Institutionen för Matematik, KTH

# Tentamen del 2
# Numeriska metoder SF1544 och BE3003
# 8.00-11.00 12/4 2017

**Inga hjälpmedel är tillåtna** (ej heller miniräknare).

Råd för att undvika poängavdrag: Skriv lösningar med fullständiga meningar och utförliga motiveringar.

Del 2 rättas bara om Del 1 i denna tentamen är godkänd.

Betygsgränser inklusive bonuspoäng: D 10, C 20, B 30 och A 40 poäng.

Det finns två alternativa uppgifter 1; bara en av dem får lämnas in.

**1.** Betrakta differentialekvationen

$$\begin{aligned}
z'(t) &= y(t)z(t)\,, \quad t > 0\\
y'(t) &= \frac{1-y(t)}{\epsilon}\,, \quad t > 0\\
z(0) &= 1\,,\\
y(0) &= \frac{1}{2}\,.
\end{aligned}$$

**1a.** (10p) Skriv en Matlabkod för att approximera $z(1)$ och $y(1)$ med bakåt Eulermetoden och en indelningen med steglängd $\Delta t < 0.1$ med parametervalet $\epsilon = 0.001$.

**1b.** (5p) Vad händer om approximationen i uppgift 1a görs med framåt Eulermetoden istället och steglängden $\Delta t = 0.1$ för parametervalet $\epsilon = 0.001$? Motivera ditt svar.

*1a. Vi har med indelningen $t_n = n\Delta t$, $n = 0, 1, 2, \ldots$, och approximationen $\begin{bmatrix} y_n \\ z_n \end{bmatrix} \simeq \begin{bmatrix} y(t_n) \\ z(t_n) \end{bmatrix}$ bakåt Eulermetoden*

$$\begin{aligned}
y_{n+1} &= y_n + \frac{\Delta t}{\epsilon}(1 - y_{n+1})\,,\\
z_{n+1} &= z_n + \Delta t y_{n+1} z_{n+1}\,.
\end{aligned}$$

*Vi får*

$$\begin{aligned}
y_{n+1} &= \frac{y_n + \Delta t/\epsilon}{1 + \Delta t/\epsilon}\,,\\
z_{n+1} &= \frac{z_n}{1 - \Delta t\, y_{n+1}}\,.
\end{aligned}$$

*Ett exempel på en Matlabkod är*

```
T=1;
N=100000;
dt=T/N;
eps= 0.001;
y=zeros(N,1);
z=zeros(N,1);
t=zeros(N,1);
y(1)=0.5;
z(1)=1;
t(1)=0;

for n=1:N-1
    t(n+1)=n*dt;
    y(n+1)=(y(n)+dt/eps)/(1+dt/eps);
    z(n+1)=z(n)/(1-dt*y(n+1));
end
plot(t,y)
hold on
plot(t,z)
```

*Ett annat exempel på Matlabkod baserad på fixpunktiterationer är*

```
T=1;
N=10000;
dt=T/N;
K=10;
v=[0.5;1];
eps=0.001;
F=@(y,z) [(1-y)/eps;y*z];

for n=1:N-1
    w=v;
    for k=1:K
        w=v+dt*F(w(1),w(2));
    end
    v=w;
end
display(v);
```

*1b. Med framåt Eulermetoden har vi*

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{\Delta t}{\epsilon}(1 - y_n)\,, \\
z_{n+1} &= z_n + \Delta t y_n z_n\,.
\end{aligned}
$$

*och* $y_{n+1} = y_n(1 - \Delta t/\epsilon) + \Delta t/\epsilon$. *Låt* $v_n = y_n - 1$. *Då är*

$$v_{n+1} = v_n - \Delta t v_n/\epsilon = (1 - \Delta t/\epsilon)v_n$$

*så*

$$|v_n| = |(1 - \Delta t/\epsilon)^n v_0| = 99^n |v_0| \to \infty$$

*när* $n \to \infty$. *Framåt Eulermetoden fungerar därför ej. Även den andra Matlabkoden baserad på fixpunktiterationer blir instabil om* $N$ *inte är tillräckligt stort.*

**Alternativ uppgift 1.** (15p) Formulera och bevisa en sats om konditionstal för lösning av linjära ekvationsystem.

**2.** Betrakta egenvärdesproblemet

$$\begin{aligned} -u''(x) + u(x) &= \lambda u(x)\,, \quad 0 < x < 1\,, \\ u(0) &= 0\,, \\ u'(1) &= 0\,, \end{aligned} \tag{1}$$

där $\lambda \in \mathbb{R}$ är egenvärde med tillhörande egenfunktion $u : [0,1] \to \mathbb{R}$.

**2a.** (5p) Formulera en finit differensmetod för att approximera (1).

**2b.** (5p) Beskriv en iterativ metod för att bestämma det minsta egenvärdet till problemet i uppgift 2a. (Tips: alla egenvärden är positiva).

**2c.** (10p) Skriv ett Matlabprogram för att bestämma det minsta egenvärdet till differensmetoden i uppgift 2a med hjälp av iterationerna från uppgift 2b.

*2a. Gör indelningen* $x_n = n\Delta x$, $n = 0, 1, 2, \ldots, N+1$, *med* $\Delta x = 1/(N+1)$, *och finit differensapproximationerna*

$$u_n \simeq u(x_n),$$
$$\frac{u_{n+1} - 2u_n + u_{n+1}}{(\Delta x)^2} \simeq u''(x_n).$$

*som insatt i* (1) *ger*

$$-\frac{u_{n+1} - 2u_n + u_{n+1}}{(\Delta x)^2} + u_n = \lambda u_n, \quad n = 1, 2, \ldots, N,$$
$$u_0 = 0,$$
$$u_{N+1} = u_N.$$

*Bilda egenvektorn* $\mathbf{u} = (u_1, u_2, \ldots, u_N, u_{N+1}) \in \mathbb{R}^{N+1}$ *och* $(N+1) \times (N+1)$ *matrisen* $A$ *med elementen*

$$A_{mn} = \frac{1}{(\Delta x)^2} \begin{cases} 2 + (\Delta x)^2 & n = m \leq N \\ 1 & n = m = N+1 \\ -1 & |n-m| = 1 \\ 0 & \textit{annars}\,. \end{cases}$$

*Egenvärdesproblemet kan då skrivas* $A\mathbf{u} = \lambda \mathbf{u}$, *med egenvärden* $\lambda \in (0, \infty)$ *och egenvektorer* $\mathbf{u} \in \mathbb{R}^{N+1}$.

*2b. Vi söker det minsta egenvärdet till* $A$ *med iterationer. Om* $A$ *har egenvärdena* $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N+1}$ *har dess invers* $A^{-1}$ *egenvärdena* $1/\lambda_1 \geq 1/\lambda_2 \geq \ldots \geq 1/\lambda_{N+1}$. *Vi kan därför använda potensiterationer tillämpat på matrisen* $A^{-1}$, *dvs.* $\mathbf{u}_{k+1} = A^{-1}\mathbf{u}_k$. *För att slippa bilda inversen löser vi i varje iteration istället* $A\mathbf{u}_{k+1} = \mathbf{u}_k$.

*2c. Ett exempel på Matlabkod är*

```
N=1000;
dx=1/(N+1);
A=zeros(N+1,N+1);
u=ones(N+1,1);
ub=zeros(N+1,1);
K=10;

for n=2:N
    A(n,n)=(2+0*dx*dx)/(dx*dx);
    A(n,n-1)=-1/(dx*dx);
    A(n,n+1)=-1/(dx*dx);
end
A(1,1)=(2+dx*dx)/(dx*dx);
A(1,2)=-1/(dx*dx);
A(N+1,N+1)=1/(dx*dx);
A(N+1,N)=-1/(dx*dx);
for k=1:K
    ub=u/norm(u);
    u=A\ub;
    lam=u'*ub;
end
display(1/lam)
```

**3.**(15p) Följande adaptiva kvadraturalgoritm beskrivs i kursboken

To approximate $\int_a^b f(x)\mathrm{d}x$ within tolerance TOL

$c = \dfrac{a+b}{2}$

$S_{[a,b]} = (b-a)\dfrac{f(a)+f(b)}{2}$

**if**$|S_{[a,b]} - S_{[a,c]} - S_{[c,b]}| < 3\mathrm{TOL}\Big(\dfrac{b-a}{b_{orig} - a_{orig}}\Big)$

accept $S_{[a,c]} + S_{[c,b]}$ as approximation over $[a,b]$

**else**

repeat above recursively for $[a,c]$ and $[c,b]$

**end**.

Ersätt i denna algoritm trapetsmetoden $S_{[a,b]} = (b-a)\frac{f(a)+f(b)}{2}$ med Eulerapproximationen $S_{[a,b]} = (b-a)f(a)$ och förklara hur den adaptiva kvadraturalgoritmen ska ändras för att en optimal approximation av $\int_0^1 f(x)\mathrm{d}x$ ska erhållas med Eulerapproximationen. Motivera ditt svar noga.

*3. Med Eulers metod har vi*

$$\int_a^b f(x)\mathrm{d}x = \underbrace{(b-a)f(a)}_{=S_{ab}} + f'(\xi)(b-a)^2/2$$

*där $\xi$ är mellan $a$ och $b$. Vi får*

$$\int_a^b f(x)\mathrm{d}x = \int_a^c f(x)\mathrm{d}x + \int_c^b f(x)\mathrm{d}x = S_{ac} + S_{cb} + f'(\xi_1)(b-a)^2/8 + f'(\xi_2)(b-a)^2/8,$$

*med $\xi_1$ mellan $a$ och $c = (a+b)/2$ och $\xi_2$ mellan $c$ och $b$, vilket ger felet*

$$(2) \qquad |\int_a^b f(x)\mathrm{d}x - (S_{ac} + S_{cb})| = |f'(\xi_1)(b-a)^2/8 + f'(\xi_2)(b-a)^2/8| \simeq \frac{(b-a)^2}{2}\frac{|f'(c)|}{2}.$$

*Vi har också*

$$|S_{ab} - (S_{ac} + S_{bc})| = \frac{(b-a)^2}{2}\Big| - f'(\xi) + \frac{f'(\xi_1) + f'(\xi_2)}{4}\Big| \simeq \frac{(b-a)^2}{2}\frac{|f'(c)|}{2}$$

*Eftersom detta uttryck är det samma som uppskattningen av felet i* (2) *ska den adaptiva algoritmen ändras till*

*To approximate* $\int_a^b f(x)\mathrm{d}x$ *within tolerance* TOL

$$c = \frac{a+b}{2}$$

$$S_{[a,b]} = (b-a)f(a)$$

**if** $|S_{[a,b]} - S_{[a,c]} - S_{[c,b]}| < \mathrm{TOL}\left(\frac{b-a}{b_{orig} - a_{orig}}\right)$

*accept* $S_{[a,c]} + S_{[c,b]}$ *as approximation over* $[a,b]$

**else**

*repeat above recursively for* $[a,c]$ *and* $[c,b]$

**end**

*för Eulers metod.*