Numerische Mathematik 2, Übungen, WS15/16, Blatt 1

Bearbeitung: Hausaufgaben bis 8.10.2015, Programmieraufgabe bis 15.10.2015

Hausaufgaben

Für die folgenden Hausaufgaben soll ein lineares Gleichungssystem $A\boldsymbol{x} = \boldsymbol{b}$, $n \in \mathbb{N}$, $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$, iterativ gelöst werden. Die Matrix A sei durch $A = M + N$ additiv zerlegt, wobei lineare Gleichungssysteme mit der Matrix M leichter zu lösen sind als mit der Matrix A. Durch diese Zerlegung ergibt sich das iterative Verfahren $M\boldsymbol{x}^{k+1} = \boldsymbol{b} - N\boldsymbol{x}^k$ oder $\boldsymbol{x}^{k+1} = T\boldsymbol{x}^k + \boldsymbol{c}$ mit $\boldsymbol{c} = M^{-1}\boldsymbol{b}$ und der Iterationsmatrix $T = -M^{-1}N$.

1. Ein Signal f soll durch die Lösung u des folgenden Randwertproblems entrauscht werden. Das Randwertproblem

$$-\mu u''(x) + u(x) = f(x), \quad x \in (0,1), \quad u'(x) = 0, \quad x \in \{0,1\}, \quad \mu > 0$$

sei durch finite Differenzen folgendermaßen diskretisiert: $m \in \mathbb{N}$, $h = 1/m$, $x_i = ih$, $i = 0, \ldots, m$, $u_i \approx u(x_i)$, $f_i = f(x_i)$,

$$-\frac{\mu}{h}\left[\frac{u_{i+1} - u_i}{h} - \frac{u_i - u_{i-1}}{h}\right] + u_i = f_i, \quad i = 1, \ldots, m-1$$

$$-\frac{\mu}{h}\left[\frac{u_{i+1} - u_i}{h}\right] + u_i = f_i, \quad i = 0, \quad -\frac{\mu}{h}\left[-\frac{u_i - u_{i-1}}{h}\right] + u_i = f_i, \quad i = m$$

Für die sogenannte *lexikografische Reihenfolge* $\boldsymbol{u} = \langle u_0, u_1, \ldots, u_{m-1}, u_m\rangle^\top$ und $\boldsymbol{f} = \langle f_0, f_1, \ldots, f_{m-1}, f_m\rangle^\top$ schreiben Sie das letzte lineare Gleichungssystem in der Form $A\boldsymbol{u} = \boldsymbol{f}$ ($A \in \mathbb{R}^{n\times n}$, $n = m+1$). Für die Jacobi Iteration zur Lösung dieses Systems geben Sie die Matrizen M_J und N_J mit $A = M_\mathrm{J} + N_\mathrm{J}$ explizit an. Zeigen Sie, der Spektralradius $\rho(T_\mathrm{J})$ der Iterationsmatrix $T_\mathrm{J} = -M_\mathrm{J}^{-1}N_\mathrm{J}$ erfüllt $\rho(T_\mathrm{J}) < 1$, $\forall m \in \mathbb{N}$, $\forall \mu > 0$. (Hinweis: Der Spektralradius erfüllt $\rho(A) \le \|A\|$ für jede induzierte Norm $\|\cdot\|$.)

2. Das Randwertproblem im letzten Beispiel soll jetzt mit einer Gauß-Seidel Iteration gelöst werden. Für die sogenannte *rot-schwarz Reihenfolge* $\boldsymbol{u} = \langle u_0, u_2, \ldots, u_{m_1}, u_1, u_3, \ldots, u_{m_2}\rangle^\top$ und $\boldsymbol{f} = \langle f_0, f_2, \ldots, f_{m_1}, f_1, f_3, \ldots, f_{m_2}\rangle^\top$ $m_1 = \max\{2k \le m : k \in \mathbb{N}\}$, $m_2 = \max\{2k+1 \le m : k \in \mathbb{N}\}$, schreiben Sie das lineare Gleichungssystem in der Form $A\boldsymbol{u} = \boldsymbol{f}$. Für die entsprechende Gauß-Seidel Iteration zur Lösung dieses Systems geben Sie die Matrizen M_GS und N_GS mit $A = M_\mathrm{GS} + N_\mathrm{GS}$ explizit an. Zeigen Sie, der Spektralradius $\rho(T_\mathrm{GS})$ der Iterationsmatrix $T_\mathrm{GS} = -M_\mathrm{GS}^{-1}N_\mathrm{GS}$ erfüllt $\rho(T_\mathrm{GS}) < 1$, $\forall m \in \mathbb{N}$, $\forall \mu > 0$.

3. Für die additive Zerlegung $A = M + N$ zeigen Sie, die Iterationsmatrix $T = -M^{-1}N$ erfüllt $T = [I - M^{-1}A]$. Da $\rho(T) < 1$ erfüllt werden soll, wird M^{-1} eine *Approximierte Inverse* genannt. Leiten Sie die Approximierte Inverse M_SGS für die symmetrische Gauß-Seidel Methode her:

$$\sum_{j<i} a_{ij}x_j^{k+\frac{1}{2}} + a_{ii}x_i^{k+\frac{1}{2}} + \sum_{j>i} a_{ij}x_j^k = b_i, \quad i = 1, \ldots, n$$

$$\sum_{j<i} a_{ij}x_j^{k+\frac{1}{2}} + a_{ii}x_i^{k+1} + \sum_{j>i} a_{ij}x_j^{k+1} = b_i, \quad i = 1, \ldots, n$$

Zeigen Sie, wenn A symmetrisch ist, ist M_SGS symmetrisch, während M_GS für Gauß-Seidel im allgemeinen nicht symmetrisch ist.

Programmieraufgabe

Alle Codes sollen an
`stephen.keeling@uni-graz.at`
mit Betreff
Num2 Programmieraufgabe
per Email bis zum 15.10.2015 geschickt werden.

Für $N > 8$ seien verrauschte Daten $\boldsymbol{v} = \{v_{i,j}\}_{i,j=0}^N$ gegeben, d.h. $\boldsymbol{v} = \boldsymbol{u}^\star +$ Rauschen, und $\boldsymbol{u}^\star$ ist eine zu rekonstruierende Funktion. Diese Daten werden als Bilder betrachtet, und die Einträge eines Bildes, z.B. $\boldsymbol{v}$, werden als $(N+1)\times(N+1)$-Feld oder als $(N+1)^2$-Vektor mit der lexikografischen Reihenfolge der Einträge gespeichert:

$$\langle v_{0,0}, \ldots, v_{N,0}, v_{0,1}, \ldots, v_{N,1}, \ldots, v_{0,N}, \ldots, v_{N,N}\rangle^{\mathrm{T}}$$

Ein $(N+1)\times(N+1)$ Feld `v` wird in einen $(N+1)^2$-Vektor so umgewandelt (überschrieben):

```
N1   = N+1; N1N1 = N1*N1;
v    = reshape(v,N1N1,1);
```

und ein $(N+1)^2$-Vektor wird in ein $(N+1)\times(N+1)$ Feld `v` so umgewandelt (überschrieben):

```
v = reshape(v,N1,N1);
```

Für $\mu > 0$, $D_0 = -I$,

$$D_1 = \begin{bmatrix} 2 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 3 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 3 \end{bmatrix}$$

$D_0, D_1, D_2 \in \mathbb{R}^{(N+1)\times(N+1)}$ und

$$D = \begin{bmatrix} D_1 & D_0 & & & & & \\ D_0 & D_2 & D_0 & & & & \\ & D_0 & D_2 & D_0 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & D_0 & D_2 & D_0 & \\ & & & & D_0 & D_2 & D_0 \\ & & & & & D_0 & D_1 \end{bmatrix} \in \mathbb{R}^{(N+1)^2\times(N+1)^2}$$

schreiben Sie einen Matlab Code der Form (Familienname = `familien`, erster Buchstabe des Vornamens = `v`)

```
u = familienv(N,mue,ustar,v,gon)
```

zur Lösung des linearen Gleichungssystems ($N =$ `N`, $\mu =$ `mue`, $\boldsymbol{u}^\star =$ `ustar`, $\boldsymbol{v} =$ `v`)

$$\left[\frac{\mu}{h^2} D + I\right] \boldsymbol{u} = \boldsymbol{v}$$

durch eine SOR Iteration. Mit `gon=true` sollen die Ergebnisse $\{\boldsymbol{u}^\star, \boldsymbol{v}, \boldsymbol{u}\}$ so grafisch dargestellt werden:

```
u     = reshape(u,N1,N1);
v     = reshape(v,N1,N1);
h1 = figure(1); clf; set(h1,'Position',[10 10 900 300]);
subplot(1,3,1);
imagesc(ustar); colormap('gray'); axis image; axis off;
title('exakt');
subplot(1,3,2);
imagesc(v); colormap('gray'); axis image; axis off;
title('verrauscht');
subplot(1,3,3);
imagesc(u); colormap('gray'); axis image; axis off;
title('entrauscht');
```

Mit `gon=false` sollen Ergebnisse nicht grafisch dargestellt werden.

Achtung: Getestet werden die Ergebnisse mit den folgenden Daten:

```
N1    = N+1;
N4    = round(N/4);
ustar = [zeros(N4,1);ones(N1-2*N4,1);zeros(N4,1)];
ustar = kron(ustar,ustar');
v     = ustar + 0.1*randn(N1,N1);
v     = reshape(v,N1N1,1);
```

und mit dem folgenden Code:

```
Dxdiag = -ones(N1,N1); Dxdiag(N1,:) = 0;
Dxsup  =  ones(N1,N1); Dxsup(1,:)   = 0;
Dx = spdiags(Dxdiag(:), 0,N1N1,N1N1) ...
   + spdiags( Dxsup(:), 1,N1N1,N1N1);
Dydiag = -ones(N1,N1); Dydiag(:,N1) = 0;
Dysup  =  ones(N1,N1); Dysup(:,1)   = 0;
Dy = spdiags(Dydiag(:), 0,N1N1,N1N1) ...
   + spdiags( Dysup(:),N1,N1N1,N1N1);
D = Dx'*Dx + Dy'*Dy;

u = (mu*D/h^2 + speye(N1N1)) \ v;
```