Institutionen för Matematik, KTH

Tentamen del 2
Numeriska metoder SF1545
9.00-12.00 16/1 2015

Inga hjälpmedel är tillåtna (ej heller miniräknare).

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

Del 2 rättas om del 1 är godkänd.

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

Det finns två alternativa uppgifter 3.

1a. (5p) Formulera en finit differensapproximation av temperaturen $u(x)$ i positionen $x \in [0,1]$ i en stav, som uppfyller värmeledningsekvationen

$$\begin{cases} -\frac{d}{dx}\big(k(x)\frac{du}{dx}(x)\big) = f(x), \quad 0 < x < 1, \\ u(0) = 0, \\ u(1) = 0, \end{cases}$$

där $f(x) = x$ är en given källfunktion och $k(x) = 1{+}x$ är en given konduktivitetsfunktion.

1b. (5p) Skriv ett Matlab-program för att lösa problemet i uppgift 1a.

1c. (5p) Byt konduktiviteten $k(x) = 1{+}x$ mot den temperaturberoende konduktiviteten $k(x, u(x)) = 1 + \frac{u(x)}{10}$. Formulera en numerisk metod för detta ickelinjära problem (Matlab-program behöver inte skrivas).

1a. Låt $x_n = n\Delta x, \quad n = 0, 1, 2, \ldots, N+1$ *vara en indelning av* $[0,1]$ *där* $\Delta x = 1/(N+1)$ *och approximera med differenskvoterna*

$$\begin{aligned} u(x_n) &\simeq u_n, \\ \frac{d}{dx}u(x_n) &\simeq \frac{u_{n+1} - u_{n-1}}{2\Delta x}, \\ \frac{d^2}{dx^2}u(x_n) &\simeq \frac{u_{n+1} - 2u_n + u_{n+1}}{\Delta x^2}\,. \end{aligned}$$

Ekvationen kan skrivas

$$-k(x)u''(x) - k'(x)u'(x) = f(x), \quad 0 < x < 1$$

och differensapproximationerna ger

$$\begin{aligned} -k(x_n)\frac{u_{n+1} - 2u_n + u_{n+1}}{\Delta x^2} - k'(x_n)\frac{u_{n+1} - u_{n-1}}{2\Delta x} &= f(x_n), \quad n = 1, 2, 3, \ldots, N, \\ u_0 &= 0, \\ u_{N+1} &= 0, \end{aligned}$$

vilket kan skrivas

(1)

$$-\big(k(x_n)+\frac{\Delta x k'(x_n)}{2}\big)u_{n+1}+2k(x_n)u_n-\big(k(x_n)-\frac{\Delta x k'(x_n)}{2}\big)u_{n-1}=\Delta x^2 f(x_n), \quad n=1,2,3,\dots,N,$$
$$u_0=0,$$
$$u_{N+1}=0,$$

där $k(x_n)=1+x_n$ *och* $k'(x_n)=1$*. Ekvationerna i matrisform lyder* $A\mathbf{u}=\mathbf{f}$ *där*

$$\mathbf{u}=\begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_N \end{bmatrix}, \qquad \mathbf{f}=\begin{bmatrix} f(x_1)\Delta x^2 \\ \cdot \\ \cdot \\ \cdot \\ f(x_N)\Delta x^2 \end{bmatrix}$$

och

$$A_{nm}=\begin{cases} -(1+x_n+\Delta x/2), & m=n+1\le N \\ 2(1+x_n), & n=m \\ -(1+x_n-\Delta x/2), & m=n-1\ge 1 \\ 0, & |m-n|>1\,. \end{cases}$$

1b. Ett exempel på matlabprogram är

```
clear all
close all
N=100;
dx=1/(N+1);
u=zeros(N,1);
A=zeros(N,N);
F=zeros(N,1);

k=@(x) 1+x;
kp=@(x) 1;
f=@(x) x;

for n=2:N-1
    x=n*dx;

    A(n,n)=2*k(x);
    A(n,n+1)= -k(x)-kp(x)*dx/2;
    A(n,n-1)= -k(x)+kp(x)*dx/2;

    F(n)=f(x)*dx^2;
end
```

```
A(1,1)= 2*k(dx);
A(1,2)= -k(dx)-kp(dx)*dx/2;

A(N,N)=2*k(N*dx);
A(N,N-1)=-k(N*dx)+kp(N*dx)*dx/2;

u=A\F;

plot(u)
```

1c. Låt nu $k(x) = 1 + u(x)/10$. Vi har $k'(x) = u'(x)/10$ och får enligt (1) ekvationssystemet $F_n(\mathbf{u}) = 0, \quad n = 1, 2, \ldots, N, \quad u_0 = u_{N+1} = 0$ där

$$F_n(\mathbf{u}) = -(1 + \frac{u_n}{10})\frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x^2} - \frac{1}{10}\big(\frac{u_{n+1} - u_{n-1}}{2\Delta x}\big)^2 - f(x_n), \quad n = 1, 2, \ldots, N$$

och $u_0 = u_{N+1} = 0$. Jacobimatrisen har komponenterna

$$F'_{nm} = \begin{cases} -(1 + u_n/10)/\Delta x^2 - \frac{1}{10}(u_{n+1}/(2\Delta x^2) + \frac{1}{10}u_{n-1}/(2\Delta x^2), & m = n + 1 \leq N \\ -\frac{1}{10}\frac{u_{n+1}-2u_n+u_{n-1}}{\Delta x^2} + (1 + \frac{u_n}{10})\frac{2}{\Delta x^2}, & m = n \\ -(1 + u_n/10)/\Delta x^2 - \frac{1}{10}u_{n-1}/(2\Delta x^2) + \frac{1}{10}u_{n+1}/(2\Delta x^2), & m = n - 1 \geq 1 \\ 0, & |m - n| > 1 \end{cases}$$

Newtoniterationerna $\mathbf{u}[i] \in \mathbb{R}^N$ med startgissningen $\mathbf{u}[0] = 0 \in \mathbb{R}^N$ kan skrivas

$$\mathbf{u}[i + 1] = \mathbf{u}[i] - \big(\mathbf{F}'(\mathbf{u}[i])\big)^{-1}\mathbf{F}(\mathbf{u}[i]), \quad i = 0, 1, 2 \ldots$$

där $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^N$ är vektorn med komponenterna $F_n, \quad n = 1, \ldots, N$ och $\mathbf{F}'(\mathbf{u})$ är $N \times N$ matrisen med komponenterna F'_{nm}.

2a. (4p) Formulera en Monte Carlo-metod för att approximera integralen

$$\int_0^1 e^{-x^2}\, dx$$

och skriv ett Matlabprogram som utför approximationen numeriskt.

2b. (5p) Formulera Monte Carlo-metoden för att approximera

$$\int_0^1 \int_0^1 \int_0^1 \frac{dx_1 dx_2 dx_3}{e^{x_1^2} + e^{x_2^2} + e^{x_3^2}} \tag{2}$$

och skriv ett Matlabprogram som utför approximationen numeriskt.

2c. (2p) Anta att det räcker med 1000 funktionsevalueringar i Monte Carlo-metoden för att ge ett förväntat fel av storleksordningen 0.1. Hur många funktionsevalueringar behövs ungefär för att ge ett förväntat fel av storlek 10^{-3}?

2d. (4p) Formulera en numerisk metod för att approximera integralen (2) som inte bygger på Monte Carlo-metoden. Antag att 10^4 funktionsevalueringar räcker för att få beräkningsfelet av storlek 0.1. Hur många funktionsevalueringar behövs ungefär för att få felet 10^{-3}?

2e. (5p) Monte Carlo-metoden har en fördel jämfört med vanlig kvadratur vid integralberäkningar i hög dimension. Motivera matematiskt varför Monte Carlo-metoden också har en fördel när integranden inte är deriverbar.

2a. Låt $X(n)$, $\quad n = 1, 2, 3, \ldots, N$ vara oberoende likformigt fördelade stokastiska variabler på $[0, 1]$. Då approximerar medelvärdet $\sum_{n=1}^{N} \frac{e^{-(X(n))^2}}{N}$ integralen.

Ett exempel på matlabprogram är

```
N=100000;
I=0;
for n=1:N
    x=rand;
    I=I+exp(-x^2)/N;
end
display('integral=');
display(I);
```

2b. Låt $X_1(n), X_2(n), X_3(n)$, $\quad n = 1, 2, 3, \ldots, N$ vara oberoende likformigt fördelade stokastiska variabler på $[0, 1]$. Då approximerar medelvärdet

$$\sum_{n=1}^{N} \frac{N^{-1}}{e^{X_1(n)^2} + e^{X_2(n)^2} + e^{X_3(n)^2}}$$

integralen $\int_0^1 \int_0^1 \int_0^1 \frac{dx_1 dx_2 dx_3}{e^{x_1^2} + e^{x_2^2} + e^{x_3^2}}$.

Ett exempel på matlabprogram är

```
N=100000;
I=0;
for n=1:N
    x1=rand;
    x2=rand;
    x3=rand;
    I=I+1/(exp(x1^2)+exp(x2^2)+exp(x3^2))/N;
end
display('integral=');
display(I);
```

2c. Låt $I := \int_0^1\int_0^1\int_0^1 \frac{dx_1dx_2dx_3}{e^{x_1^2}+e^{x_2^2}+e^{x_3^2}}$. *Vi har kvadraten av medelkvadratfelet*

$$
\begin{aligned}
\epsilon^2 &= \mathbb{E}[(\sum_{n=1}^{N}\frac{N^{-1}}{e^{x_1^2}+e^{x_2^2}+e^{x_3^2}} - I)^2]\\
&= \frac{1}{N}\mathbb{E}[(\frac{1}{e^{x_1^2}+e^{x_2^2}+e^{x_3^2}} - I)^2]\\
&= \frac{1}{N}\int_0^1\int_0^1\int_0^1 \Big(\frac{1}{e^{x_1^2}+e^{x_2^2}+e^{x_3^2}} - I\Big)^2 dx_1dx_2dx_3\\
&=: C/N
\end{aligned}
\tag{3}
$$

så om $\epsilon = 0.1$ *kräver* $N = 10^3/3$ *har vi* $\epsilon^2 = C/N$ *och* $C = N\epsilon^2 = 10^3\times 10^{-2}/3$. *Detta ger för* $\epsilon = 10^{-3}$ *att* $N = C/\epsilon^2 = 10^3\times 10^{-2}\times 3^{-1}/10^{-6} = 10^7/3$ *och svaret blir* 10^7 *funktionsevalueringar.*

2d. Gör en indelning $(x_1, x_2, x_3) = (n\Delta x, m\Delta x, k\Delta x)$ *där*

$$
\begin{aligned}
n &= 0, 1, 2, \ldots, N,\\
m &= 0, 1, 2, \ldots, N,\\
k &= 0, 1, 2, \ldots, N.
\end{aligned}
$$

Vi har då för $f(x_1, x_2, x_3) = 1/(e^{x_1^2}+e^{x_2^2}+e^{x_3^2})$ *integralapproximationen*

$$
\int_0^1\int_0^1\int_0^1 f(x_1,x_2,x_3)dx_1dx_2dx_3 = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\sum_{k=0}^{N-1} f(n\Delta x, m\Delta x, k\Delta x)\Delta x^3 + \mathcal{O}(\Delta x)
$$

och felet $\epsilon = \mathcal{O}(\Delta x) \approx C\Delta x \approx 0.1$ *Antal funktionsevalueringar är* $N^3 = 10^4$ *så* $N = 10^{4/3} = 1/\Delta x$ *och vi får* $C = 0.1/\Delta x = 0.1\times 10^{4/3}$. *Detta ger för* $\epsilon = 10^{-3}$ *att* $\Delta x = \epsilon/C = 10^{-3}/(0.1\times 10^{4/3})$. *Då blir antal funktionsevalueringar* $N^3 = 1/\Delta x^3 = (0.1\times 10^{4/3}/10^{-3})^3 = 10^{10}$.

2e. Monte Carlo-metoden för approximation av $\int_0^1 f(x)dx$ *ger kvadraten av medelkvadratfelet (enligt* (3) *med* $I := \int_0^1 f(x)dx$*)*

$$
\mathbb{E}[\big(\sum_{n=1}^{N} f(X_n)/N - I\big)^2] = \int_0^1 (f(x)-I)^2dx/N.
$$

Felet för integration med Eulermetoden blir i varje intervall

$$
\begin{aligned}
\int_{x_n}^{x_{n+1}} f(x)dx - f(x_n)\Delta x &= \int_{x_n}^{x_{n+1}} \big(f(x)-f(x_n)\big)dx\\
&= \int_{x_n}^{x_{n+1}} f'(\xi)(x-x_n)dx\\
&\le \max_{x_n\le y\le x_{n+1}} |f'(y)| \int_{x_n}^{x_{n+1}} (x-x_n)dx\\
&= \max_{x_n\le y\le x_{n+1}} |f'(y)|\Delta x^2/2,
\end{aligned}
$$

för något ξ mellan x_n och x_{n+1} bestämt av Taylors formel. Om f är deriverar blir det globala felet begränsat av

$$\sum_{n=0}^{N-1} \max_{x_n\le y\le x_{n+1}} |f'(y)|\Delta x^2/2 \le \max_{0\le y\le 1} |f'(y)|\Delta x/2$$

men om f' inte är begränsad får vi ingen eller lägre konvergensfart. Monte Carlo-metoden däremot konvergerar om $\int_0^1 (f(x)-I)^2 dx$ är begränsad vilket det kan vara även om f' ej existerar, t.ex. $f(x) = x^{-1/3}$ för $x \in (0,1)$ har $f'(x) = -x^{-4/3}/3$ och $\max_{0\le y\le 1} |f'(y)| = \infty$ men $\int_0^1 (f(x)-I)^2 dx = \int_0^1 (x^{-1/3} - 3/2)^2 dx < \infty$.

3. Vinkelutslaget $x(t)$ vid tiden t för en pendel med längden L uppfyller

$$\frac{d^2x(t)}{dt^2} = -\frac{g}{L}\sin x(t), \quad x(0) = 0, x'(0) = 1,$$

där g är gravitationskonstanten.

3a. (5p) Formulera Eulermetoden för att numeriskt bestämma en approximation till $x(t)$.

3b. (5p) Skriv ett Matlabprogram för Eulermetoden i uppgift 3a som bestämmer en approximation i tidsintervallet $[0,b]$, $b > 0$.

3c. (5p) Modifiera programmet i uppgift 3b så att en approximation av periodtiden skrivs ut. Periodtiden T definieras av $x(t+T) = x(t)$, för alla t.

3a. Låt $x_1(t) = x(t)$ och $x_2(t) = x'(t)$ då uppfyller $X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ ekvationen

$$\frac{d}{dt}X(t) = \begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -\frac{g}{L}\sin x_1(t) \end{bmatrix}.$$

Definiera $f : \mathbb{R}^2 \to \mathbb{R}$ för $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ som

$$f(y) = \begin{bmatrix} y_2 \\ -\frac{g}{L}\sin(y_1) \end{bmatrix}.$$

Med $\Delta t > 0$ och indelningen $t_n = n\Delta t$, för $n = 0,1,2,\ldots$, och approximationen $X(t_n) \simeq X_n$ får vi framåt Eulermetoden

$$\frac{1}{\Delta t}(X_{n+1} - X_n) = f(X_n), \quad n = 0,1,2,\ldots$$

$$X_0 = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

3b. Ett exempel på program är

```
N=100;
TT=1;
dt=TT/N;
g=1;
L=1;
```

```
a=g/L;
X=zeros(N,2);
F=zeros(2,1);
t=zeros(N,1);

X(1,:)=[0;1];
t(1)=0;

for n=1:N-1
    x1=X(n,1);
    x2=X(n,2);

    X(n+1,:)=X(n,:) + dt*[x2; -a*sin(x1)];
    t(n)=(n-1)*dt;
end
t(n+1)=t(n)+dt;
plot(t,X(:,1));
```

3c. Vi kan bestämma periodtiden med den tid pendeln passerar $x = 0$ *andra gången efter starten (eller dubbla tiden tills den passerar första gången). Här är ett exempel på matlabprogram*

```
clear all
close all
N=10000;
TT=10;
dt=TT/N;
g=1;
L=1;
a=g/L;
T=0;
X=zeros(N,2);
F=zeros(2,1);
t=zeros(N,1);

X(1,:)=[0,1];
t(1)=0;

forsta_perioden=0;


for n=1:N-1
    x1=X(n,1);
    x2=X(n,2);
```

```
    X(n+1,:)=X(n,:) + dt*[x2, -a*sin(x1)];
    t(n)=(n-1)*dt;
%     if(x1<0 && forsta_perioden==0)  %alternativ 1
%         T=2*t(n);
%         forsta_perioden=1;
%     end
    if( X(n,1)<0 && X(n+1,1)>0 && forsta_perioden==0) %alternativ 2
       T=t(n);
       forsta_perioden=1;
    end
end
t(n+1)=t(n)+dt;
plot(t,X(:,1));
display('periodtid=');
display(T);
```

Alternativ uppgift 3. *(15p) Formulera och bevisa en sats som relaterar det globala och lokala felet för approximation av ordinära differentialekvationer med Eulers metod.*