Uppsala universitet
Institutionen för informationsteknologi
Teknisk databehandling

# Lösningar till tentamen i Beräkningsvetenskap II (nya versionen), 5.0 hp, 2011-06-01

## Del A

1. Eftersom temperaturen ser ut att ha ett maximum, nära klockan 13, så bör (minst) ett andragradspolynom användas, för att kunna återspegla kurvans utseende. Bäst är att välja de tre tidpunkterna 12, 14 och 16 och interpolera med ett andragradspolynom.

   Vi har tidpunkterna $t_0 = 12$, $t_1 = 14$ och $t_2 = 16$. Ansätt Newtons interpolationspolynom

   $$p(t) = A_0 + A_1(t - t_0) + A_2(t - t_0)(t - t_1)$$

   Vi har nu kravet att interpolationspolynomet ska gå igenom de tre givna punkterna, dvs. det ska anta värdena $19.6$, $20.3$ samt $19.4$ då $t$ är $12$, $14$ resp. $16$.

   $p(12) = 19.6$ ger direkt $A_0 = 19.6$.

   $p(14) = 20.3$ ger $A_0 + A_1(14 - 12) = 20.3$, dvs. $A_1 = 0.35$.

   $p(16) = 19.4$ ger $A_0 + A_1(16 - 12) + A_2(16 - 12)(14 - 12) = 19.4$, dvs. $A_2 = -0.2$.

   Vi får $p(t) = 19.6 + 0.35(t-12) - 0.2(t-12)(t-14)$ och den uppskattade temperatuen kl. 13 blir $p(13)$, dvs. $20.15$.

2. Vi har formeln för Euler framåt för $y' = f(t, y)$:

   $$y_{k+1} = y_k + h \cdot f(t_k, y_k)$$

   dvs. med $t_0 = 1$, $y_0 = 0.6$, $h = 0.2$ och $f(t, y) = \cos y \cdot \sin t$:

   $$y_1 = 0.6 + 0.2 \cdot (\cos 0.6 \cdot \sin 1) = 0.7389$$

(Räkning med räknedosan i *grader* drog ned omdömet.)

Steglängden $h = 0.2$ gav ett fel av storlek $0.004$ . Eftersom Euler framåt har noggrannhetsordning 1, så är felet proportionellt mot steglängden $h$, (dvs. $h^1$) . För att få 5 korrekta decimaler, skall felet vara högst $0.5 \cdot 10^{-5}$ . Steglängden får då högst vara

$$h = 0.2 \cdot (0.5 \cdot 10^{-5}/0.004) = 0.00025$$

(Att tolka ”5 korrekta decimaler” på annat sätt än ovan gav minusmarkering om det låg ”i närheten”, annars fel.)

3. a) Vi tillämpar Heuns metod på differentialekvationen $y' = \lambda y$:

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_n, y_n + hf(t_n, y_n))\right) \\ &= y_n + \frac{h}{2}\left(\lambda y_n + \lambda(y_n + h\lambda y_n)\right) \\ &= y_n + \frac{h}{2}(\lambda y_n + \lambda y_n + h\lambda^2 y_n) \\ &= y_n + h\lambda y_n + \frac{h^2\lambda^2}{2} y_n. \end{aligned}$$

b) Vi har

$$y_{n+1} = y_n + h\lambda y_n + \frac{h^2\lambda^2}{2} y_n = \left(1 + h\lambda + \frac{h^2\lambda^2}{2}\right) y_n$$

så metoden är stabil om $\left|1 + h\lambda + \frac{h^2\lambda^2}{2}\right| \leq 1$.

4. Det bästa värde man kan få på integralen från de 6 försöken är medelvärdet av de värden man fått i de olika försöken, dvs. $0.6583$ .

Med 1000 punkter i varje försök uppskattades felet till $0.0018$ . Med 1000000 punkter i varje försök, skulle man alltså använda 1000 gånger så många punkter. Eftersom det uppskattade felet i en Monte Carlometod för detta problem är proportionellt mot $N^{-1/2}$ , där $N$ är antal punkter, så borde felet vid 1000000 punkter i varje försök vara

$$0.0018/\sqrt{1000} \approx 5.7 \cdot 10^{-5}$$

5. a) En Monte-Carlo metod är ofta att föredra om problemet har väldigt hög dimension. Det kan till exempel vara en integral över $\Omega$, där $\Omega \subset \mathbb{R}^N$ och $N$ är stort. Anledningen är att för deterministiska metoder (som exempelvis trapetsmetoden) ökar felet exponentiellt med antalet dimensioner (givet fix steglängd), medan felet för en Monte-Carlo metod alltid (som sämst) är $O(\frac{1}{\sqrt{N}})$ (där $N$ är antalet samplingar) oberoende av dimension. För vissa högdimensionella problem blir alltså deterministiska metoder opraktiska då antalet beräkningspunkter växer alltför snabbt.

   b) En adaptiv metod för lösning av en differentialekvation anpassar steglängden beroende på egenskaper hos lösningen. Varierar lösningen snabbt använder metoden en kortare steglängd, och varierar den långsamt kan man använda en längre steglängd utan att förlora noggrannhet.

## Del B

6. Lösare *i)*: Stabilitetsvillkoret är $|\lambda h| < 2$ och aktuellt problem hade egenskaper som motsvarade $\lambda = 1000$. Man kan alltså högst använda steglängden $h = 0.002$, dvs. man måste ta minst 500 steg. Lösning med denna metod kommer alltså att ta minst $500 \cdot 10^{-6} = 5 \cdot 10^{-4}$ sekunder.

   Det fel man får vid den största möjliga steglängden är högst $(b-a)C_1 h^3 = 1 \cdot 60 \cdot 0.002^3 = 4.8 \cdot 10^{-7}$.

   Lösare *ii)*: Stabilitetsvillkoret är $|\lambda h| < 50$. Man kan alltså högst använda steglängden $h = 0.05$, dvs. man måste ta minst 20 steg. Lösning med denna metod kommer alltså att ta minst $20 \cdot 10^{-5} = 2 \cdot 10^{-4}$ sekunder.

   Det fel man får vid den största möjliga steglängden är högst $(b-a)C_2 h^2 = 1 \cdot 3 \cdot 0.05^2 = 0.0075$.

   Detta betyder att om man vill använda lösare *i)* så är det stabilitetskravet som sätter gränsen $h < 0.002$ om noggrannhetskravet är att önskat fel $\varepsilon$ är större än $4.8 \cdot 10^{-7}$. Om man vill ha ett fel $\varepsilon < 4.8 \cdot 10^{-7}$, så måste gälla att $1 \cdot 60 \cdot h^3 \leq \varepsilon$, dvs. $h \leq \sqrt[3]{\varepsilon/60}$.

   Om man vill använda lösare *ii)* så sätter stabilitetskravet gränsen $h < 0.05$ för $\varepsilon$ ner till 0.0075, och för mindre $\varepsilon$ ska gälla att $h \leq \sqrt{\varepsilon/3}$.

   För betyg 4 krävdes att det framgick att det *inte är metoderna som bestämmer* vilka felgränser en användare kan begära, utan att det är olika egenskaper hos metoderna som vid olika feltoleranser bestämmer den maximalt tillåtna steglängden.

   Effektivitetsaspekterna: Metod *i)* tar alltid minst $5 \cdot 10^{-4}$ sekunder att utföra om stabilitetsgränsen ska hållas. Felet blir då högst $4.8 \cdot 10^{-7}$. Metod *ii)* kan dock lösa problemet snabbare, på $2 \cdot 10^{-4}$ sekunder, om felet får vara 0.0075 eller större. Vid lägre felgräns måste även den metoden ta mindre steg och bli långsammare. När tar metod *ii)* också $5 \cdot 10^{-4}$ sekunder på sig? Jo, om 50 steg tas, dvs. om steget är $h = 0.02$. Den ger då ett fel som är högst $1 \cdot 3 \cdot 0.02^2 = 0.0012$. Vid lägre felgränser än så måste steget vara mindre och metoden blir

långsammare. Metod *ii)* är alltså snabbare om felet får vara 0.0012 eller större. Om det ska vara mindre än så bör metod *i)* användas, trots sina sämre stabilitetsegenskaper. Dock måste steget alltid vara högst 0.002 även om detta i vissa fall kan tyckas ge ett ”onödigt” litet fel.

7. a) I Matlabkod kan det exempelvis se ut så här (vi antar att $P_0$, $S$ och funktionen $P$ är givna):

```
N=1e4;
resultat = zeros(N,1);
for i=1:N
    %Vi ska börja varje simulering i punkten P_0.
    %k räknar antalet steg.
    p = P_0;
    k=0;
    %Så länge avståndet till startpunkten är mindre
    %än S i alla riktningar fortsätter vi.
    while(max(abs(p-P_0))<S)
        r = rand;
        %Sannolikheterna ges av funktionen P.
        %Har man använt en likformig fördelning
        %får man avdrag.
        if(r<P(1,p))
            p = p+[1 0 0];
        elseif(r<P(1,p)+P(2,p))
            p = p+[0 1 0];
        else
            p = p+[0 0 1];
        end
        k = k+1;
    end
    resultat(i) = k;
end
%Genom att ta medelvärdet av resultaten från våra
%simuleringar får vi en uppskattning av det sökta
%medelvärdet.
antal_hopp = mean(resultat);
```

b) Antag att vi först uppskattar medelvärdet genom att göra $N_1$ simuleringar och sedan gör en andra uppskattning baserat på $N_2$ simuleringar. Då kommer $e_1 \sim O(\frac{1}{\sqrt{N_1}})$ och $e_2 \sim O(\frac{1}{\sqrt{N_2}})$. Vi har alltså

$$\frac{e_2}{e_1} \approx \frac{1/\sqrt{N_1}}{1/\sqrt{N_2}} = \frac{\sqrt{N_1}}{\sqrt{N_2}}.$$

Eftersom vi vet att $N_1 = 10^3$ får vi

$$N_2 \approx 10^3 \left(\frac{e_1}{e_2}\right)^2.$$

8. (a)

$$y \sim a + b \cdot e^{Cx}$$

$C$ är känd. Detta betyder att den andra basfunktionen $e^{Cx}$ har kända värden i de punkter som ska användas. Detta betyder att det inte är svårare att lösa detta problem än att approximera med t.ex. $a + bx$. Det går att sätta upp ett överbestämt ekvationssystem med känd matris och känt högerled och lösa detta i minstakvadratmening.

(b)

$$y \sim A + b \cdot e^{cx}$$

A är känd. Detta problem är ickelinjärt, och alltså svårare att lösa än det i *(a)*. Om man flyttar över den kända konstanten och approximerar $y - A \sim b \cdot e^{cx}$, så får man dock en speciell form som kan lineariseras genom logaritmering:

$$\ln(y - A) \sim \ln b + cx$$

och detta är en approximation av en känd funktion med ett förstagradspolynom i $x$. Man får dock inte exakt samma approximation som för det ursprungliga problemet.

(c)

$$y \sim a + b \cdot e^{cx}$$

Detta är ett ickelinjärt problem, som inte kan logaritmeras, (man kan inte logaritmera "+", vilket några försökte). Om man dock känner till ett värde på $c$, som man har anledning att tro ligger "nära" det önskade $c$-värdet, så kan man lösa problemet för ett antal fixa $c$-värden $c_1 \ldots c_m$ i ett intervall nära det troliga värdet, (man löser då problem av den typ som finns i *(a)* ), och se vilket av dessa $c$-värden som ger den minsta kvadratavvikelsen. Eventuellt kan man interpolera mellan dessa för att ta reda på vilket $c$ som ger minimum. Motsvarande teknik kan användas om man känner till ett rimligt värde på $a$, man löser då på samma sätt att antal problem av typ *(b)*.