Namn: ....

Personnummer: ....

Program, årskurs: ....

KTH Computer Science
and Communication

Tentamen, del 1
DN1240 – Numeriska metoder gk II för F

Fredag 14 december 2012 kl 14–17

DEL 1: 20 poäng. Inga hjälpmedel. Betygsgräns för betyg E: 14 poäng (inkl. bonuspoäng).

Bonus. Ange här dina giltiga bonuspoäng från HT-12:

☐ Antal bonuspoäng

1. Ekvationen $\sin(x) = x/2$ ska lösas med Newtons metod. Utför en iteration med startapproximationen $x_0 = \pi$. **(2 p)**

Resultatet blir:

☐ 0	☐ $-2\pi/3$
☐ $\pi/3$	☐ $2\pi/3$
☐ $-\pi/3$	☐ π
☐ $\pi/2$	☐ något annat värde

2. Ett stort linjärt ekvationssystem (med **full systemmatris**, dvs med bara nollskilda element) löses med Gausselimination. Sedan löses ett dubbelt så stort linjärt ekvationssystem. Lösningstiden blir **(1.5 p)**

- ☐ ungefär densamma
- ☐ ungefär dubblerad
- ☐ ungefär 3 gånger längre
- ☐ ungefär 4 gånger längre
- ☐ ungefär 8 gånger längre
- ☐ ungefär 16 gånger längre

3. Ett stort **triangulärt** linjärt ekvationssystem löses med hjälp av ett program som utnyttjar systemets struktur. Sedan löses ett liknande system, som är dubbelt så stort. Lösningstiden blir **(1.5 p)**

- ☐ ungefär densamma
- ☐ ungefär dubblerad
- ☐ ungefär 3 gånger längre
- ☐ ungefär 4 gånger längre
- ☐ ungefär 8 gånger längre
- ☐ ungefär 16 gånger längre

4. Vi vill beräkna $y = 1 + \sin(x)$ när x är behäftat med ett (absolut) fel e_x med $|e_x| \ll 1$. Vad kan vi säga om felet e_y i y? **(1 p)**

- ☐ $e_y \approx e_x(1 + \sin(x))$
- ☐ $e_y \approx e_x \cos(x)$
- ☐ $e_y \approx \cos(e_x)$
- ☐ $e_y \approx e_x \sin(x)$
- ☐ $e_y \approx e_x(1 + \tan(x))$
- ☐ $e_y \approx e_x \tan(x)$
- ☐ $e_y \approx e_x + \sin(e_x)$
- ☐ $e_y \approx e_x/\cos(x)$

5. Villkoret för att fixpunktsiterationen $x_{n+1} = \sin(x_n) + x_n/2$ för ekvationen i 1) ska konvergera (för tillräckligt god startgissning) mot en rot $\alpha \in (0, \pi)$ lyder **(2 p)**

- ☐ $|\cos\alpha| < 1/2$
- ☐ $|\sin\alpha - \alpha/2| < 1$
- ☐ $|\sin\alpha| < 1/2$
- ☐ $|\sin\alpha + \alpha/2| > 1$
- ☐ $|\sin\alpha + \alpha/2| < 1$
- ☐ $|\cos\alpha - 1/2| > 1$
- ☐ $|\cos\alpha - 1/2| < 1$
- ☐ $|\cos\alpha + 1/2| > 1$
- ☐ $|\cos\alpha + 1/2| < 1$
- ☐ något annat villkor

6. Betrakta följande tabellerade data om funktionen $f(x)$

x	-2	-1	0	1	2
$f(x)$	2	0	1	2	8

.

a) Lägg ett interpolationspolynom (av minimalt gradtal) genom alla punkterna i tabellen. Vad blir polynomets gradtal? **(1 p)**

- ☐ 1
- ☐ 4
- ☐ 5
- ☐ 6
- ☐ 7

b) Anpassa en rät linje $g(x) = c_1 + c_2 x$ med hjälp av minsta kvadratmetoden till punkterna i tabellen.

(1.5p) c_1 blir:

- ☐ 1
- ☐ 5/13
- ☐ 10
- ☐ 13/5
- ☐ 2/5
- ☐ något annat

(1.5p) c_2 blir:

- ☐ 1.5
- ☐ 5/7
- ☐ 5
- ☐ 7/5
- ☐ 3/5
- ☐ något annat

c) Använd de tabellerade värdena för att skatta integralen $\int_{-2}^{2} f(x)\,dx$ med trapetsregeln och steglängden $h = 1$. Resultatet blir: **(2 p)**

- ☐ 1.5
- ☐ 2
- ☐ 8
- ☐ 11
- ☐ något annat

7. Differentialekvationen $y'' - xy' + \sin(\pi x) = 0$ med begynnelsedata $y(1/2) = 1$ och $y'(1/2) = 1$, löses med framåt Euler och steglängden $h = 0.2$.

Approximationen av **derivatan** $y'(0.7)$ blir: **(2 p)**

- ☐ 0.6
- ☐ 0.7
- ☐ 0.75
- ☐ 0.8
- ☐ 0.85
- ☐ 0.9
- ☐ 1.0
- ☐ 1.1
- ☐ 1.2
- ☐ 1.3
- ☐ 1.4
- ☐ något annat värde

Approximationen av **lösningen** $y(0.9)$ blir (efter två steg): **(2 p)**

- ☐ 0.98
- ☐ 1.0
- ☐ 1.02
- ☐ 1.1
- ☐ 1.2
- ☐ 1.3
- ☐ 1.32
- ☐ 1.38
- ☐ 1.4
- ☐ 1.42
- ☐ 1.5
- ☐ något annat värde

8. En numerisk integrationsmetod ger resultatet I_h när steglängden är h. För successivt halverade värden på h ges

$$I_h = 0.56401, \quad I_{h/2} = 0.51599, \quad I_{h/4} = 0.50401, \quad I_{h/8} = 0.50100$$

Vilken noggrannhetsordning har metoden? **(2 p)**

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ annat

V.g. glöm inte skriva ditt namn och personnummer överst på framsidan!

KTH Engineering Sciences

Tentamen, del 2
DN1240 – Numeriska metoder gk II för F

Fredag 14 december 2012 kl 14–17

DEL 2: Inga hjälpmedel. Rättas endast om del 1 är godkänd. Betygsgränser inkl bonuspoäng: 10p D, 20p C, 30p B, 40p A.

Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag.

1. Man vill lösa ekvationssystemet

$$x + y + \sin(y) - 1.1 = 0,$$
$$xy^2 + y + e^y - x = 0,$$

med Newtons metod.

(a) Inför beteckningar och formulera Newtons metod för detta ekvationssystem. **(3 p)**

(b) Vi antar att du känner till att det finns en rot där y är mycket liten (t.ex. via fysikaliska resonemang). Hitta en bra startgissning genom att linjärisera ekvationerna runt $y = 0$ och lösa detta förenklade problem. **(3 p)**

(c) Beskriv detaljerat en algoritm (baserad på Newtons metod) i form av ett Matlab-program som bestämmer roten med ett fel mindre än 10^{-6} i både x och y. **(6 p)**

2. Randvärdesproblemet

$$u_{xx} + u_x - \cos^2(x/2)u = 1, \qquad u(0) = 0, \quad u(2\pi) = 2,$$

ska lösas med finita differensmetoden där derivatorna approximeras med centraldifferenser. Metoden leder till ett linjärt ekvationssystem $A\boldsymbol{u} = \boldsymbol{b}$ med n obekanta.

(a) Inför lämpliga beteckningar och förklara innebörden av elementen i lösningsvektorn $\boldsymbol{u}$. Var noga med att definiera alla variabler du använder. **(2 p)**

(b) Noggrannhetsordningen för metoden är två. Förklara med hjälp av dina definierade variabler precis vad detta innebär. **(2 p)**

(c) Härled uttryck för alla element i A-matrisen och i högerledet $\boldsymbol{b}$. (Inget ekvationssystem behöver dock lösas.) **(6 p)**

(d) Antag att vi istället vill hitta en 2π-periodisk funktion som uppfyller differentialekvationen. Det betyder att Dirichlet-randvillkoren nu ersätts med periodiska randvillkor:

$$u(x) = u(x + 2\pi), \qquad \forall x.$$

Föreslå en modifikation av metoden ovan för detta fall.
Hur ändrar sig A och $\boldsymbol{b}$? **(5 p)**

3. För att lösa begynnelsevärdesproblemet

$$\frac{dy}{dt} = f(t, y), \qquad y(0) = y_0,$$

föreslår någon metoden

$$u_{n+1} = u_n + h[\alpha f(t_n, u_n) + \beta f(t_{n-1}, u_{n-1})], \qquad u_0 = y_0,$$

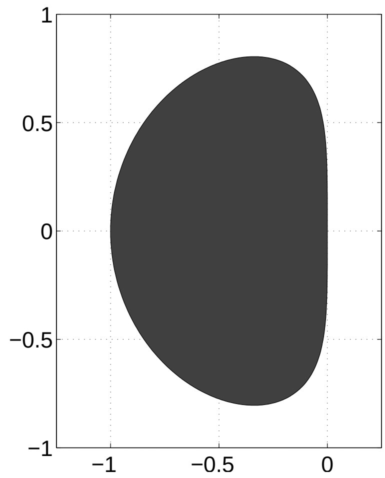
där h är en konstant steglängd och α, β är två reella koefficienter, oberoende av h.

(a) Bestäm α, β så att det lokala trunkationsfelet blir så litet som möjligt (i termer av h). Vad blir metodens noggrannhetsordning (för globala felet) med detta val av koefficienter? **(6 p)**

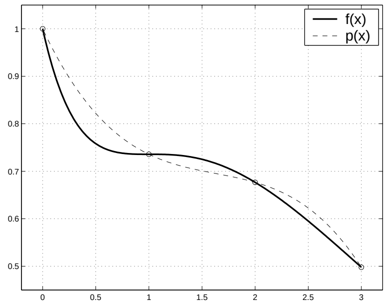
(b) Stabilitetsområdet för metoden i a) ges i figuren intill. Bestäm för vilka steglängder h som metoden är absolutstabil när

$$f(t, y) = \begin{pmatrix} -7 & 1 \\ 5 & -3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \qquad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in \mathbb{R}^2.$$

(3 p)



4. (a) Låt $p(x)$ vara det tredjegradspolynom som interpolerar $f(x) = \exp(-x)(1+x^2)$ i punkterna $x = 0, 1, 2, 3$ (se figur). Skriv en detaljerad algoritm i Matlab som först bestämmer polynomet (utan att använda `polyfit`-funktionen) och sedan med god noggrannhet räknar ut skillnaden i båglängden på kurvorna $f(x)$ och $p(x)$ när $0 \leq x \leq 3$. Hur kan man gå tillväga för att kontrollera tillförlitligheten i resultatet? **(8 p)**



Tips: Båglängden L för en kurva $y(x)$ i intervallet $x \in [a, b]$ ges av integralen

$$L = \int_a^b \sqrt{1 + y'(x)^2} dx.$$

(b) Låt $q(x)$ vara ett fjärdegradspolynom som interpolerar $f(x)$ i samma punkter som ovan. Strukturera en numerisk metod för att bestämma polynomet så att det också har precis samma båglängd som $f(x)$ när $0 \leq x \leq 3$. Beskriv vilka matematiska delproblem som behöver lösas och vilka numeriska metoder som är lämpliga. Inför beteckningar och formulera en algoritm. Programkod behövs dock ej. Diskutera hur felen i metoderna bidrar till en felgräns för det som söks. **(6 p)**

KTH Engineering Sciences

Tentamen, del 2
DN1240 – Numeriska metoder gk II för F

Fredag 14 december 2012 kl 14–17

Lösningar

DEL 2: Inga hjälpmedel. Rättas endast om del 1 är godkänd. Betygsgränser inkl bonuspoäng: 10p D, 20p C, 30p B, 40p A.

Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag.

1. Man vill lösa ekvationssystemet

$$\begin{aligned} x + y + \sin(y) - 1.1 &= 0, \\ xy^2 + y + e^y - x &= 0, \end{aligned}$$

med Newtons metod.

(a) Inför beteckningar och formulera Newtons metod för detta ekvationssystem. **(3 p)**

Lösning:
Vi vill alltså lösa $\boldsymbol{F}(\boldsymbol{x}) = 0$, med

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \boldsymbol{F}(\boldsymbol{x}) = \begin{pmatrix} x + y + \sin(y) - 1.1 \\ xy^2 + y + e^y - x \end{pmatrix}.$$

Jakobianen $J(\boldsymbol{x})$ till $\boldsymbol{F} = (f_1, f_2)^T$ är

$$J(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 1 & 1 + \cos(y) \\ y^2 - 1 & 2xy + 1 + e^y \end{pmatrix}.$$

Newtons metod för system kan då formuleras som:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - J(\boldsymbol{x}_n)^{-1}\boldsymbol{F}(\boldsymbol{x}_n).$$

(b) Vi antar att du känner till att det finns en rot där y är mycket liten (t.ex. via fysikaliska resonemang). Hitta en bra startgissning genom att linjärisera ekvationerna runt $y = 0$ och lösa detta förenklade problem. **(3 p)**

Lösning:
Vi linjäriserar ekvationerna runt $y = 0$, dvs approximerar $\sin(y) \approx y$, $xy^2 \approx 0$ och $e^y \approx 1 + y$. Det ger det förenklade problemet

$$\begin{aligned} x + 2y - 1.1 &= 0, \\ 1 + 2y - x &= 0, \end{aligned}$$

vilket har lösningen $x = 1.05$ och $y = 0.025$. Detta blir vår startgissning $\boldsymbol{x}_0 = (x_0, y_0)$.

(c) Beskriv detaljerat en algoritm (baserad på Newtons metod) i form av ett Matlab-program som bestämmer roten med ett fel mindre än 10^{-6} i både x och y. **(6 p)**

Lösning:
En detaljerad algoritm i Matlab med startgissningen $\boldsymbol{x}_0 = (1.05; 0.025)^T$ skulle kunna se ut såhär:

```
X=[1.05; 0.025];  % Startgissning
TOL = 1e-6;  % Feltolerans

r = X; % Dummy, vadsomhelst större än TOL

while (norm(r,inf)>TOL) % Max-normen

  x=X(1); y=X(2);

  f1 = x+y+sin(y)-1.1;
  f2 = x*y^2+y+exp(y)-x;

  F = [f1; f2];
  J = [1 1+cos(y); y^2-1 2*x*y+1+exp(y)];

  r = -J\F;

  X=X+r;

end
disp('(x,y) =')
disp(X');
```

Algoritmen avbryter när maxnormen $||\boldsymbol{r}_n||_\infty$ är mindre än `TOL` varför felet också kommer vara mindre än `TOL`, i detta fall 10^{-6}. Lösningen blir

$$x \approx 1.0504779, \qquad y \approx 0.0247623.$$

2. Randvärdesproblemet

$$u_{xx} + u_x - \cos^2(x/2)u = 1, \qquad u(0) = 0, \quad u(2\pi) = 2,$$

ska lösas med finita differensmetoden där derivatorna approximeras med centraldifferenser. Metoden leder till ett linjärt ekvationssystem $A\boldsymbol{u} = \boldsymbol{b}$ med n obekanta.

(a) Inför lämpliga beteckningar och förklara innebörden av elementen i lösningsvektorn $\boldsymbol{u}$. Var noga med att definiera alla variabler du använder. **(2 p)**

Lösning:
Vi vill diskretisera problemet och delar därför in intervallet $[0, 2\pi]$ i $n+1$ delar med längden $h = 2\pi/(n+1)$. Delningspunkterna kallar vi $x_j = jh$. Vi låter u_j approximera exakta lösningen i dessa punkter, dvs $u_j \approx u(x_j)$. De u_j-värden som motsvarar inre punkter är våra obekanta och utgör elementen i $\boldsymbol{u}$-vektorn, dvs $\boldsymbol{u} = (u_1, \ldots, u_n)^T$.

(b) Noggrannhetsordningen för metoden är två. Förklara med hjälp av dina definierade variabler precis vad detta innebär. **(2 p)**

Lösning:
Det betyder att felet i våra approximativa värden u_j jämfört med den exakta lösningens värden $u(x_j)$ kan begränsas av en konstant multiplicerat med steglängden i kvadrat, dvs (i maxnorm)

$$\max_{1 \le j \le n} |u_j - u(x_j)| \le Ch^2,$$

för något värde C som är oberoende av h (och n).

(c) Härled uttryck för alla element i A-matrisen och i högerledet $\boldsymbol{b}$. (Inget ekvationssystem behöver dock lösas.) **(6 p)**

Lösning:
I varje inre punkt, $j = 1, \ldots, n$, approximerar vi derivatorna i ekvationen med andra ordningens differenskvoter,

$$u_{xx}(x_j) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}, \qquad u_x(x_j) \approx \frac{u_{j+1} - u_{j-1}}{2h}.$$

Det ger

$$\frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} + \frac{u_{j+1} - u_{j-1}}{2h} - \cos^2(x_j/2)u_j = 1, \qquad j = 1, \ldots, n.$$

Multiplicera med h^2 och samla ihop termerna

$$u_{j-1}\left(1 - \frac{h}{2}\right) + u_j\left(-2 - h^2\cos^2(x_j/2)\right) + u_{j+1}\left(1 + \frac{h}{2}\right) = h^2, \tag{1}$$

där $j = 1, \ldots, n$. Vi har nu n ekvationer men $n+2$ obekanta. Utnyttja randvillkoren för att eliminera u_0 och u_{n+1}:

$$u_0 = 0, \qquad u_{n+1} = 2.$$

Detta ger för $j = 1$,

$$u_1\left(-2 - h^2\cos^2(x_1/2)\right) + u_2\left(1 + \frac{h}{2}\right) = h^2, \tag{2}$$

och för $j = n$,

$$u_{n-1}\left(1 - \frac{h}{2}\right) + u_n\left(-2 - h^2\cos^2(x_n/2)\right) + = h^2 - 2\left(1 + \frac{h}{2}\right), \qquad (3)$$

Tillsammans ger (1,2,3) det linjära ekvationssystemet $A\boldsymbol{u} = \boldsymbol{b}$ med $\boldsymbol{u} = (u_1, \ldots, u_n)^T \in \mathbb{R}^n$,

$$A = \begin{pmatrix} a_1 & b & & & \\ c & a_2 & b & & \\ & \ddots & \ddots & \ddots & \\ & & c & a_{n-1} & b \\ & & & c & a_n \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$$a_j = -2 - h^2\cos^2(x_j/2), \qquad b = 1 + \frac{h}{2}, \qquad c = 1 - \frac{h}{2},$$

och högerledet

$$\boldsymbol{b} = \begin{pmatrix} h^2 \\ h^2 \\ \vdots \\ h^2 \\ h^2 - 2\left(1 + \frac{h}{2}\right) \end{pmatrix} \in \mathbb{R}^n.$$

(d) Antag att vi istället vill hitta en 2π-periodisk funktion som uppfyller differentialekvationen. Det betyder att Dirichlet-randvillkoren nu ersätts med periodiska randvillkor:

$$u(x) = u(x + 2\pi), \qquad \forall x.$$

Föreslå en modifikation av metoden ovan för detta fall.
Hur ändrar sig A och $\boldsymbol{b}$? **(5 p)**

Lösning:
Med de nya randvillkoren är inte längre värdena på u_0 och u_{n+1} kända. Vi vet dock att de är lika pga periodiciteten, $u_0 = u_{n+1}$, och det räcker att addera en av dem till vektorn av obekanta, som nu är $\boldsymbol{u} = (u_0, \ldots, u_n)^T$. Som tidigare får vi ekvationerna

$$cu_{j-1} + a_j u_j + bu_{j+1} = h^2, \qquad j = 0, \ldots, n.$$

I första och sista ekvationen utnyttjar vi de periodiska randvillkoren,
$u(x_j) = u(x_j + 2\pi) \Rightarrow u_j = u_{j+n+1}$. Detta ger för $j = 0$, med $u_{-1} = u_n$,

$$cu_n + a_0 u_0 + bu_1 = h^2,$$

och för $j = n$, med $u_{n+1} = u_0$,

$$cu_{n-1} + a_j u_n + bu_0 = h^2.$$

Ekvationsystemet blir därför $A\boldsymbol{u} = \boldsymbol{b}$ med $\boldsymbol{u} = (u_0, \ldots, u_n)^T \in \mathbb{R}^{n+1}$,

$$A = \begin{pmatrix} a_0 & b & & & c \\ c & a_1 & b & & \\ & \ddots & \ddots & \ddots & \\ & & c & a_{n-1} & b \\ b & & & c & a_n \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

och högerledet

$$\boldsymbol{b} = \begin{pmatrix} h^2 \\ h^2 \\ \vdots \\ h^2 \\ h^2 \end{pmatrix} \in \mathbb{R}^{n+1}.$$

Matrisen och vektorerna är alltså ett steg större. Koefficienterna a_j, b, c är samma som tidigare. Matrisen får extra element i övre högra och nedre vänstra hörnet. Högerledet blir en konstant vektor.

3. För att lösa begynnelsevärdesproblemet

$$\frac{dy}{dt} = f(t, y), \qquad y(0) = u_0,$$

föreslår någon metoden

$$u_{n+1} = u_n + h[\alpha f(t_n, u_n) + \beta f(t_{n-1}, u_{n-1})], \qquad u_0 = y_0,$$

där h är en konstant steglängd och α, β är två reella koefficienter, oberoende av h.

(a) Bestäm α, β så att det lokala trunkationsfelet blir så litet som möjligt (i termer av h). Vad blir metodens noggrannhetsordning (för globala felet) med detta val av koefficienter? **(6 p)**

Lösning:

Lokala trunkationsfelet definieras som residualen när exakta lösningen sätts in i den numeriska metoden. Vid tiden t_n blir då lokala trunkationsfelet τ_n givet av relationen

$$y(t_{n+1}) = y(t_n) + h[\alpha f(t_n, y(t_n)) + \beta f(t_{n-1}, y(t_{n-1}))] + \tau_n.$$

Eftersom $y(t)$ löser differentialekvationen har vi vidare att $f(t_n, y(t_n)) = y'(t_n)$ och $f(t_{n-1}, y(t_{n-1})) = y'(t_{n-1})$. Detta ger

$$\tau_n = y(t_n + h) - \left(y(t_n) + h[\alpha y'(t_n) + \beta y'(t_n - h)]\right), \tag{4}$$

där vi också unnyttjat att $t_{n\pm1} = t_n \pm h$. Vi Taylor-utvecklar nu $y(t_n+h)$ och $y'(t_n-h)$,

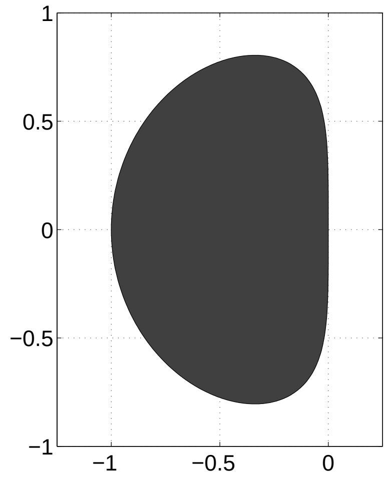
$$y(t_n+h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3), \qquad y'(t_n-h) = y'(t_n) - hy''(t_n) + O(h^2).$$

Efter insättning i (4) får vi

$$\begin{aligned} \tau_n &= y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) - \left(y(t_n) + h\alpha y'(t_n) + h\beta(y'(t_n) - hy''(t_n))]\right) + O(h^3) \\ &= hy'(t_n)[1 - \alpha - \beta] + \frac{h^2}{2}y''(t_n)[1 + 2\beta] + O(h^3). \end{aligned}$$

Vi ser att om $\alpha + \beta = 1$ blir $\tau_n = O(h^2)$. Om också $\beta = -1/2$ blir $\tau_n = O(h^3)$. Det senare ger det optimala valet av koefficienter: $\alpha = 3/2$ och $\beta = -1/2$. Det globala felet blir en ordning lägre än det lokala trunkationsfelet, så noggrannhetsordningen för metoden blir 2 med detta val av koefficienter.

(b) Stabilitetsområdet för metoden i a) ges i figuren intill. Bestäm för vilka steglängder h som metoden är absolutstabil när



$$f(t, y) = \begin{pmatrix} -7 & 1 \\ 5 & -3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \qquad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in \mathbb{R}^2.$$

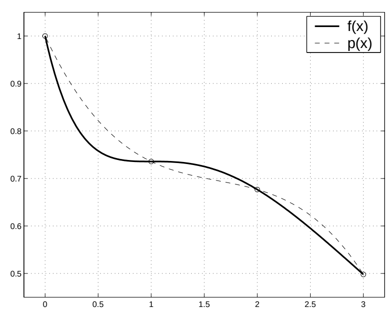
(3 p)

Lösning:
För ett system av linjära ODEer $y' = By$ är den numeriska metoden absolutstabil när $h\lambda_k$ ligger i stabilitetsområdet $\mathcal{A}$ för alla egenvärden λ_k till B. Här är $\mathcal{A}$ givet i bilden och

$$B = \begin{pmatrix} -7 & 1 \\ 5 & -3 \end{pmatrix}.$$

Egenvärdena är rötterna till det karaktäristiska polynomet, $p(\lambda) = (7+\lambda)(3+\lambda) - 5$, dvs $\lambda_1 = -2$ och $\lambda_2 = -8$. Eftersom egenvärdena är reella är $h\lambda_k \in \mathcal{A}$ ekvivalent med $-1 < h\lambda_k < 0$. (Detta avläser vi i bilden.) För λ_1 får vi $-1 < -2h < 0$, dvs $0 < h < 1/2$ och för λ_2 får vi på samma sätt att $0 < h < 1/8$. Båda villkoren måste vara uppfyllda, så metoden är absolutstabil när $0 < h < 1/8$.

4. (a) Låt $p(x)$ vara det tredjegradspolynom som interpolerar $f(x) = \exp(-x)(1+x^2)$ i punkterna $x = 0, 1, 2, 3$ (se figur). Skriv en detaljerad algoritm i Matlab som först bestämmer polynomet (utan att använda `polyfit`-funktionen) och sedan med god noggrannhet räknar ut skillnaden i båglängden på kurvorna $f(x)$ och $p(x)$ när $0 \leq x \leq 3$. Hur kan man gå tillväga för att kontrollera tillförlitligheten i resultatet? **(8 p)**



Tips: Båglängden L för en kurva $y(x)$ i intervallet $x \in [a, b]$ ges av integralen

$$L = \int_a^b \sqrt{1 + y'(x)^2} dx.$$

Lösning:
Kalla punkterna $y_0, \ldots, y_3$ (så att $y_j = j$) och skriv polynomet på den naiva ansatsen

$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3.$$

(Newtons ansats är egentligen bättre, men i det här enkla fallet duger den naiva ansatsen.) Polynomets koefficienter bestämms genom att lösa det linjära ekvationssystemet

$$\begin{pmatrix} 1 & y_0 & y_0^2 & y_0^3 \\ 1 & y_1 & y_1^2 & y_1^3 \\ 1 & y_2 & y_2^2 & y_2^3 \\ 1 & y_3 & y_3^2 & y_3^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} f(y_0) \\ f(y_1) \\ f(y_2) \\ f(y_3) \end{pmatrix}.$$

För att beräkna längden på kurvorna behöver vi derivatan av f och av p. Vi får

$$f'(x) = \exp(-x)(2x - 1 - x^2), \qquad p'(x) = c_1 + 2c_2 x + 3c_3 x^2.$$

Integralen som ger L beräknar vi slutligen med trapetsregelen. Vi delar in intervallet $[0, 3]$ i n delar med längden $h = 3/n$ och kallar delningspunkterna $x_j = jh$. För längden av f-kurvan betecknar vi integranden $I_f(x) = \sqrt{1 + f'(x)^2}$ och får

$$L_f = \int_0^3 I_f(x)dx \approx h\left(\frac{I_f(x_0)}{2} + I_f(x_1) + \cdots + I_f(x_{n-1}) + \frac{I_f(x_n)}{2}\right).$$

Approximationen av längden på p-kurvan blir likadan med $I_f(x)$ utbytt mot $I_p(x) = \sqrt{1 + p'(x)^2}$. En detaljerad algorim i Matlab ges av

```
% Del 1, beräkna koefficienterna till p(x)

y = (0:3)'; f = exp(-y).*(1+y.^2);
A = [ones(4,1) y  y.^2 y.^3];

c = A\f;

% Del 2, beräkna kurvornas längd

n = 100; h = 3/n; x = 0:h:3;

% Funktionernas derivator

fp = exp(-x).*(2*x-1-x.^2);
pp = c(2) + 2*c(3)*x + 3*c(4)*x.^2;

% Integranderna

If = sqrt(1+fp.^2);
Ip = sqrt(1+pp.^2);

% Trapetsregeln applicerad på integranderna

Lf = (sum(If)-If(1)/2-If(end)/2)*h;
Lp = (sum(Ip)-Ip(1)/2-Ip(end)/2)*h;

disp('Längdskillnad:')
disp(Lf-Lp)
```

Tillförlitligheten kan kontrolleras genom att beräkna integralerna med dubbla antalet indelningar ($2n$) och jämföra resultaten.

Lösningen blir

$$L_f \approx 3.0854, \qquad L_p \approx 3.0604, \qquad L_f - L_p \approx 0.0251.$$

(b) Låt $q(x)$ vara ett fjärdegradspolynom som interpolerar $f(x)$ i samma punkter som ovan. Strukturera en numerisk metod för att bestämma polynomet så att det också har precis samma båglängd som $f(x)$ när $0 \leq x \leq 3$. Beskriv vilka matematiska delproblem som behöver lösas och vilka numeriska metoder som är lämpliga. Inför beteckningar och formulera en algoritm. Programkod behövs dock ej. Diskutera hur felen i metoderna bidrar till en felgräns för det som söks. **(6 p)**

Lösning:

Matematiskt problem

Vi skriver $q(x)$ som

$$q(x) = p(x) + \alpha r(x),$$

där $p(x)$ är polynomet som vi beräknade i uppgift a) ovan, och $r(x)$ är valt så att det är noll i punkterna, mer specifikt

$$r(x) = x(x-1)(x-2)(x-3) = x^4 - 6x^3 + 11x^2 - 6x. \tag{5}$$

Vi vet då att för varje val av α är $q(x)$ ett fjärdegradspolynom som interpolerar punkterna. Det återstår att finna det α som gör att $q(x)$ har samma båglängd som $f(x)$. Definiera funktionen

$$L(\alpha) := \int_0^3 \sqrt{1 + (p'(x) + \alpha r'(x))^2}dx - L_f,$$

där L_f är båglängden för f och p, r är definierade ovan. Det matematiska problemet är alltså att lösa ekvationen $L(\alpha) = 0$.

Lämpliga numeriska metoder

Ekvationen $L(\alpha) = 0$ är skalär och vi kan använda en iterativ metod för skalära olinjära ekvationer. I dessa metoder behöver $L(\alpha)$ evalueras, men genom att välja *sekantmetoden* slipper vi att även behöva beräkna derivatan $L'(\alpha)$. I metoden evaluerar vi $L(\alpha)$ approximativt med *trapetsregeln* som i a). Vi noterar att utöver α behöver vi här också veta koefficienterna till p' och r' samt båglängden L_f.

Algoritm

En lämplig startgissning är $\alpha_0 = 0$ som vi sett i a) ger en avvikelse på mindre än 1%. För sekantmetoden behöver vi ytterligare en startgissning som vi väljer till det närliggande värdet $\alpha_{-1} = 0.1$. Algoritmen blir som följer:

i. Välj steglängd h för trapetsregeln och tolerans τ för avbrottskriteriet i sekantmetoden.

ii. Beräkna koefficienterna till $p'(x)$ och L_f som i a). Koefficienterna till $r'(x)$ ges från (5).

iii. Välj startgissningar $\alpha_{-1} = 0.1$, $\alpha_0 = 0$.

iv. Beräkna en approximation av $L(\alpha_{-1})$ med trapetsregeln $\Rightarrow \tilde{L}(\alpha_{-1})$

v. Låt $n = 0$ och iterera så länge som $|\alpha_n - \alpha_{n-1}| > \tau$

1. Beräkna en approximation av $L(\alpha_n)$ med trapetsregeln $\Rightarrow \tilde{L}(\alpha_n)$
2. Beräkna α_{n+1} med sekantmetoden,

$$\alpha_{n+1} = \alpha_n - \tilde{L}(x_n)\frac{x_n - x_{n-1}}{\tilde{L}(x_n) - \tilde{L}(x_{n-1})}.$$

3. $n = n + 1$

vi. Det sökta polynomet approximeras med $q(x) \approx p(x) + \alpha_n r(x)$.

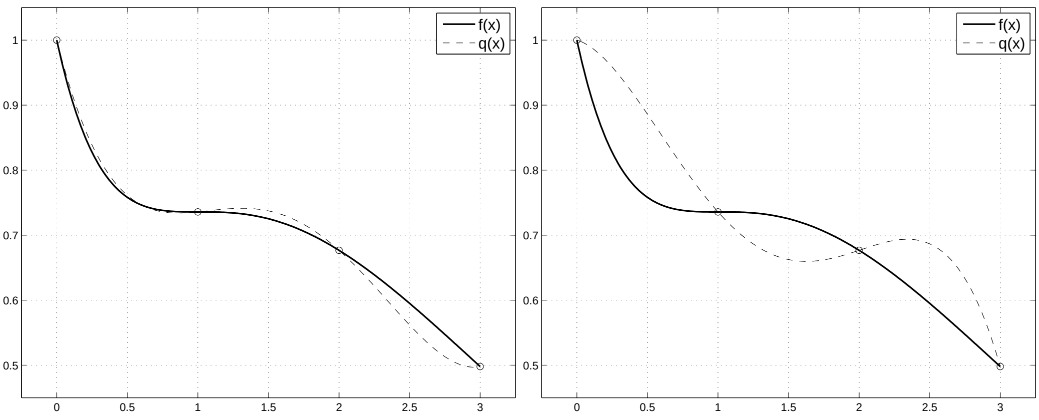
En implementation i Matlab kan se ut som nedan. (Koden fortsätter från koden i a) ovan.)

```
% Tolerans

tol = 1e-6;

% Koefficienterna för r(x)

cr = [0 -6 11 -6 1];   % r(j)=0, j=0,1,2,3

% Derivatan av r(x)

rp = cr(2) + 2*cr(3)*x + 3*cr(4)*x.^2 + 4*cr(5)*x.^3;

% Startgissningar

al1 = 0; al0 = 0.1;

% L(x0)

I0 = sqrt(1+(pp+al0*rp).^2);
L0 = (sum(I0)-I0(1)/2-I0(end)/2)*h - Lf;

while(abs(al1-al0)>tol)
   I1 = sqrt(1+(pp+al1*rp).^2);
   L1 = (sum(I1)-I1(1)/2-I1(end)/2)*h - Lf;

   tmp = al1 - L1*(al1-al0)/(L1-L0);  % Sekantmetoden
   al0 = al1;
   al1 = tmp;

   L0 = L1;
end

disp('alpha:')
disp(al1)
```

Problemet har två lösningar med

$$\alpha_1 \approx 0.06471, \qquad \alpha_2 \approx -0.06830.$$

Motsvarande polynom $q(x)$ är plottade i figurerna nedan.



Feldiskussion

Felet i algoritmen beror dels på toleransen τ, dels på steglängden h. Vi gör en enkel analys av situationen. Låt α^* vara den sökta lösningen, så att $L(\alpha^*) = 0$, och låt $\tilde{\alpha}$ vara den exakta lösningen när $L(x)$ approximeras med trapetsregeln, så att $\tilde{L}(\tilde{\alpha}) = 0$. Avbrottskriteriet i algoritmen och det faktum att trapetsregeln är en andra ordningens metod ger feluppskattningarna

$$|\tilde{\alpha} - \alpha_n| \leq \tau, \qquad |L(\tilde{\alpha}) - \tilde{L}(\tilde{\alpha})| \leq Ch^2,$$

där vi kan välja konstanten C oberoende av h. För små fel har vi också (felfortplantning) att

$$L(\tilde{\alpha}) - L(\alpha^*) \approx (\tilde{\alpha} - \alpha^*)L'(\alpha^*).$$

Tillsammans ger detta felet i α_n,

$$\begin{aligned}|\alpha_n - \alpha^*| &\leq |\alpha_n - \tilde{\alpha}| + |\tilde{\alpha} - \alpha^*| \approx |\alpha_n - \tilde{\alpha}| + \left|\frac{L(\tilde{\alpha}) - L(\alpha^*)}{L'(\alpha^*)}\right| \\ &= |\alpha_n - \tilde{\alpha}| + \left|\frac{L(\tilde{\alpha}) - \tilde{L}(\tilde{\alpha})}{L'(\alpha^*)}\right| \leq \tau + \frac{C}{L'(\alpha^*)}h^2.\end{aligned}$$

Från denna feluppskattning ser vi tex:

- Felet begränsas av både τ och h. Båda måste minska för att metoden ska konvergera.
- För att inte räkna ut något onödigt noggrant kan vi balansera felen från τ och h. Vi bör då välja τ proportionellt mot h^2, vilket gör den totala metoden andra ordningens noggrann i h.
- Faktorn $1/L'(\alpha^*)$ är (det absoluta) konditionstalet för lösningen av $L(\alpha) = 0$. När $L'(\alpha^*)$ är litet är detta problem illa konditionerat och svårlöst. (I vårt fall är $L'(0.065) \approx 0.8$ och $L'(-0.068) \approx -0.69$.)

Vi kan också välja en önskad tolerans τ och lösa problemet för successivt mindre h-värden till dess att skillnaden mellan beräknade α-värden också är mindre än τ. Det ger ett totalt fel mindre än 2τ.