Interdisziplinäres Zentrum
für Wissenschaftliches Rechnen (IWR)
Ruprecht–Karls–Universität Heidelberg

Dr. Peter Zaspel

Übungsblatt 12

11.07.2016 – 18.07.2016

Einführung in die Numerik SS 2016

Aufgabe 1. CG-Algorithmus - Beispiel (3+3 Punkte)
Jetzt sei

$$A = \begin{pmatrix} 1 & \frac{3}{2} \\ \frac{3}{2} & \frac{9}{2} \end{pmatrix}, \qquad b = \begin{pmatrix} 6.5 \\ 16.5 \end{pmatrix}$$

a. Zeigen Sie, dass A positiv definit ist.

b. Bestimmen Sie die Lösung von $Ax = b$ mit Hilfe des CG-Algorithmus zum Startwert $x^{(0)} = (1,1)^T$. Geben Sie auch alle Zwischenresultate $x^{(i)}$, $r^{(i)}$, $p^{(i)}$, α_i, β_i an.

Aufgabe 2. Eigenschaften der näherungsweisen Lösungen des CG-Verfahrens (5+3 Punkte)
Sei $\vec{x}_n$ die n-te Iterierte im CG-Verfahren für ein lineares Gleichungssystem $A\vec{x} = \vec{b}$ mit $A \in \mathbb{R}^{m \times m}$ symmetrisch, positiv definit. Zeigen Sie:

a. Falls das iterative Verfahren ist noch nicht konvergiert ist, d.h. $\vec{r}_{n-1} \neq 0$, ist $\vec{x}_n$ als Minimierer von $\|\vec{x}_\star - \vec{x}_n\|_A$ eindeutig.

b. Die Konvergenz ist monoton, d.h.

$$\|\vec{x}_\star - \vec{x}_n\|_A \leq \|\vec{x}_\star - \vec{x}_{n-1}\|_A \,,$$

und die exakte Lösung $\vec{x}_\star$ wird (in exakter Arithmetik) angenommen für $n' \leq m$ viele Iterationen.

Aufgabe 3. Dimensionalität des Krylov-Unterraumes (6 Punkte)
Zeigen Sie Lemma 5.5.7 aus der Vorlesung, d.h. zeigen Sie die folgende Aussage:

Für den Krylov-Unterraum $\mathcal{K}(A, \vec{v})$ gilt $\dim(\mathcal{K}_m) = m$ genau dann wenn für den Grad μ von $\vec{v}$ $\mu \geq m$ gilt, d.h.

$$\dim(\mathcal{K}_m) = m \Leftrightarrow \mathrm{grad}(\vec{v}) \geq m \,.$$

Hieraus folgt

$$\dim(\mathcal{K}_m) = \min\{m, \mathrm{grad}(\vec{v})\} \,.$$

Hinweis: Sie dürfen die Aussage von Lemma 5.5.6 im Beweis verwenden.

Aufgabe 4. Praktische Aufgabe CG-Verfahren (3+2+5+0 Punkte)
In dieser praktischen Aufgabe sollen Sie ein Programm schreiben, dass das lineare Gleichungssystem wie in Aufgabe 5 von Blatt 10 mittels des CG-Verfahren löst. Als Basis dient die Datei `cg.cpp`, die Sie im Moodle-Kurs herunterladen können.

a. Schreiben Sie eine Funktion `void mat_vec(double* y, double** A, double* x, int n)`, die eine vollbesetzte Matrix `A`$\in \mathbb{R}^{n\times n}$ mit einem Vektor `x` multipliziert und das Resultat in Vektor `y` speichert.

b. Vervollständigen Sie die Funktion `double scalar_product(double* x, double* y, int n)` zur Berechnung des Standard-Skalarprodukts zwischen zwei Vektoren `x`,`y`$\in \mathbb{R}^n$.

c. Implementieren Sie das CG-Verfahren in der Routine `void cg(double** A, double* x, double* b, int n, int iter_max, double thresh)` unter Einsatz von `mat_vec` und `scalar_product`, wobei `A`$\in \mathbb{R}^{n\times n}$ die Systemmatrix, `b` die rechte Seite und `x` der Vektor ist, der zu Beginn die erste Näherung an die Lösung und nach Ablauf des Algorithmus die Näherung an die Lösung enthält. Das CG-Verfahren soll nach maximal `iter_max` Schritten abgebrochen werden. Alternativ wird das Verfahren beendet, wenn die 2-Norm des Residuums unter `thresh` fällt.

d. Die bereits implementierte Hauptroutine setzt das lineare Gleichungssystem auf und löst es mittels des CG-Verfahren.

Abgabe: 18.07.2016, 14:00-14:15 Uhr. (Mappen in der Vorlesung)