Universität Konstanz WS 15/16
Fachbereich Mathematik und Statistik
S. Trenz, Prof. Dr. S. Volkwein

Übungen zu Theorie und Numerik partieller Differentialgleichungen

http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching

Blatt 9

Abgabe: Dienstag, 12.01.2016, in der Vorlesung!

Aufgabe 1 (Theorie) (4 Punkte)

Gegeben sei die Aufgabe

$$\begin{aligned} -\Delta v &= \lambda v \text{ in } \Omega, \\ v|_{\partial\Omega} &= 0. \end{aligned} \tag{1}$$

Eine Lösung $v \in C^2(\Omega) \cap C^0(\bar{\Omega})$, $v \neq 0$, heißt *Eigenfunktion* zum *Eigenwert* λ. Zeigen Sie:

a) Für jeden Eigenwert λ von (1) gilt $\lambda > 0$.

b) Seien v_1, v_2 die Eigenfunktionen zu Eigenwerten λ_1 und λ_2 mit $\lambda_1 \neq \lambda_2$. Dann sind v_1 und v_2 orthogonal bezüglich des Skalarproduktes

$$\langle u, w \rangle = \int_\Omega u(x)w(x)dx.$$

c) Berechnen Sie die Eigenwerte und Eigenfunktionen von (1) im Fall $\Omega =]0,1[^2$ explizit indem Sie hierzu den *Separationsansatz* $v(x_1, x_2) = w(x_1)z(x_2)$ verwenden.

Aufgabe 2 (Theorie) (4 Punkte)

Es sei $A \in \mathbb{R}^{N \times N}$ eine symmetrische und positiv definite Matrix. Die Vektoren $\{d_i\}_{i=1}^{N-1}$ aus $\mathbb{R}^N$ heißen *konjugiert* oder *A-orthogonal*, falls gilt:

$$\begin{aligned} d_i^T A d_j &= 0 \quad \text{für } i \neq j,\ 0 \leq i, j \leq N-1, \\ d_i^T A d_i &\neq 0 \quad \text{für } 0 \leq i \leq N-1. \end{aligned}$$

Zeigen Sie:

a) Die Vektoren $d_0, d_1, \ldots, d_{N-1}$ bilden eine Basis des $\mathbb{R}^N$.

b) Gegeben sei das Gleichungssystem $Ax = b$. Zerlegt man die Lösung $x = A^{-1}b$ nach der Basis $d_0, \dots, d_{N-1}$, so gilt:

$$x = \sum_{k=0}^{N-1} \alpha_k d_k \quad \text{mit} \quad \alpha_k = \frac{d_k^T A x}{d_k^T A d_k}, \; k = 0, \dots, N-1.$$

c) Für jedes $x_0 \in \mathbb{R}^N$ liefert die für $k \geq 0$ durch

$$x_{k+1} = x_k + \alpha_k d_k \quad \text{mit } \alpha_k = \frac{-d_k^T(Ax_k - b)}{d_k^T A d_k}$$

erzeugte Folge nach höchstens N Schritten die Lösung $x_N = A^{-1}b$.

Aufgabe 3 (Matlab) (6 Punkte)

Wir betrachten eine numerisch effiziente Methode zur Lösung von großen linearen Gleichungssystem der Form $Ax = b$, das *konjugierte Gradienten Verfahren* (siehe Algorithmus 1). Dieses stellt insbesondere als *iteratives* Verfahren eine Alternative zu klassischen direkten Lösungsmethoden linearer Gleichungssysteme dar, die z.B. aus der FD-Diskretisierung elliptischer Randwertprobleme resultieren.

Schreiben Sie ein Matlab-File `mycg.m` für die Funktion

```
function [x,normr,niter] = mycg(A,b,x0,tol)
```

die als Übergabewerte **ausschließlich und in dieser Reihenfolge** die Matrix `A`, die rechte Seite `b`, einen Startwert `x0` und eine Toleranz `tol` akzeptiert. Als Rückgabe erhält man die Lösung `x`, sowie die Gesamtanzahl an Iterationen `niter` und einen Residuumsvektor `normr` der Länge `niter`, der als Einträge die Normen der Residuen pro Iteration enthält.

Betrachten Sie nun hierzu die Problemstellung auf Blatt 8, Aufgabe 3, mit $f(x, y) = 20$ und $g(x, y) = y\cos(4\pi x)$ und lösen Sie das lineare Gleichungssystem, indem Sie das CG-Verfahren in einem `main.m`-File für verschiedene $M = 10, 20, 40, 80$ und Toleranzen $\tau = 10^{-1}, 10^{-2}, 10^{-4}, 10^{-6}$ aufrufen. Wählen Sie zunächst als Startwert den Nullvektor $x_0 = \mathbf{0}$.

1. Plotten Sie für jeden Aufruf die erhaltenen Normen der Residuen im Iterationsverlauf (Hinweis: hier könnte der Matlab Plotbefehl `semilogy` hilfreich sein).

2. Schreiben Sie die erhaltene Anzahl an Iterationen in eine Tabelle:

	M			
τ	10	20	40	80
10^{-1}				
10^{-2}				
10^{-4}				
10^{-6}				

3. Lösen Sie das Gleichungssystem $Ax = b$ mittels LU-Zerlegung und dem Backslash-Operator "\". Vergleichen Sie die Differenz der Lösung mit der aus dem CG-Verfahren in der 2-Norm sowie die Rechenzeit unter Verwendung von `tic/toc` für verschiedene M und τ. Was beobachten Sie?

4. Was stellen Sie fest, wenn Sie als Startvektor x_0 nicht den Nullvektor verwenden?

Algorithm 1 Conjugate Gradient Method

Require: $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $x^0 \in \mathbb{R}^n$, Tolerance $\tau \geq 0$
Ensure: A is positive definite
  $k = 0$;
  $r^0 = b - Ax^0$;
  $p^0 = r^0$;
  while $\|r_k\| \geq \tau$ **do**
    $\alpha_k = \frac{(p^k)^T r^k}{(p^k)^T A p^k}$;
    $x^{k+1} = x^k + \alpha_k p^k$;
    $r^{k+1} = r^k - \alpha_k A p^k$;
    $\beta_k = \frac{(Ap^k)^T r^{k+1}}{(Ap^k)^T p^k}$;
    $p^{k+1} = r^{k+1} - \beta_k p^k$;
    $k = k + 1$;
  end while
