Numerische Mathematik 1 Übungen
Blatt 5 – Bearbeitung bis zum 25.11.2014 bzw. 2.12.2014

Hausaufgaben

Bearbeitung bis zum 25.11.2014

1. Finden Sie die Iterationsmatrix $T_{\rm SGS}$ für die Symmetrische Gauß-Seidel Methode. Leiten Sie die Approximierte Inverse $M_{\rm SGS}$ für die Symmetrische Gauß-Seidel Methode. Zeigen Sie, wenn A symmetrisch ist, ist $M_{\rm SGS}$ symmetrisch, während $M_{\rm GS}$ im allgemeinen nicht symmetrisch ist.

2. Zeigen Sie die Äquivalenz der Normen $\|\cdot\|_1$ und $\|\cdot\|_2$. Beweisen Sie den Satz: Für $\boldsymbol{x}^k = \langle (x_1)_k, \dots, (x_n)_k\rangle^{\rm T}, \boldsymbol{x} = \langle x_1, \dots, x_n\rangle^{\rm T} \in \mathbb{R}^n$ gilt

$$\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x} \quad \Leftrightarrow \quad \lim_{k\to\infty} (x_i)_k = x_i, \quad i = 1, \dots, n.$$

Bearbeitung bis zum 4.11.2014

3. Sei bewiesen: Angenommen hat $A \in \mathbb{R}^{n\times n}$ die Eigenwerte $\{\lambda_i\}_{i=1}^n$ mit den entsprechenden Eigenvektoren $\{\boldsymbol{v}^i\}_{i=1}^n$, die eine Basis für $\mathbb{R}^n$ bilden. Zusätzlich gelten $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0$. Für einen beliebigen Startvektor $\boldsymbol{x}^0 = \sum_{i=1}^n \alpha_i \boldsymbol{v}^i$ für die Vektoriteration gelten $\alpha_1 \neq 0$ und $\|\boldsymbol{x}^0\|_\infty = 1$. Dann für die Vektoriteration $\boldsymbol{y}^k = A\boldsymbol{x}^{k-1}$, $\boldsymbol{x}^k = \boldsymbol{y}^k / y^k_{p_k}$ ($|y^k_{p_k}| = \|\boldsymbol{y}^k\|_\infty$), $\mu_k = y^k_{p_{k-1}}$ gelten $\mu_k \to \lambda_1$ und $s_k\boldsymbol{x}^k \to \boldsymbol{v}^1/\|\boldsymbol{v}^1\|_\infty$ ($s_k^2 = 1$) für $k \to \infty$.

   Beweisen Sie Konvergenz der Inversen Vektoriteration unter geeigneten Voraussetzungen.

4. Zur Bestimmung der Householder Transformation P, die für gegebenes $\boldsymbol{x} = \langle x_1, \dots, x_n\rangle^{\rm T}$ und $\sigma = {\rm s}(x_1)\|\boldsymbol{x}\|_2$ erfüllt $P\boldsymbol{x} = -\sigma\hat{\mathbf{e}}^{(1)}$, wird der folgende Algorithmus verwendet:

$$\begin{array}{rclcrclc}
\boldsymbol{v} &=& \boldsymbol{x}/\|\boldsymbol{x}\|_\infty & \quad & \theta &=& \tau u_1 & \quad {\rm s}(t) = \left\{\begin{array}{rl} 1, & t \geq 0 \\ -1, & \text{sonst}\end{array}\right. \\
\tau &=& {\rm s}(x_1)\|\boldsymbol{v}\|_2 & & \sigma &=& \tau\|x\|_\infty & \\
\boldsymbol{u} &=& \boldsymbol{v} + \tau\hat{\mathbf{e}}^{(1)} & & P &=& I - \boldsymbol{u}\boldsymbol{u}^{\rm T}/\theta & \quad P\boldsymbol{x} = -\sigma\hat{\mathbf{e}}^{(1)}
\end{array}$$

   Zeigen Sie, $\theta = \frac{1}{2}\|\boldsymbol{u}\|_2^2$ folgt aus den Definitionen von $\boldsymbol{v}$, τ, $\boldsymbol{u}$ und θ. Zeigen Sie, es gilt $P\boldsymbol{x} = -\sigma\hat{\mathbf{e}}^{(1)}$.

Programmieraufgabe

Alle Kodes sollen an `stephen.keeling@uni-graz.at`
per Email bis zum 30.11.2014 geschickt werden.

1. Das folgende Problem der Reaktion, Konvektion und Diffusion $(r, k, d \geq 0)$

$$\left\{\begin{array}{rcll}
v_t + rv + kv_x &=& d(v_{xx} + v_{yy}), & (x,y) \in (0,1)^2, \quad t > 0 \\
v &=& 0, & x = 0, y = 0, 1, \quad t > 0 \\
v_x &=& 0, & x = 1, \quad t > 0 \\
v &=& v_0, & (x,y) \in (0,1), \quad t = 0
\end{array}\right.$$

   sei auf dem räumlichen Gitter $\{(x_i, y_j) : x_i = ih, y_j = jh, i, j = 0, \dots, N\}$, $h = 1/N$, in $\mathbb{R}^2$ und dem zeitlichen Gitter $\{t^n = n\tau : n = 0, \dots, M\}$, $\tau = 1/M$, so diskretisiert:

$$(I + \tau B)\boldsymbol{V}^{n+1} = \boldsymbol{V}^n, \quad n = 0, \dots, M$$

wobei im $n$ten Zeitschritt die exakte Lösung so approximiert wird: $\{v(x_i, y_j, t^n)\}_{i,j=0}^N \approx \boldsymbol{V}^n$. Hier wird $\boldsymbol{V}^n$ von einem langen Vektor in ein zwei-dimensionales Feld für grafische Darstellung so umgewandelt (überschreiben),

```
N1 = N+1;
V  = reshape(V,N1,N1);
```

und umgekehrt für die Lösung der obigen Systeme,

```
N1N1 = N1*N1;
V    = reshape(V,N1N1,1);
```

Die Anfangswerte $\{v(x_i, y_j, 0)\}_{i,j=0}^N \approx \boldsymbol{V}^0$ sind

```
N4 = round(N/4);
V0 = [zeros(N4,1);ones(N1-2*N4,1);zeros(N4,1)];
V0 = kron(V0,V0');
V0 = reshape(V0,N1N1,1);
```

und der Differentialoperator wird approximiert durch

$$B = \frac{d}{h^2}D + \frac{k}{h}F + rI$$

mit $D_0, D_1 \in \mathbb{R}^{(N+1)\times(N+1)}$, $D \in \mathbb{R}^{(N+1)^2\times(N+1)^2}$

$$D_0 = -I, \quad D_1 = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & D_0 & & & \\ D_0 & D_1 & D_0 & & \\ & \ddots & \ddots & \ddots & \\ & & D_0 & D_1 & D_0 \\ & & & D_0 & D_1 \end{bmatrix}$$

$D_2 \in \mathbb{R}^{(N+1)\times(N+1)}$, $F \in \mathbb{R}^{(N+1)^2\times(N+1)^2}$

$$D_2 = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} D_2 & & \\ & \ddots & \\ & & D_2 \end{bmatrix}$$

Schreiben Sie einen Matlab Kode der Form (`N`$= N$, `M`$= M$, `r`$= r$, `k`$= k$, `d`$= d$, `V`$= \boldsymbol{V}^M$)

```
V = familienv(N,M,r,k,d)
```

zur Lösung der obigen Systeme $(I+\tau B)\boldsymbol{V}^{n+1} = \boldsymbol{V}^n$, $n = 0, \ldots, M$, wobei für jedes n die Methode der Konjugierten Gradienten implementiert werden soll. In jedem Zeitschritt soll die Iteration mit dem aktuellsten $\boldsymbol{V}$ gestartet wird. Dabei soll der Präkonditionierer $D^{\frac{1}{2}}$ verwendet werden, wobei $\mathcal{D} = \mathrm{diag}\{(I+\tau B)_{ii}\}_{i=1}^n$, d.h. $C = \mathcal{D}^{\frac{1}{2}}$ in den Vorlesungsnotizen. Verwenden Sie die effiziente Implementierung auf Seite 23 im

```
http://math.uni-graz.at/keeling/numI_ss09/fr16_book.pdf
```

in der gilt $M = \mathcal{D}^{-1}$. Das Abbruchskriterium in diesem Lehrbuch soll nach den Vorlesungsnotizen angepasst werden. Für welche Werte r, k, d ist Konvergenz der Methode garantiert? Wenn diese Bedingung nicht erfüllt ist, soll der Kode eine andere geeignete Methode verwenden. (Welche?) Stellen Sie die Ergebnisse so grafisch dar:

```
A = eye(N1) + tau*B;
V = V0;
for n=1:M
  V = LoeseSystem(A,V);              % V <- A^(-1) * V
  surf(reshape(V,N1,N1)); axis([1 N1 1 N1 0 1]);
  view([1 1 1]); drawnow; pause(0.1);
end
```