KTH Matematik

Tentamen del 2
SF1511, 2017-03-16, kl 8.00-11.00,
Numeriska metoder och grundläggande programmering

Del 2, Max 50p + bonuspoäng (max 4p).
Inga hjälpmedel. Rättas endast om del 1 är godkänd.
Betygsgränser inkl bonuspoäng: 10p D, 20p C, 30p B, 40p A.

P1. Givet differentialekvationen

$$\frac{d^3y}{dx^3} - \cos(x)\frac{dy}{dx} - \alpha\frac{d^2y}{dx^2} = 0, \qquad y(0) = 10, \quad \frac{dy}{dx}(0) = 0, \quad \frac{d^2y}{dx^2}(0) = 1.$$

(4p) **a.** Skriv om problemet till ett system av differentialekvationer av första ordningen.

(6p) **b.** Skriv ett MATLAB-program som löser det omskrivna problemet för $\alpha = 0.1$ för x på intervallet 0 till 5 och ritar upp funktionen $w(x) = y(x) + 5y'(x)$.

(7p) **c.** När ett program som löser differentialekvationen kördes för $\alpha = 0.05$ fick man $y(5)$ till 17.46407. När α ändrades till 0.5 blev $y(5) = 24.77647$. Du vill hitta ett α som ger $y(5) = 20$. Man kan göra detta på flera olika sätt. Beskriv en algoritm, tex i form av ett MATLAB-program, som med fem decimalers noggrannhet hittar α. Diskutera hur effektiv din algoritm är. Alltför ineffektiva metoder ger inte full poäng.

Lösning:
P1.a.
Låt $u_1(x) = y(x)$, $u_2(x) = y'(x)$ och $u_3(x) = y''(x)$. Det ger

$$\frac{d}{dx}\begin{bmatrix} u_1(x) \\ u_2(x) \\ u_3(x) \end{bmatrix} = \begin{bmatrix} u_2(x) \\ u_3(x) \\ \cos(x)u_2(x) + \alpha u_3(x) \end{bmatrix}, \qquad \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 1 \end{bmatrix},$$

eftersom $u_3'(x) = y'''(x) = \cos(x)y'(x) + \alpha y''(x) = \cos(x)u_2(x) + \alpha u_3(x)$.

P1.b.

```
alfa = 0.1;                % alfa
h    = 0.2;                % Steglängd för oberoende variabeln x
xVek = [0:h:5]';           % x-intervall
u    = [10  0  1]';        % Startvärden för y, y' och y''
uMat = u';                 % Lagra u
for x = xVek(1:end-1)'
    u = u + h*funk(x, u, alfa);  % Framåt Euler
    uMat = [uMat ; u'];    % Lagra u
end
w = uMat(:,1) + 5*uMat(:,2);   % w = y + 5y'
plot (xVek, w)
```

P1.c.
Gör först om programmet i deluppgift (b) till en funktion som tar α som indata och som returnerar $y(5)$. Man behöver byta ut raden `alpha=0.1` till en funktionsheader och lägga till en rad på slutet som ser till att rätt värde returneras:

```
function y5 = y5funk(alpha)

... <som tidigare, men utan raden som sätter alpha till 0.1> ...

y5 = u(1);
```

Man anropar sedan funktionen `y5funk` i ett yttre program som använder tex sekantmetoden för att hitta det α som ger `y5funk`$= 20$, enligt:

```
tol = 0.5e-5;                          % Feltolerans

a0 = 0.05; f0 = y5funk(a0)-20;         % Startgissningar från uppgiftstexten
a1 = 0.5;  f1 = y5funk(a1)-20;

while abs(a1-a0)>tol
    any = a1 - f1*(a1-a0)/(f1-f0);     % Sekantmetoden
    a0  = a1;  f0 = f1;
    a1  = any; f1 = y5funk(a1)-20;
end
disp(a1)
```

P2. Ändarna av en två meter lång stav hålls vid de konstanta temperaturerna 20°C respektive 100°C. Den stationära temperaturfördelningen $u(x)$ i staven bestäms då av differentialekvationen

$$-k\frac{d^2u}{dx^2} = e^x, \qquad u(0) = 20, \quad u(2) = 100,$$

där k är stavens värmeledningsförmåga.

(10p) **a.** Formulera en andra ordningens finit differensmetod för detta randvärdesproblem. Låt $\mathbf{u} = (u_0, \ldots, u_{n+1})^T$ eller $\mathbf{u} = (u_1, \ldots, u_n)^T$ (välj det du föredrar) innehålla approximationerna $u_j \approx u(x_j)$, där $x_j = hj$ och $h = 2/(n+1)$. Metoden leder till ett linjärt ekvationssystem $A\mathbf{u} = \mathbf{b}$ av storlek $n \times n$. Härled uttryck för elementen i A-matrisen och i högerledet $\mathbf{b}$ när $n = 3$. (Ekvationssystemet behöver inte lösas.)

(10p) **b.** Implementera metoden i MATLAB för fallet $k = 5$. Programmet ska räkna ut vektorn $\mathbf{u}$. Det ska vara generellt och kunna anropas med olika värden på n.

(4p) **c.** Utöka ditt MATLAB-program så att det också beräknar en andra ordningens approximation av $u'(1)$.

Var god vänd

Lösning:
P2.a.
I varje inre punkt, $j = 0, \ldots, n+1$ eller $j = 1, \ldots, n$, approximerar vi andraderivatan i ekvationen med andra ordningens centraldifferenskvot

$$\frac{d^2u}{dx_j^2} \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}.$$

Det ger

$$-k\frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} = e^{x_j}, \qquad j = 0, \ldots, n+1 \text{ eller } j = 1, \ldots, n.$$

Multiplicera med h^2 och dividera med k,

$$-u_{j-1} + 2u_j - u_{j+1} = h^2\frac{e^{x_j}}{k}, \qquad j = 0, \ldots, n+1 \text{ eller } j = 1, \ldots, n. \tag{1}$$

För $j = 0, \ldots, n+1$, sätt in randvillkoren på första och sista raden i matris A. Det ger det linjära ekvationssystemet $A\mathbf{u} = \mathbf{b}$ med

$$A = \begin{pmatrix} 1 & & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & & 1 \end{pmatrix},$$

och högerledet

$$\mathbf{b} = \begin{pmatrix} 20 \\ h^2\frac{e^{x_1}}{k} \\ \vdots \\ h^2\frac{e^{x_n}}{k} \\ 100 \end{pmatrix}.$$

När $n = 3$ har vi $h = 1/2$ och systemet blir

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} 20 \\ \frac{e^{0.5}}{4k} \\ \frac{e}{4k} \\ \frac{e^{1.5}}{4k} \\ 100 \end{pmatrix}.$$

För $j = 1, \ldots, n$, utnyttja randvillkoren för att eliminera u_0 och u_{n+1}:

$$u_0 = 20, \qquad u_{n+1} = 100.$$

Detta ger för $j = 1$,

$$2u_1 - u_2 = h^2\frac{e^{x_1}}{k} + 20, \tag{2}$$

och för $j = n$,

$$-u_{n-1} + 2u_n = h^2\frac{e^{x_n}}{k} + 100. \tag{3}$$

Tillsammans ger (1,2,3) det linjära ekvationssystemet $A\mathbf{u} = \mathbf{b}$ med

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix},$$

och högerledet

$$\mathbf{b} = \begin{pmatrix} h^2\frac{e^{x_1}}{k} + 20 \\ h^2\frac{e^{x_2}}{k} \\ \vdots \\ h^2\frac{e^{x_{n-1}}}{k} \\ h^2\frac{e^{x_n}}{k} + 100 \end{pmatrix}.$$

När $n = 3$ har vi $h = 1/2$ och systemet blir

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \frac{e^{0.5}}{4k} + 20 \\ \frac{e}{4k} \\ \frac{e^{1.5}}{4k} + 100 \end{pmatrix}.$$

P2.b. och c.

```
clear all, close all, clc
n   = 3;              % Antal inre punkter
k   = 5;              % Stavens värmeledningsförmåga
h   = 2/(n+1);        % Steglängd i x
x   = [0:h:2]';       % Oberoende variabeln

% För j = 0, ... , n+1
sup = -ones(n+1, 1);  % Superdiagonalen
d   = 2*ones(n+2, 1); % Huvuddiagonalen
sub = -ones(n+1, 1);  % Subdiagonalen
A   = diag(sup,1) + diag(d,0) + diag (sub,-1); % A-matrisen
A(1,1)        = 1;    % Ändra första raden
A(1,2)        = 0;    % Ändra första raden
A(end, end-1) = 0;    % Ändra sista  raden
A(end, end)   = 1;    % Ändra sista  raden
b   = 0.05*exp(x);    % b-vektorn
b(1) = 20;            % Ändra första elementet
b(end) = 100;         % Ändra sista  elementet
u   = A\b;            % Beräkna u

% För j = 1, ... , n
sup = -ones(n-1, 1);        % Superdiagonalen
d   = 2*ones(n, 1);         % Huvuddiagonalen
sub = -ones(n-1, 1);        % Subdiagonalen
A   = diag(sup,1) + diag(d,0) + diag (sub,-1); % A-matrisen
b   = 0.05*exp(x(2:end-1)); % b-vektorn
b(1) = b(1) + 20;           % Justera 1:a elementet
b(end) = b(end) + 100;      % Justera sista elementet
u   = A\b;                  % Beräkna u
u   = [20 ; u ; 100];       % Lägg till ränderna
```

```
% Beräkna u'(1), dvs uppgift P2.c.
if   1 == mod(n,2)          % Om n är udda
uPrimVidx1 = (u(n/2+2.5) - u(n/2+0.5)) / (2*h)
else                        % Om n är jämnt
    uPrimVidx1 = (u(n/2+2) - u(n/2+1)) / h
end
```

P3. Givet $I = \int_0^4 f(x)dx$ med värden enligt tabellen.

x	0	1	2	3	4
f(x)	200	250	281,5	299,5	300

I_h betecknar approximationen av I, då I beräknas med trapetsregeln med steglängden h.
Om någon deluppgift behöver resultatet från en föregående deluppgift och du inte har egna resultat, får du använda en rimlig uppskattning.

(1p) **a.** Beräkna I_4 för hand.

(1p) **b.** Beräkna I_2 för hand.

(1p) **c.** R_2 betecknar en Richardsonextrapolation utifrån I_4 och I_2. Beräkna R_2 för hand.

(1p) **d.** Beräkna I_1 för hand.

(1p) **e.** R_1 betecknar en Richarsonextrapolation utifrån I_2 och I_1. Beräkna R_1 för hand.

(2p) **f.** Gör en Richardsonextrapolation för hand utifrån R_2 och R_1.

(2p) **g.** Skriv ett matlabprogram som beräknar I_h, där steglängden h är en variabel som tilldelas ett värde i början av programmet. Antag att det i samma mapp finns en färdigskriven fil funk.m med indata x och utdata f(x) enligt ovan, d.v.s fAvx = funk(x).
Ovanstående tabell ska alltså inte användas.

Lösning:
P3.a.
$4(200/2 + 300/2) = \mathbf{1000}$

P3.b.
$2(200/2 + 281.5 + 300/2) = \mathbf{1063}$

P3.c.
$1063 + (1063 - 1000)/(2^2 - 1) = \mathbf{1084}$

P3.d.
$1 \cdot (200/2 + 250 + 281.5 + 299.5 + 300/2) = \mathbf{1081}$

P3.e.
$1081 + (1081 - 1063)/(2^2 - 1) = \mathbf{1087}$

P3.f.
$1087 + (1087 - 1084)/(2^4 - 1) = \mathbf{1087.2}$

P3.g.

```
clear all, close all, clc
h = 0.2;       % Valfri steglängd
x = 0:h:4;     % x-vektor
f = funk(x);   % Färdigskriven funktion   fAvx=funk(x)
I = h * (sum(f) - f(1)/2 - f(end)/2)   % Trapetsregeln
```
