Numerische Mathematik 2, Übungen, WS15/16, Blatt 9

Bearbeitung: Hausaufgaben bis 03.12.2015, Programmieraufgabe bis 10.12.2015

Hausaufgaben

1. (a) Geben Sie ein Beispiel einer Matrix $A \in \mathbb{R}^{n\times n}$ an, wobei die übliche Voraussetzung der Konvergenz des QR-Verfahrens verletzt wird, dass die Eigenwerte verschiedene Beträge haben. Zeigen Sie explizit das Ergebnis des QR-Verfahrens für Ihr Beispiel.

   (b) Geben Sie ein Beispiel einer Matrix $A \in \mathbb{R}^{n\times n}$ an, wobei die übliche Voraussetzung der Konvergenz des QR-Verfahrens verletzt wird, dass die Matrix Y mit $A = XDY$, $X = Y^{-1}$, $D = \text{diag}\{\lambda_1, \dots, \lambda_n\}$, eine Dreickszerlegung $Y = L_Y R_Y$ besitzt, wobei $L_Y = I+$ streng untere Dreicksmatrix und $R =$ obere Dreicksmatrix. Zeigen Sie explizit das Ergebnis des QR-Verfahrens für Ihr Beispiel.

2. Gegeben sei eine Matrix $A \in \mathbb{R}^{n\times n}$. Mit $P^{(0)} = I$ konstruieren Sie eine Householder Transformation $P^{(i)}$, $1 \le i \le n-2$, die die $i$te Spalte $\boldsymbol{a}$ einer Matrix $P^{(i-1)} \cdots P^{(0)}A$ in die Form $P^{(i)}\boldsymbol{a} = \langle \tilde{a}_1, \dots, \tilde{a}_{i+1}, 0, \dots, 0\rangle^\top$ transformiert. (a) Beweisen Sie, $P = P^{(n-2)} \cdots P^{(1)}$ ist eine orthogonale Matrix und PA ist eine obere Hessenberg Matrix. (b) Falls A SPD ist, beweisen Sie, $PAP^\top$ ist eine tridiagonale SPD Matrix.

3. Gegeben sei eine SPD Matrix $A \in \mathbb{R}^{n\times n}$. Sei P eine (wie im letzten Beispiel) orthogonale Matrix, die so bestimmt ist dass $A_1 = PAP^\top$ eine tridiagonale SPD Matrix ist. Eine Folge $\{A_k\}_{k\ge1}$ wird mit einem QR Verfahren, $A_k = Q_k R_k$, $A_{k+1} = R_k Q_k$, bestimmt. (a) Beweisen Sie, es existieren Givens Transformationen $\{Q^{(i)}\}_{i=1}^{n-1}$ und eine orthogonale Matrix $Q_k^\top = Q^{(n-1)} \cdots Q^{(1)}$, wobei $A_k = Q_k R_k$ gilt mit einer oberen Dreicksmatrix R_k. (b) Beweisen Sie, $A_{k+1} = R_k Q_k$ ist eine SPD tridiagonale Matrix.

Programmieraufgabe

Alle Codes sollen an
`stephen.keeling@uni-graz.at`
mit Betreff
Num2 Programmieraufgabe
per Email bis zum 10.12.2015 geschickt werden.

Ein reines Signal $v : [0,1] \to \mathbb{R}$ soll durch ein gemessenes Signal $u : [0,1] \to \mathbb{R}$ abgeschätzt werden, wobei nicht verrauschte Daten u durch eine *Faltung* $u = \kappa * v$ gegeben sind,

$$u(x) = \int_0^1 \kappa(x-t)v(t)dt, \quad \kappa(z) = \frac{\exp[-z^2/s]}{\sqrt{\pi s}}, \quad s > 0.$$

Dass heisst, nicht verrauschte Daten sind mindestens *verschwommen*. Die gemessenen Daten $u(x_i)$, $x_i = (i-1)/(m-1)$, $i = 1, \dots, m$, $m \in \mathbb{N}$, sind aber verrauscht, und das exakte Signal $v^\star(t_j)$, $t_j = (j-1)/(n-1)$, $j = 1, \dots, n$, $n \in \mathbb{N}$, $m > n$, soll trotz Unschärfe und Rauschen abgeschätzt werden. Durch die Diskretisierung

$$u(x_i) \approx \sum_{j=1}^{n} \frac{\kappa(x_i - t_j)}{n-1} v(t_j), \quad i = 1, \dots, m$$

stellt man sich die Abschätzung durch die Minimierung von $\|K\boldsymbol{v} - \boldsymbol{u}\|_2$ bezüglich $\boldsymbol{v}$ vor, wobei

$$\begin{array}{rcl} \boldsymbol{u} & = & \langle u(x_1), \dots, u(x_m) \rangle^\top \in \mathbb{R}^m \\ \boldsymbol{v} & = & \langle v(t_1), \dots, v(t_n) \rangle^\top \in \mathbb{R}^n \\ \boldsymbol{v}^\star & = & \langle v^\star(t_1), \dots, v^\star(t_n) \rangle^\top \in \mathbb{R}^n \\ K & = & \{\kappa(x_i - t_j)/(n-1) : 1 \le i \le m, 1 \le j \le n\} \in \mathbb{R}^{m \times n}. \end{array}$$

Mit der Singulärwert-Zerlegung $K = U\Sigma V^\top$, $U \in \mathbb{R}^{m\times m}$, $\Sigma \in \mathbb{R}^{m\times n}$, $V \in \mathbb{R}^{n\times n}$, ist das Minimum gegeben durch $\boldsymbol{v}_0 = V\Sigma^\dagger U^\top \boldsymbol{u}$, wobei $\Sigma^\dagger \in \mathbb{R}^{n\times m}$ die Pseudo-Inverse von Σ ist. Aber je mehr verrauscht die gemessenen Daten $\boldsymbol{u}$ sind, desto stärker sind die Schwingungen in der Abschätzung $\boldsymbol{v}_0$. Um diese Schwingungen zu vermeiden, wird die Methode der abgebrochenen Singulärwerte verwendet

$$\boldsymbol{v}^\star \approx \boldsymbol{v}_r = V\Sigma_r^\dagger U^\top \boldsymbol{u}$$

wobei für $r \in (0,1)$,

$$\lceil \sigma_i^{-1} \rceil_r = \left\{ \begin{array}{rl} 1/\sigma_i & \sigma_i \ge r \max\{\sigma_j\}_{j=1}^n \\ 0 & \text{sonst} \end{array} \right. \qquad \Sigma_r^\dagger = \langle \mathrm{diag}\{\lceil \sigma_j^{-1} \rceil_r\}_{j=1}^n, 0 \rangle \in \mathbb{R}^{n\times m}.$$

Schreiben Sie einen Matlab Code der Form (`vr`$= \boldsymbol{v}_r$, `n`$= n$, `u`$= u$, `s`$= s$, `r`$= r$, `vstar`$= \boldsymbol{v}^\star$)

```
vr = familienv(n,u,s,r,vstar,gon)
```

zur Bestimmung von $\boldsymbol{v}_r$. Die Methode der Singulärwert-Zerlegung soll implementiert werden, wie sie auf dem 10. Übungsblatt beschrieben ist. Mit `gon=true` soll das Ergebnis so grafisch dargestellt werden:

```
plot(x,u,'r',t,v,'g',t,vstar,'b')
legend('daten','entfaltet','exakt')
```

Achtung: Getestet werden die Ergebnisse z.B. mit den folgenden Daten

```
n = 51; m = 101;
x = linspace(0,1,m)'; t = linspace(0,1,n);
s = 1.0e-2; r = 0.05;

k = @(z) (exp(-z.^2/s)/sqrt(pi*s))/(n-1);
K = k(kron(x,ones(1,n))-kron(ones(m,1),t));

vstar = sign(t-0.25)'-sign(t-0.75)';
u = K*vstar + 0.05*randn(m,1);
```

und mit dem folgenden Code:

```
m = length(u);

x = linspace(0,1,m)';
t = linspace(0,1,n);

k = @(z) (exp(-z.^2/s)/sqrt(pi*s))/(n-1);
K = k(kron(x,ones(1,n))-kron(ones(m,1),t));
```

```
[U,S,V] = svd(K);

Srdag = diag(S);
Srdag = Srdag.*(Srdag > r*max(Srdag));
Srdag = (Srdag ~= 0)./(Srdag + (Srdag == 0));
Srdag = [diag(Srdag),zeros(n,m-n)];

v = V*Srdag*U'*u;

if (gon)
    plot(x,u,'r',t,v,'g',t,vstar,'b')
    legend('daten','entfaltet','exakt')
end
```