Numerische Mathematik 2, Übungen, WS15/16, Blatt 3

Bearbeitung: Hausaufgaben bis 22.10.2015, Programmieraufgabe bis 29.10.2015

Hausaufgaben

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

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

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

$$\frac{\mu}{h^2}\left[\underbrace{\frac{u_{i+2} - 2u_{i+1} + u_i}{h^2}}_{i \leq m-2} - 2\underbrace{\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}}_{1 \leq i \leq m-1} + \underbrace{\frac{u_i - 2u_{i-1} + u_{i-2}}{h^2}}_{2 \leq i}\right] + u_i = f_i, \quad i = 0, \ldots, m$$

wobei ein Term ausgelassen werden soll, wenn die untergeklammerte Bedingung nicht erfüllt wird. Für die lexikografische Reihenfolge $\boldsymbol{u} = \{u_i\}_{i=0}^m$, $\boldsymbol{f} = \{f_i\}_{i=0}^m$, schreiben Sie das letzte lineare Gleichungssystem in der Form $A\boldsymbol{u} = \boldsymbol{f}$, $A \in \mathbb{R}^{(m+1)\times(m+1)}$. Zur Lösung dieses Systems soll die SOR Methode verwendet werden:

$$\boldsymbol{u}_k = (D + \omega L)^{-1}[((1-\omega)D - \omega U)\boldsymbol{u}_{k-1} + \omega \boldsymbol{f}], \quad k = 1, 2, \ldots$$

wobei $\omega > 0$ und $A = D + L + U$ mit D eine diagonale Matrix, L eine streng untere Dreiecksmatrix und U eine streng obere Dreiecksmatrix. Zeigen Sie, diese Iteration konvergiert für jedes $\omega \in (0, 2)$. (Hinweis: Es gilt $A = (\mu/h^4)\mathcal{D} + I$, wobei $\mathcal{D}$ sich bezüglich des kanonischen quadratischen Splines $s(x)$ darstellen lässt:

$$\mathcal{D} = \left\{\int_2^{m+1} s''(x-i)s''(x-j)dx\right\}_{i,j=0}^m, \quad s(x) = \begin{cases} \frac{1}{2}x^2, & x \in [0,1] \\ \frac{3}{4} - (x - \frac{3}{2})^2, & x \in [1,2] \\ \frac{1}{2}(x-3)^2, & x \in [2,3]. \end{cases}$$

Bemerkung: Mit der dreifärbigen Reihenfolge $\tilde{\boldsymbol{u}} = \{\{u_{3k}\}, \{u_{3k+1}\}, \{u_{3k+2}\}\}$ und der entsprechenden Matrix $\tilde{A}$ gibt es eine Permutationsmatrix P, wobei gilt $\tilde{A} = P^\top AP$, und daher konvergiert SOR auch mit dieser Reihenfolge.)

2. Sei A eine tridiagonale Matrix mit keinem diagonalen Element gleich Null. Zeigen Sie, A ist irreduzibel genau dann wenn kein nebendiagonales Element Null ist.

3. Sei $A \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, irreduzibel-diagonaldominant. Weiters hat A positive diagonale Elemente und nicht positive nebendiagonale Elemente. Zeigen Sie, A ist regulär und alle Elemente von A^{-1} sind nicht negativ. Hinweis: Sei $T_\mathrm{J} = I - D^{-1}A$ die Jacobi Iterationsmatrix. Argumentieren Sie dass $\rho(T_\mathrm{J}) < 1$ gilt, und daher sind $(I - T_\mathrm{J})$ und $D(I - T_\mathrm{J})$ regulär. Dann nutzen Sie die Neumann Reihe $(I - T_\mathrm{J})^{-1} = \sum_{k=0}^\infty T_\mathrm{J}^k$.

Programmieraufgabe

Alle Codes sollen an
`stephen.keeling@uni-graz.at`
mit Betreff
Num2 Programmieraufgabe
per Email bis zum 29.10.2015 geschickt werden.

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

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

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

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

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 Code der Form (`N`$= N$, `M`$= M$, `r`$= r$, `k`$= k$, `d`$= d$, `V`$= \boldsymbol{V}^M$)

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

zur Lösung der obigen Systeme $(I + \tau B)\boldsymbol{V}^n = \boldsymbol{V}^{n-1}$, $n = 1, \ldots, M$, wobei für jedes n die Symmetrische Gauß-Seidel Methode implementiert werden soll. Mit `gon=true` soll das Ergebnis für jedes $n = 0, \ldots, M$ so grafisch dargestellt werden:

```
surf(reshape(V,N1,N1)); axis([1 N1 1 N1 0 1]);
view([1 1 1]); drawnow; pause(0.1);
```

In jedem Zeitschritt soll die Iteration mit dem aktuellsten `V` (d.h. $\boldsymbol{V}^{n-1}$) gestartet werden. Die Lösung $\boldsymbol{V}^M$ zur Zeit $t^M = 1$ soll zum Schluss der Rechnungen ausgegeben werden. (Für welche Werte r, k, d ist Konvergenz der Methode garantiert?)

Achtung: Getestet werden die Ergebnisse z.B. mit dem folgenden Code:

```
    N=51; M=51; r=0.001; k=0.1; d=0.01;
    N1=N+1; N1N1=N1*N1; N4=round(N/4); h=1/N; tau=1/M;
    tol = 1.0e-6; kmax = 100000;

    V = [zeros(N4,1);ones(N1-2*N4,1);zeros(N4,1)];
    V = kron(V,V');
    V = reshape(V,N1N1,1);

    D0 = -speye(N1);
    D1 = spdiags(4*ones(N1,1), 0,N1,N1) ...
       - spdiags(  ones(N1,1),+1,N1,N1) ...
       - spdiags(  ones(N1,1),-1,N1,N1);
    D1(N1,N1) = 3;

    D  = kron(speye(N1),D1) ...
       + kron(spdiags(ones(N1,1),+1,N1,N1),D0) ...
       + kron(spdiags(ones(N1,1),-1,N1,N1),D0);

    D2 = spdiags(ones(N1,1), 0,N1,N1) ...
       - spdiags(ones(N1,1),-1,N1,N1);
    F  = kron(speye(N1),D2);

    B = (d/h^2)*D + (k/h)*F + r*speye(N1N1);

    A = speye(N1N1) + tau*B;
    [L,U] = lu(A);

%   surf(reshape(V,N1,N1)); axis([1 N1 1 N1 0 1]); view([1 1 1]); drawnow; pause(0.1);
    for n=1:M
        V = L\V;
        V = U\V;
%       surf(reshape(V,N1,N1)); axis([1 N1 1 N1 0 1]); view([1 1 1]); drawnow; pause(0.1);
    end
```