearen Gleichungssystems $Ax = b$.

Konkrete Verfahren

- Methode des Steilsten Abstiegs
- Konjugierte Gradienten

Methode des Steilsten Abstiegs

Allgemeines

- In jedem Iterationsschritt wird in Richtung des steilen Abstiegs der Fuktion f gegangen. Dies entspricht dem aktuellen Residuum $r^{(i)}$ bzw. dem Gradienten der Funktion f an der Stelle $x^{(i)}$.
- Optimale Länge, mit der in Richtung $r^{(i)}$ gegangen wird:

$$\alpha_i = \frac{{r^{(i)}}^T r^{(i)}}{{r^{(i)}}^T A r^{(i)}}$$

Algorithmus

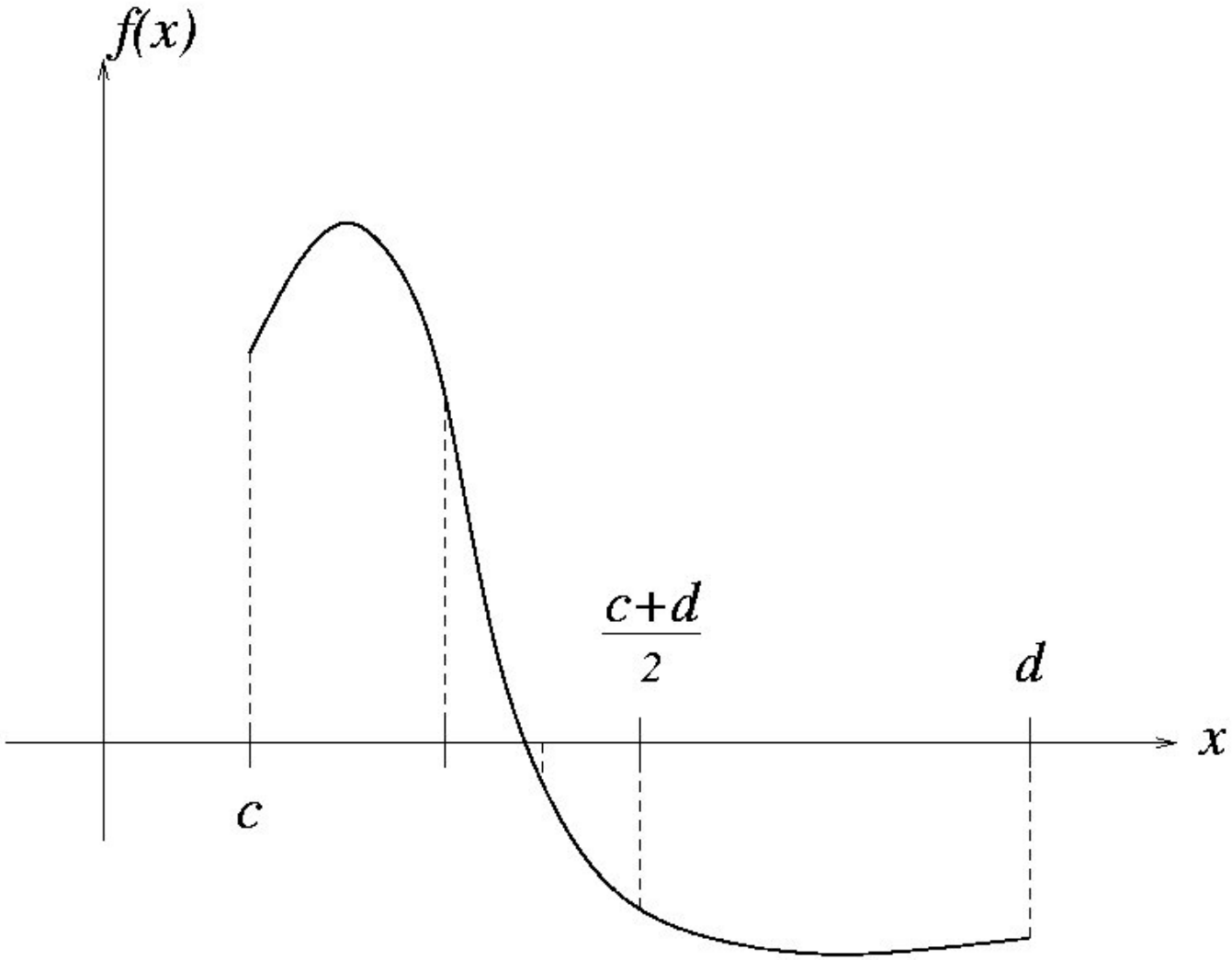
$$\begin{aligned} r^{(i)} &:= b - Ax^{(i)} \\ \alpha_i &:= \frac{{r^{(i)}}^T r^{(i)}}{{r^{(i)}}^T A r^{(i)}} \\ x^{(i+1)} &:= x^{(i)} + \alpha_i r^{(i)} \end{aligned}$$

Konjugierte Gradienten

- Neue Suchrichtungen $d^{(i)}$.
- Der durch $d^{(0)}, \ldots, d^{(i-1)}$ aufgespannte Unterraum sollte orthogonal zu $e^{(i)}$ sein.
- Dann muss die neue Suchrichtung $d^{(i)}$ so gewählt werden, dass sie einerseits orthogonal zu $d^{(0)}, \ldots, d^{(i-1)}$ und andererseits orthogonal zu $e^{(i)}$ ist.
- Problem: Wir kennen $e^{(i)}$ nicht!!!
- Deshalb weichen wir wieder auf die Residuen $r^{(i)}$ aus und konstruieren "A-orthogonale" bzw. "konjugierte" Suchrichtungen anstatt orthogonaler.
- Zwei Vektoren u und v heißen "A-orthogonal" oder "konjugiert", falls

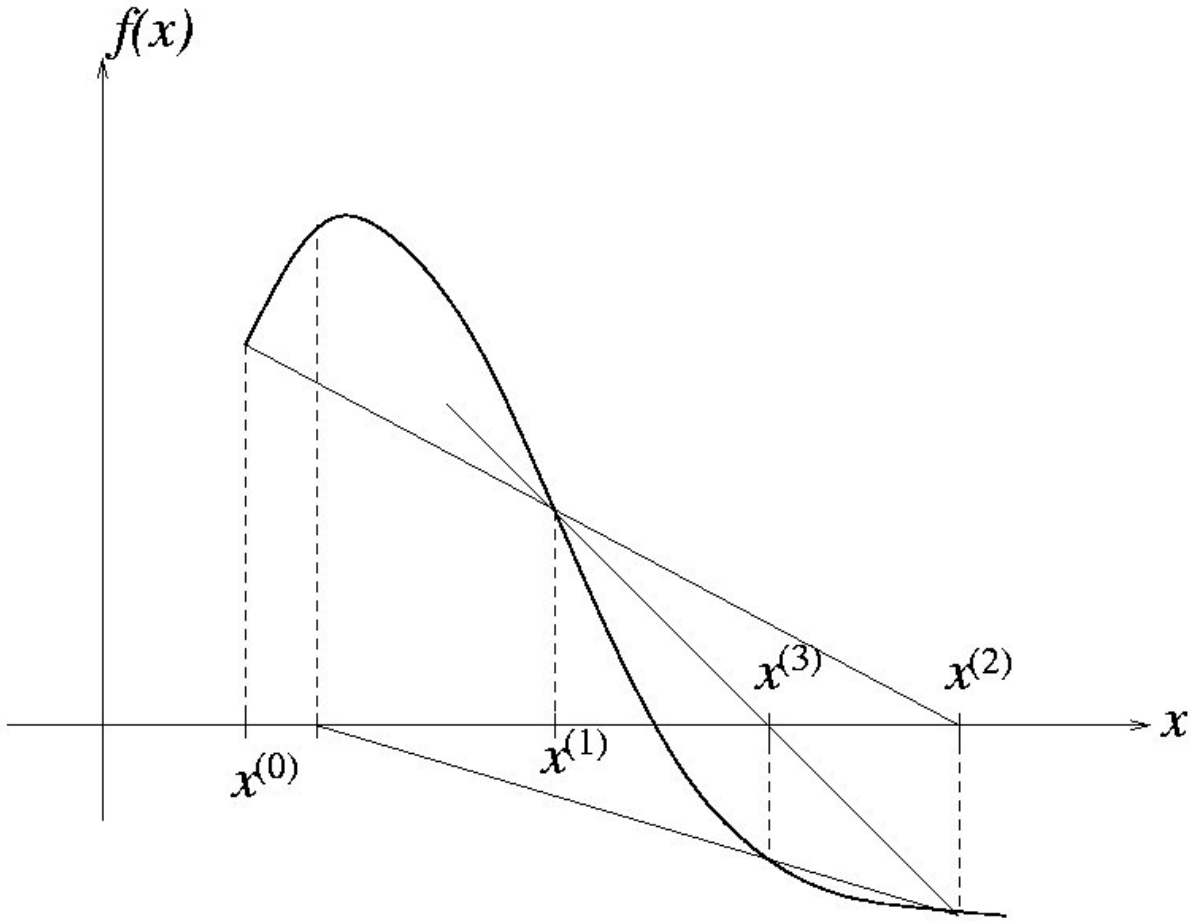
$$u^T A v = 0$$

Bisektionsmethode



Sekantenmethode

$$x^{(i+1)} = x^{(i)} - (x^{(i)} - x^{(i-1)}) \cdot \frac{f(x^{(i)})}{f(x^{(i)}) - f(x^{(i-1)})}$$



Newtonmethode

$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$$