DN1212+DN1214+DN1215+DN1240+DN1241+DN1243 mfl
Tentamen i Grundkurs i numeriska metoder
Del 2 (av 2)
Lördag 2012-02-04, kl 9-12

Skrivtid 3 tim. **Inga hjälpmedel.** Rättas endast om del 1 är godkänd. Betygsgräns (inkl bonuspoäng): 10p D, 20p C, 30p D, 40p A. Maximal poäng 50 + bonuspoäng från årets laborationer (max 4p).

Var god notera att miniräknare **ej** är tillåten på denna tentamen.

Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag. Då algoritmbeskrivning begärs, avses normalt beskrivning i Matlab.

Eftersom miniräknare ej är tillåten är det tillåtet att lämna enkla beräkningsuttryck oförenklade, tex $c = 0.5 \cdot 0.2^3 \cdot \cos(\pi/3)$ i stället för det uträknade $c = 0.002$

( ) **P0.** Ange dina bonuspoäng och den kursomgång (linje och termin) där poängen erhållits. Endast poäng från senaste kursomgången är giltiga.

P1.

(5) **a)** Ekvationen $e^x = Kx(1-x^2)$ där $K = 8$ har en rot nära 0. Ge en detaljerad algoritm, gärna i form av ett Matlab-program, för att bestämma roten med (minst) 7 decimaler.

(2) **b)** Ekvationen har en negativ rot. Grovlokalisera den med ca 1 siffra.

(2) **c)** Har ekvationen några fler rötter? (Glöm inte motiveringen!)

(3) **d)** Ge en detaljerad algoritm för hur man skattar hur mycket roten i delupppgift a kan flyttas om konstanten K inte är exakt 8 utan ligger i intervallet 7.9-8.1.

P2. Man vill numeriskt skatta integralen

$$\int_{-0.8}^{0.8} cos(x^2-1)dx$$

(2) **a)** Inför lämpliga beteckningar och formulera den allmänna formeln för trapetsregeln för denna integral.

(2) **b)** Skriv upp uttrycket som erhålls med trapetsregeln och intervallet uppdelat i 4 delar.

(2) **c)** Beskriv hur man praktiskt skattar trunkeringsfelet i trapetsregelvärdet erhållet med steglängden $d = 0.1$?

(2) **d)** Med steglängden $d = 0.1$ skattades trunkeringsfelet till 0.0942. Vad förväntar man att trunkeringsfelet i trapetsregelvärdet erhållet med steglängden $d = 0.05$ ungefär blir?

Var god vänd

P3.
Givet ekvationssystemet

$$x^2 + 5\sin(y) = 0.5$$
$$3\sin(x) - y^2 = 0.6$$

(2) **a)** Bestäm en startgissning till systemet.

(5) **b)** Beskriv detaljerat en algoritm, gärna i form av ett Matlabprogram, hur man bestämmer en rot till systemet med minst 6 decimaler.

P4. En sex kilograms raket skickas upp i luften. Drivkraften uppåt beror av förbränningen av raketbränslet, vilket i sin tur gör att raketen blir lättare, så massan kommer att bero av tiden:

$$m(t) = \begin{cases} M(t) \text{ för } t \leq 5 \\ M(5) \text{ för } t > 5 \end{cases} \qquad \text{där} \quad M(t) = 6 - 0.2t + 0.2t^2$$

Drivkraften nedåt är jordens gravitationskraft, som antas vara konstant $g = 9.81$. Raketen startar från en plattform 2 meter ovan marken. Efter ca 5 minuters färd är bränslet slut och raketen faller till marken i sin fallskärm. Raketfärden beskrivs av

$$my'' = -mg - km' - C_d|y'|$$

där $k = 0.1$ är en förbränningskonstant och luftmotståndet beskrivs av $C_d = 0.2$ (tiden mäts i minuter).

(3) **a)** Formulera om differentialekvationsproblemet till ett system av första ordningen.

(3) **b)** Genomför två kompletta steg med Eulers metod och tidssteget 0.5 (använd här g=10).

(5) **c)** Skriv ett Matlab-progem som skattar raketens höjd över marken som funktion av tiden för de första 5 minutrarnas färd. Skriv ut höjden vid 5 minuter.

(3) **d)** Lägg till satser eller beskriv detaljerat hur man plottar $m(t)$ och $y(t)$ i samma diagram. (Inga andra kurvor skall plottas.)

P5. En åskådare har fotograferat raketbanan vid fem tidpunkter under flygningen.

$t =$	1.2	2.3	2.9	3.5	4.1
$y =$	12.1	24.5	36.7	39.4	32.8

(2) **a)** Skriv ett Matlabprogram (eller beskriv noggrannt en algoritm) som lägger ett lämpligt interpolationspolynom genom samtliga fem mätpunkter. Beräkna vad polynomet ger för höjd vid $t = 4$.

Kompisen filmade flygningen istället, 5 sekunder med 24 bilder per sekund. När de tittade på filmen upptäckte de att banan inte alls beskrivs av ett polynom utan i stället är på formen $y(t) = y_0 + c_1 t + c_2 t^2 + \alpha/(t+1)$

(7) **b)** Skriv ett Matlabprogram (eller beskriv noggrannt en algoritm) som med minstakvadratmetoden och samtliga filmrutor bestämmer de okända parametrarna y_0, c_1, c_2 och α. Du får anta att samtliga y-värden från filmen redan finns i vektorn *hojd*.

Lycka till och gott fortsatt ”nummande”!

DN1212+DN1214+DN1215+DN1240+DN1241+DN1243 mfl nummegrundkurser, KTH 2012-02-04

Kort förslag till lösning

**************** Svar till DEL 1 ****************

Integral Trapetsregeln och 3 delar på $\int_1^{10} x^2 dx$ blir 346.5

Interpol Med kvadratisk interpolation genom tre punkter fås $y(4) = 7/3$ och $y(3) = 2$.
Beräknas lättast med Newtons ansats!

New-Raph $x_{n+1} = x_n - f(x_n)/f'(x_n)$

Konvergens Linjär konvergens.
Ty skillnaderna är $\Delta x = (0.36, 0.21, 0.12, 0.07, 0.04)$ och då blir kvoterna
$(0.21/0.36, 0.12/0.21, 0.07/0.12, 0.04/0.07) = (0.58, 0.57, 0.58, 0.57)$ dvs alla ungefär 0.6.

(Testar man för kvadratisk konvergens får man $(0.21/0.36^2, 0.12/0.21^2, 0.07/0.12^2, 0.04/0.07^2) =$
$= (1.6, 2.7, 4.9, 8.2)$ vilket inte alls är konstant)

MKV

$$A = \begin{pmatrix} | & | \\ 1 & x^2 \\ | & | \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 2 & 4 \end{pmatrix} \quad A^T A = \begin{pmatrix} 3 & 6 \\ 6 & 18 \end{pmatrix}, A^T y = \begin{pmatrix} 13 \\ 31 \end{pmatrix} \Rightarrow c = (A^T A)^{-1} A^T b = \begin{pmatrix} 8/3 \\ 5/6 \end{pmatrix}$$

Euler $y(2.5)$ skattas till 4.0 och $y'(3.0)$ skattas till 16.5 (och $y(3.0)$ till 7.5)

Felgräns Med tex Störningsräkning skattas felgränsen till 24%. (Med Min-o-Max snarare till 25%)

3gradsMKV 4 parametrar (ty det är 4 parametrar i ett tredjegradspolynom).
MKV minimerar euklidiska normen för residualen, $r = b - Ac$.

Diffekv En fjärde ordningens differentialekvation skrivs om till fyra stycken första ordningens.
Var och en av dessa behöver ett bygynnelsevärde.

**************** Svar till DEL 2 ****************

P1a En ickelinjär ekvation. Roten skall bestämmas noggrannt: välj tex Newton-Raphsons metod: $x_{n+1} = x_n - t_n$ där $t_n = f(x_n)/f'(x_n)$. Här blir $x_0 = 0$ och tex $f(x) = Kx(1 - x^2) - e^x$ och $f'(x) = K(1 - x^2) + Kx(-2x) - e^x$ där $K = 8$. För 7 decimaler måste man iterera tills $|t_n|$ är mindre än $0.5 \cdot 10^{-7}$. (Detta är en rätt säker skattning av felet om vi har kvadratisk konvergens.)

```
x=0;
t=1;
while abs(t)>1e-9;
    f=K*x*(1-x^2)-exp(x);
    d=K*(1-x^2)+K*x*(-2*x)-exp(x);
    t=f/d
    x=x-t;
end;
rot=x
```

P1b För $x < 0$ så kan e^x approximeras med 0. Nollställen till $0 = Kx(1 - x^2)$ är ju $x = 0$ och $x = \pm 1$, så den negativa roten torde vara vid $x \approx -1$.

P1c Ja. Då $x \to -\infty$ så går $f \to -\infty$. Då $x \to +\infty$ så går $f \to +\infty$. Funktionen $f(x)$ måste alltså ha ett udda antal rötter. Vi vet om två - så det finns minst en till.

P1d Hur påverkar osäkra indata resultatet? Kollas med tex störningsräkning:
Med algoritmen i **P1a** bestäms roten med $K = 8$. Kalla detta rotvärde x_{ost}
Bestäm sedan rotens värde med genom att köra om algoritmen i **P1a** fast nu med $K = 8.1$. Kalla detta värde x_{s}
Osäkerheten i rotvärdet pga den osäkra konstanten skattas med $E_{\text{tab}} = |x_{\text{s}} - x_{\text{ost}}|$, dvs $x = x_{\text{ost}} \pm E_{\text{tab}}$
(Vi antar här att $E_{\text{tot}} = E_{\text{tab}} + E_{\text{trunk}} + E_{\text{ber}} \approx E_{\text{tab}}$ då vi ju bestämt roten med minst 7 decimaler och datorn räknar med 15-16 siffror).

P2a Om vi betecknar integranden med $f(x)$ och undre och övre gräns med a respektive b så blir trapetsregeln

$$T(h) = h \cdot \left\{ \frac{1}{2} f(a) + f(a+h) + f(a+2h) + \ldots f(b-h) + \frac{1}{2} f(b) \right\} \quad \text{med} \quad h = (b-a)/N$$

där N är det antal delar man delar upp intervallet i.

P2b Med $N = 4$ så blir $h = 0.4$ och $x = -0.8, -0.4, 0, 0.4$ och 0.8 dvs

$$T(0.4) = 0.4 \cdot \left\{ \frac{1}{2} f(-0.8) + f(-0.4) + f(0.0) + f(0.4) + \frac{1}{2} f(0.8) \right\} \quad \text{med} \quad f(x) = cos(x^2 - 1)$$

P2c Trunkeringsfelet skattas genom att jämföra med samma typ av beräkning men med dubbla steglängden: $E_{\text{trunk,T(d=0.1)}} = |T(d = 0.1) - T(d = 0.2)|$ Detta är en rätt säker skattning om $\frac{T(d=0.2) - T(d=0.4)}{T(d=0.1) - T(d=0.2)} \approx 4$.

P2d Som framgår av **P4c** skall trunkeringsfelet avta med en faktor 4 så vi förväntar oss att trunkeringsfelet med steget $d = 0.05$ blir ungefär $0.0942/4 = 0.02355$

P3a Högerleden är ganska små tal, så vi förväntar oss att x och y är små. Då gäller att $x^2 < x$ och $\sin(x) \approx x$ så systemet kan förenklas till

$$\begin{aligned} x^2 + 5\sin(y) &= 0.5 \\ 3\sin(x) - y^2 &= 0.6 \end{aligned} \quad \Longrightarrow \quad \begin{aligned} 5y &= 0.5 \\ 3x &= 0.6 \end{aligned}$$

med lösningen $x = 0.2$ och $y = 0.1$, vilket är en bra startgissning.

P3b Ett ickelinjärt ekvationssystem, löses bäst med Newtons metod för system. Sätt $\bar{z} = (x \quad y)^T$ så blir iterationerna $\bar{z}_{n+1} = \bar{z}_n - \bar{t}_n$ där $\bar{t}_n$ är lösningen till $\mathbf{J}(\bar{z}_n)\bar{t}_n = f(\bar{z}_n)$. I detta fall blir $f = \begin{pmatrix} x^2 + 5\sin(y) - 0.5 \\ 3\sin(x) - y^2 - 0.6 \end{pmatrix}$ och $\mathbf{J} = \begin{pmatrix} 2x & 5\cos(y) \\ 3\cos(x) & -2y \end{pmatrix}$. Iterera tills $||\bar{t}_n|| < 0.5 \cdot 10^{-6}$ för 6 säkra decimaler. (Detta är en rätt säker skattning av felet om vi har kvadratisk konvergens.) Startgissning enligt **P3a**.

```
x=0.2; y=0.1;
t=1;
while norm(t)>1e-8;
    f=[x*x+5*sin(y)-0.5;
    3*sin(x)-y*y-0.6 ];
    J=[2*x, 5*cos(y);
    3*cos(x), -2*y ];
    t=J\f;
    disp(norm(t))
    x=x-t(1);
    y=y-t(2);
end;
rot=[x; y]
```

P4a En andra ordningens differentialekvation blir två stycken första ordningens. Inför hjälpfunktionerna u_1 och u_2.

$$\bar{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \Longrightarrow \bar{u}' = \begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} u_2 \\ -g - k \cdot m'/m - C_d \cdot |u_2|/m \end{pmatrix} \quad \text{med} \quad \bar{u}_0 = \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\text{med} \quad m(t) = \begin{cases} 6 - 0.2t + 0.2t^2 \text{ för } & t \le 5 \\ 10 \text{ för } & t > 5 \end{cases} \quad \text{och} \quad m'(t) = \begin{cases} 0.4t - 0.2 \text{ för } & t \le 5 \\ 0 \text{ för } & t > 5 \end{cases}$$

P4b Eulers metod för två hjälpfunktioner är

$$\begin{aligned} u_{1n+1} &= u_{1n} + hu_{1n}' & u_{1n+1} &= u_{1n} + hu_{2n} \\ u_{2n+1} &= u_{2n} + hu_{2n}' \Rightarrow & u_{2n+1} &= u_{2n} + h(-g - k \cdot m'(t_n)/m(t_n) - C_d/m(t_n)|u_{2n}|) \text{ med } \bar{u}_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \\ x_{n+1} &= x_n + h & x_{n+1} &= x_n + h \end{aligned}$$

$$\bar{u}_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad \bar{u}_0' = \begin{pmatrix} u_{20} \\ -g - k \cdot m(t_0)/m(t_0) - C_d/m(t_0)|u_{20}| \end{pmatrix} = \begin{pmatrix} 0.0000 \\ -9.8667 \end{pmatrix}$$

$$\bar{u}_1 = \bar{u}_0 + h \cdot \bar{u}_0' = \begin{pmatrix} 2 \\ 0 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 0.0000 \\ -9.8667 \end{pmatrix} = \begin{pmatrix} 2.0000 \\ -4.9333 \end{pmatrix} \quad t_1 = t_0 + h = 0 + 0.5 = 0.5$$

$$\bar{u}_1' = \begin{pmatrix} u_{21} \\ -g - k \cdot m(t_1)/m(t_1) - C_d/m(t_1)|u_{21}| \end{pmatrix} = \begin{pmatrix} -4.9333 \\ -10.9867 \end{pmatrix}$$

$$\bar{u}_2 = \bar{u}_1 + h \cdot \bar{u}_1' = \begin{pmatrix} 2.0000 \\ -4.9333 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} -4.9333 \\ -10.9867 \end{pmatrix} = \begin{pmatrix} -0.4667 \\ -10.4267 \end{pmatrix} \quad t_2 = t_1 + h = 0.5 + 0.5 = 1.0$$

(ursäkta - det blev väldigt jobbiga siffror).

P4c En funktionsfil för derivataberäkningarna och ett huvudprogram:

```
function uprim=dudt(t,u);
k=0.1; Cd=0.2; g=9.81
if t<5;
   m=6-0.2*t+0.2*t*t;
   mprim=-0.2+0.4*t;
else
   m=10;
   mprim=0;
end;
```

```
u0=[2; 0]
[tut, uut]= ode45('dudt',[0 5],u0);
hojd5=uut(end,1)
```

P4d Lägg till följande satser:

```
M=6-0.2*tut+0.2*tut.*tut;
Y=uut(:,1);
plot(tut,M,'r',tut,Y,'g')
```

P5a För ett interpolerande polynom genom fem punkter får ansättas grad fyra.

```
t=[1.2; 2.3; 2.9; 3.5; 4.1];
y=[12.1; 24.5; 36.7; 39.4; 32.8];
A=[ones(size(t)) t t.*t t.*t.*t t.*t.*t.*t]; % (Upphojt-till-hatten ville inte...)
c=A\y;
s=4;
hojd=c(1)+c(2)*s+c(3)*s*s+c(4)*s*s*s+c(5)*s*s*s*s
```

P5b 5 sekunder med 24 bilder per sekund innebär 121 bildrutor. Vi söker värden på fyra parametrar. Ett överbestämt linjärt ekvationssystem. Linjära MKV-problem löses med tex normalekvationerna.

```
t=[0:(1/24):5]' ;
y=HOJD; % fanns ju redan
A=[ones(size(t)) t t.*t 1./(t+1)];
z=A\y;
y0=z(1)
c1=z(2)
c2=z(3)
alfa=c(4)
```