Uppsala universitet
Institutionen för informationsteknologi
Beräkningsvetenskap

Lösningsförslag till Tentamen 2021-10-26

Del A (betyg 3)

Delmål 1: *kunna skriva ett Matlab-program som gör en numerisk simulering av något fenomen, givet en matematisk modell av fenomenet.*

1. Skriv ett Matlabscript som med hjälp av Trapetsmetoden löser differentialekvationen

   $$y'(t) + 1000 \cdot y(t)^3 - t = 0, \quad t \geq 0, \quad y(0) = 1$$

   för $0 \leq t \leq 1$ och steglängden $h = 0.1$. I algoritmen kan du använda funktionen `fzero` som hittar nollstället till en funktion.

   Lösning:

   ```
   h=0.1;T=1;y0=1;
   f=@(t,y)(-1000*y^3+t);
   y(1)=y0; t=0:h:T;
   for i=1:length(t)-1
       g=@(ynew)(ynew-y(i)-h/2*(f(t(i),y(i))+f(t(i+1),ynew)));
       y(i+1)=fzero(g,y(i));
   end
   plot(t,y);
   ```

2. När man simulerar trafikflöden, t ex i en stad kan man använda stokastiska modeller och Monte Carlo-simuleringar. Med hjälp av en stokastisk process kan man då studera hur många fordon som finns vid olika trafikkorsningar en viss tidpunkt, eller hur antalet förändras med tiden.

   Antag att det finns en Matlabfunktion

   ```
   function f = traffic(f0, T)
   ```

   som simulerar trafikfödet från tidpunkt 0 till tidpunkt `T`, givet ett visst antal fordon `f0` vid tidpunkt 0. Ett anrop till funktionen ger alltså *en* simulering. Utparametern `f` är en vektor som innehåller antalet fordon vid alla korsningar. Om det exempelvis finns 10000 korsningar, så är `f` en vektor av längd 10000.
   Nu är man intresserad av att simulera antalet fordon vid korsning nummer 5 (dvs

position 5 i vektorn). Skriv ett Matlabscript som med Monte Carlo beräknar det förväntade antalet fordon i korsning nummer 5 vid tidpunkten T.

Lösning:

```
N = 1000;   % Number of simulations
f0 = 100;   % Number of vehicles to simulate
T = 10;     % Simulation end-time
num_vehicles = zeros(1,N); % Num vehicles at crossing 5
for i = 1:N
    f = traffic(f0, T);
    num_vehicles(i) = f(5);
end
expected_num_vehicles = mean(num_vehicles);
disp(expected_num_vehicles);
```

Delmål 2: *känna till viktiga begrepp i anslutning till numerisk simulering*

3. När man ska lösa differentialekvationer numeriskt måste man ta hänsyn till om ekvationen är **styv**. Vad kännetecknar en styv ODE. Ett eller flera alternativ kan vara rätt, ange alla rätta alternativ. För godkänd lösning krävs att alla rätta alternativ anges men inga felaktiga, med högst ett misstag.

   (a) Lösningen kan variera snabbt
   (b) Lösningen varierar långsamt i alla områden
   (c) Kräver litet tidssteg för stabilitet i explicita metoder
   (d) Explicita metoder är fördelaktiga
   (e) Implicita metoder är fördelaktiga
   (f) Lambda är stort för motsvarande testekvation
   (g) Noggrannheten är viktigare än stabiliteten och bestämmer steglängden
   (h) Stabiliteten är viktigare än noggrannheten och bestämmer steglängden
   (i) ODEn saknar analytisk lösning
   (j) ODEn är inte rättställd
   (k) Kan ha kraftigt skilda skalor för olika lösningskomponenter vid system

   Lösning: a,c,e,f,h,k

4. Förklara och ange skillnaden på begreppen *lokalt trunkeringsfel* och *globalt trunkeringsfel*.

   Lösning:
   Lokalt trunkeringsfel: Felet som orsakas av att ta **ett** steg med en diskretiseringsmetod.
   Globalt trunkeringsfel: Det ackumulerande felet som orsakas av att ta **flera** steg med en diskretiseringsmetod.

Delmål 3: *kunna formulera och använda de olika algoritmer och numeriska metoder som ingår i kursen*

5. När man spelar Yatzy med tärningar vill man gärna samla på så många sexor som möjligt. Sannolikheten p(k) att få k sexor i n kast är binomialfördelad. Med fem kast får vi då följande fördelning:
   p(0)=0.4019
   p(1)=0.4019
   p(2)=0.1608
   p(3)=0.0322
   p(4)=0.0032
   p(5)=0.0001

   Antag nu att vi vill simulera fall ur denna fördelning och använder då likformigt fördelade slumptal i intervallet 0 till 1. Det slumptal som vi drar är

   0.4134, 0.9997, 0.2401, 0.8002, 0.1423, 0.9123

   Använd ITS-algoritmen för att bestämma vilka utfall dessa slumptal motsvarar.

   Lösning: Beräkna den cumulativa summan (tyvärr ackumuleras avrundningsfelen i summan till ett värde över 1 med fyra decimaler men påverkar inte svaret). Mer exakt får vi:

   0.401877572016461
   0.803755144032922
   0.964506172839506
   0.996656378600823
   0.999871399176955
   1.000000000000000

   Hitta nu intervallen där slumptalen ligger, detta ger utfallen k: 1, 4, 0, 1, 0, 2
   Tex första slumptalet $0.4019 < 0.4134 < 0.8038$ ger utfall 1 sexa.

6. Antag att vi vill lösa följande problem:

$$y'' + 2y' + \frac{x}{(x+1)^2}y = \sin(x^2), \qquad x \geq 0$$
$$y(0) = 1, \quad y'(0) = 0$$

Ställ upp Euler framåt (explicita Euler) för problemet och utför ett steg med steglängden h=0.1.

Lösning: Vi börjar med att utföra ett variabelbyte:

$$\begin{bmatrix} y \\ y' \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}.$$

Den givna differentialekvationen kan då skrivas om som ett system av första ordningens differentialekvationer:

$$\underbrace{\begin{bmatrix} u' \\ v' \end{bmatrix}}_{\mathbf{u}'} = \underbrace{\begin{bmatrix} v \\ \sin(x^2) - 2v - \frac{x}{(x+1)^2}u \end{bmatrix}}_{f(x,\mathbf{u})}, \quad u(0) = 1, \quad v(0) = 0, \quad x \geq 0 \tag{1}$$

Ett steg med Euler framåt blir således:

$$\mathbf{u}_1 = \mathbf{u}_0 + hf(x_0, \mathbf{u}_0)$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} v_0 \\ \sin(x_0^2) - 2v_0 - \frac{x_0}{(x_0+1)^2}u_0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} 0 \\ \sin(0^2) - 2 \cdot 0 - \frac{0}{(0+1)^2}1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Svar: Ett steg med Euler framåt ger $y_1 = 1$ och $y_1' = 0$.

Delmål 4: *känna till egenskaper hos numeriska metoder och matematiska modeller samt kunna genomföra analys för att undersöka dessa egenskaper*

7. Härled stabilitetsvillkoret för Heuns metod tillämpad på testekvationen, $y' = \lambda y$ där $Re(\lambda) \leq 0$. Tillämpa sedan stabilitetsanalysen på problemet $y' + \frac{1000}{2+\sin(x)}y - x = 0$. Vilket är det största tidssteget som vi kan välja för att Heun ska vara stabil för det

givna problemet.

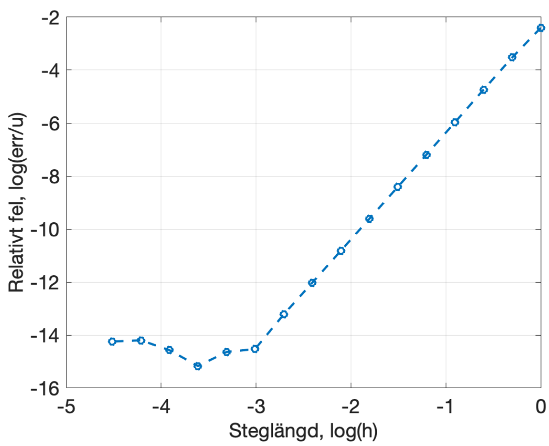
Lösning: Tillämpa Heun på testekvationen $y' = \lambda y$.

$$y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_i + hf(t_i, y_i)) = y_i + \frac{h}{2}(\lambda y_i + \lambda(y_i + h\lambda y_i))$$

$$= (1 + h\lambda + (h\lambda)^2/2)y_i = q(h\lambda) \cdot y_i$$

Heun är stabil för $|q(h\lambda)| \leq 1$. För problemet, $y' = -\frac{1000}{2+\sin(x)}y + x$, är $\lambda = -\frac{1000}{2+\sin(x)}$ och x en lägre ordningens term som inte påverkar stabiliteten. Heun är stabil för problemet om den är stabil för alla $\lambda = \lambda(x)$ i lösningsområdet, dvs betrakta *frysta koefficienter* och undersök testekvationen för alla x. Lös ekvationen $q(h\lambda) = 1$ vilket ger $\lambda h = 0$ och $\lambda h = -2$. Notera att $q(\lambda h)$ är en 2:a gradsfunktion med minimum $q(\lambda h) = 0.5$ i $\lambda h = -1$. För stabilitet måste λh ligga mellan dessa värden, dvs $0 \leq h \leq \frac{-2}{\lambda}$. Kritiska fallet blir för $\lambda_{\max} = -1000$, alltså måste $h \leq 0.002$.

8. Du har kommit över en ODE-lösare men vet inte vilken metod som ligger bakom och inte heller vilken noggrannhet den har. Du ställer då upp ett experiment där du löser en ODE med en känd analytisk lösning. På så sätt kan du jämföra den numeriska lösningen med den analytiska lösningen och beräkna felet för en given steglängd. Dina experiment genererar följande graf, se Figur 1. Förklara de olika delarna/regionerna



Figur 1: *Experiment med ODE-lösare.*

i grafen där felet beter sig olika, dvs förklara varför felet beter som det gör i dessa områden och vad det beror på. Bestäm metodens noggrannhetsordning.

Lösning: Om felet i en metod är proportionerligt mot h^p säger man att metoden har noggrannhetsordning p. För $h > 10^{-3}$ ser vi ett tydligt linjärt samband mellan $\log(\text{error})$ och $\log(h)$. Lutningskoefficienten kan läsas av som $p = 4$. Vi har då följande proportionalitet:

$$\log(\text{error}) \propto 4\log(\text{h}), \quad h \geq 10^{-3},$$
$$\Rightarrow \text{error} \propto h^4$$

Noggrannhetsordning är alltså $p = 4$. Det linjära sambandet upphör för $h \leq 10^{-3}$. Vid dessa steglängder är felet så pass litet att det domineras av inexaktheter i flyttalsaritmetik. D.v.s. avrundningsfel.

Delmål 5: *kunna använda kunskap om egenskaper för att värdera och argumentera för olika metoders och modellers lämplighet i anslutning till en given problemställning*

9. När man ska välja en ODE-lösare till ett givet problem så kan man antingen välja *i) en explicit metod, ii) en implicit metod* eller *iii) en adaptiv metod.* För var och en av metoderna ange när de är tillämpliga och fördelaktiga samt ge exempel på en tillämpning eller en specifik ODE där man ska välja respektive metod och som är det bästa valet för det problemet.

   Lösning:
   Explicit metod: Oftast enkla att implementera och mindre beräkningstunga. Sämre stabilitetsegenskaper. Fördelaktig att använda när stabilitet inte är ett problem, dvs för icke-styva problem. Exempelvis: $y' = -10^{-3} \cdot \cos(y) + t$.

   Implicit metod: Implicita metoder är mer beräkningstunga än explicita metoder men har bättre stabilitetsegenskaper. Implicita metoder bör användas för styva ekvationer. Exempelvis: $y' = -10^3 \cdot \cos(y) + t$

   Adaptiv metod: Adaptiva metoder justerar tidssteget lokalt så att en given noggrannhet uppnås överallt. Adaptiva metoder är därav fördelaktiga att använda i problem där lösningen kan variera olika mycket i olika områden. Likaså är adaptiva metoder fördelaktiga att använda på problem som innehåller båda styva och icke-styva partier. En adaptiv metod kan ta längre tidssteg i de icke-styva partierna, och således spara in på antalet beräkningar. Samtidigt kan den pressa ner tidssteget i styva partierna för att undvika instabilitet. Exempel på ODEr där en adaptiv metod är fördelaktig är t.ex. Satelliten i räkneövningen och van der Pols ekvation: $y'' - \mu(1 - y^2)y' + y = 0, \quad \mu = 1000.$

10. ODEr kan också lösas stokastiskt med Gillespies algoritm, i vilka fall kan detta vara lämpligt/olämpligt och varför. Diskutera fallen *i) epidemimodellering med litet antal individer, ii) epidemimodellering med stort antal individer, iii) dubbelpendel med små vinklar, iv) dubbelpendel med stora vinklar, v) en modell för aktiekurser.*

Lösning:
i) Lämpligt, slumpen spelar stor roll för hur smittspridningen sker (jmf projekt 2).
ii) Olämpligt, de slumpmässiga variationerna har inte längre så stor betydelse men det blir också beräkningstungt med många individer.
iii) Olämpligt, dubbelpendeln beskrivs med en deterministisk modell och bör lösas med en deterministisk metod då det inte finns några slumpmässiga variationer i tillämpningen.
iv) Olämpligt, för stora vinklar beter sig dubbelpendeln kaotiskt och modellen blir känslig för små störningar. Små slumpmässiga variationer kan då ge helt skilda lösningar, simuleringen blir rent nonsens.
v) Lämpligt, aktiemodeller innehåller en naturlig slumpvariation (volatilitet).

Del B (betyg 4/5)

11. I den här uppgiften ska du beräkna en integral med Monte Carlometoden och uppskatta dess beräkningskomplexitet. Vi delar upp uppgiften i mindre delar och delfrågor. Delfrågorna kan komplettera eventuellt missade mål från del A men för att nå upp till betyg 4 bör uppgiftens samtliga delar besvaras och mål uppfyllas.

    (a) Beräkna integralen $I = \int_0^1 \frac{\sin(x)}{\ln(x+2)} dx$ med Monte Carlometoden och 10 likformigt fördelade slumptal på intervallet $x \in [0, 1]$.

    Lösning: Integralen kan uppskattas som

    $$I \approx \frac{b-a}{N} \sum_i^N f(x_i),$$

    där $[a, b]$ är integralgränserna, N är antalet slumptal, f är funktionen vi vill integrera och x_i är slumptal dragna från en likformig fördelning på intervallet $[a, b]$. Givet att $f(x) = \frac{\sin(x)}{\ln{(x+2)}}$, $N = 10$, $a = 0$ och $b = 1$. Vi genererar 10st likformigt fördelade slumptal:

    $$x = \{0.81,\ 0.90,\ 0.12,\ 0.91,\ 0.63,\ 0.09,\ 0.27,\ 0.54,\ 0.95,\ 0.96\}$$

    och beräknar

    $$I \approx \frac{1-0}{10} \sum_i^{10} \frac{\sin(x_i)}{\ln{(x_i+2)}} = 0.55$$

    (b) Om vi skulle göra om beräkningen med 10 nya slumptal skulle vi antagligen få ett nytt resultat. Varje upprepning av beräkningen kommer alltså ha ett fel som kan variera även om vi använder samma antal slumptal. För att då få en feluppskattning av integralberäkningen kan man använda sig av konfidensintervall, beskriv vad det är och hur det kan användas för feluppskattning.

Lösning: Ett konfidensintervall bestäms för en given konfidensgrad. Om man gör 100 mätserier (varje serie med många mätningar/simuleringar) och för varje serie beräknar medelvärdet och ett konfidensintervall (med konfidensgrad 95%) så kommer 95 av dessa intervall innehålla det korrekta medelvärdet. Konfidensintervallet, med en konfidensgrad på 95%, kan beräknas som:

$$CI = \bar{I} \pm 1.96 \frac{s}{\sqrt{M}},$$

där $\bar{I} = \frac{1}{M}\sum_i^M I_i$ och $s^2 = \frac{1}{M-1}\sum_i^M (\bar{I} - I_i)^2$ är respektive medelvärde och varians för M stycken simuleringar.

(c) Storleken på felet (feluppskattningen) i integralberäkningen kommer att bero av antalet punkter N som $1/\sqrt{N}$, dvs vi har noggrannhetsordning 0.5. Antag att vi vill förbättra lösningen från uppgift (a) med en faktor 1000, uppskatta hur många slumptal N vi skulle behöva använda då.

Lösning: Givet att

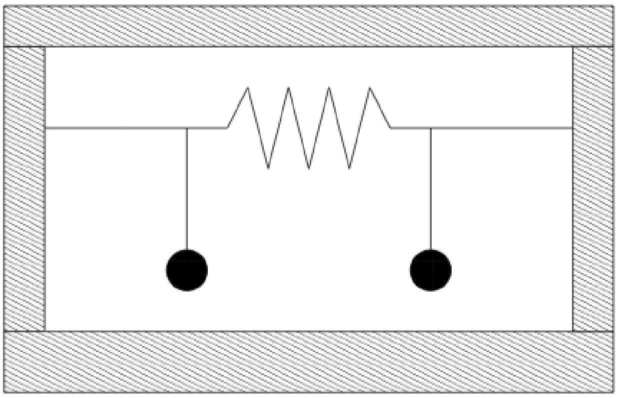
$$\text{error}(N) \propto \frac{1}{\sqrt{N}}$$

I uppgift *a)* använde vi $N = 10$ slumptal. Vi vill hitta ett nytt antal slumptal, N', så att felet har minskat med en faktor 1000. Vi ställer upp ekvationen:

$$\begin{aligned} \frac{\text{error}(10)}{\text{error}(N')} &= 10^3 \\ \Rightarrow \sqrt{\frac{N'}{10}} &= 10^3 \\ \Rightarrow N' = 10 \cdot (10^3)^2 &= 10^7. \end{aligned}$$

Svar: Vi behöver ca 10^7 antal slumptal för att minska felet med en faktor 10^3.

12. Differentialekvationer i riktiga tillämpningar innehåller ofta ett antal parametrar. Värdena på parametrarna kan vara svåra att bestämma exakt och kan vara störda av tex mätfel. Ett sådant exempel är väderprognoser där insamlingen av väderdata kan innehålla små fel av olika slag som är slumpmässigt fördelade. För att uppskatta säkerheten i en väderprognos lägger man då till små slumpmässiga störningar till parametrarna och beräknar prognosen upprepade gånger. Man gör s.k. *ensemble prognoser* och tittar på spridningen av dessa. Vi ska i den här uppgiften göra ensemble



Figur 2: *Dubbelpendel med fjäder.*

prognoser för ett modellproblem. Som exempel kan vi betrakta följande tillämpning. Här har vi två pendlar med samma massa och samma längd som är sammankopplade med en fjäder (se Figur 2).

Rörelseekvationerna för de två pendlarna kan beskrivas med differentialekvationerna

$$\theta_1'' + \sin(\theta_1) + \alpha(\theta_1 - \theta_2) = 0$$

$$\theta_2'' + \sin(\theta_2) - \alpha(\theta_1 - \theta_2) = 0$$

Här motsvarar θ_1 och θ_2 respektive vinkel för pendlarna och α en konstant som är beroende av längden och massan på pendeln samt av fjäderkonstanten hos fjädern mellan pendlarna. Parametern α inte är känd exakt och likaså har vi inte heller exakta värden på vinklarna θ_1 och θ_2 vid start. Din uppgift blir nu att lösa rörelseekvationerna för pendlarna med upprepade störningar i parametrarna α, θ_1 och θ_2. Du ska sedan beräkna väntevärde och standardavvikelse för $\theta_1(T)$ vid en sluttid T=10. Sätt också upp 95% konfidensintervall för $\theta_1(T)$. För att illustrera fördelningen av $\theta_1(T)$ rita upp ett histogram. Du ska också rita upp de olika lösningarna över alla t från t=0 till t=T i en figur för att få en uppfattning om prognosens säkerhet.

Antag att systemet startar från vila och använd följande variation på parametrarna:

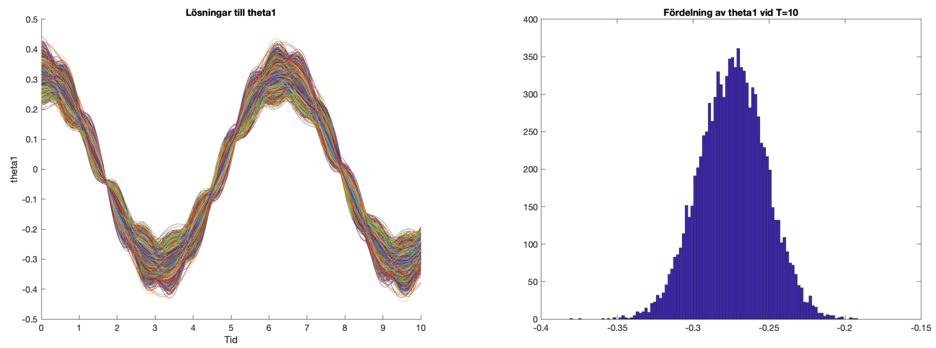
$\alpha \in N(10, 0.1)$ *(Normalfördelat med väntevärde 10 och standardavvikelse 0.1)*
$\theta_1(0) \in N(\pi/10, \pi/100)$
$\theta_2(0) \in N(\pi/10, \pi/100)$

Skriv ett Matlabscript som löser uppgiften. Du kan använda Matlabs inbyggda funktioner för ODE-lösning och slumptalsgenerering. För betyg 4 räcker det med en algoritmbeskrivning men för betyg 5 bör det vara ett detaljerat och (nästan) körbart program med motivering till val av metoder, tex varför du väljer en viss ODE-lösare.

Lösning:

```
%=================================================================
% Losning tentamen 2021-10-26, Ensemble Prognoser dubbelpendel
% Problemet ar icke-styvt, noggrannheten bestammer steglangd
% Valj explicit metod ode45
%=================================================================
clear; close all;
T=10;
N=10000;
hold on;
for i=1:N
    theta1=pi/10+pi/100*randn();
    theta2=pi/10+pi/100*randn();
    alpha=10+0.1*randn();
    u0=[theta1 0 theta2 0]';
    [t,u]=ode45(@(t,u)tentaODE(t,u,alpha),[0 T],u0);
    theta1_end(i)=u(end,1);
    plot(t,u(:,1));
end
xlabel('Tid')
ylabel('theta1');
title('Losningar till theta1')
theta1m=mean(theta1_end);
theta1s=std(theta1_end);
e1=1.96*theta1s/sqrt(N);
disp(['theta1: ' num2str(theta1m) '  ' num2str(theta1s)]);
disp(['Konfidensintervall: [' num2str(theta1m-e1) ' , ...
      ' num2str(theta1m+e1) ']' ]);
figure(2);
hist(theta1_end,100)
title('Fordelning av theta1 vid T=10');

function uprim=tentaODE(t,u,alpha)
uprim=zeros(4,1);
uprim(1)=u(2);
uprim(2)=-sin(u(1))-alpha*(u(1)-u(3));
uprim(3)=u(4);
uprim(4)=-sin(u(3))+alpha*(u(1)-u(3));
end
```



Figur 3: *Lösning, dubbelpendel med fjäder. Störningarna är stabila och påverkar inte lösningen i slutpunkten mer än själva störningen, vi har en säker prognos.*