Uppsala universitet
Institutionen för informationsteknologi
Beräkningsvetenskap

Tentamen i *Numeriska metoder och simulering* 5.0 hp, 2017-10-26

Skrivtid: $08^{00} - 13^{00}$
Hjälpmedel: En formelsamling ingår i detta uppgiftshäfte. Inga övriga hjälpmedel tillåts.

En komplett lösning ska innehålla utförliga resonemang samt motivering till alla svar.

För betyg 3 krävs: Att man klarar varje delmål för betyg 3 nedan.
För betyg 4/5 krävs: Att man klarar både betyg 3 och uppgiften för betyg 4/5

Uppgifter som testar måluppfyllelse för betyg 3

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

För att visa att du nått delmålet behöver du klara minst en av deluppgifterna i uppgift 1.

1. (a) Skriv ett program i Matlab för att med Matlab-kommandot `ode45` lösa differentialekvationen $y'(t) - 3y(t)\,(1 - y(t)) = 0$, för $0 \leq t \leq 10$, med $y(0) = 0.1$.

   (b) Antag att du har tillgång till en Matlab-funktion `g(x)` som beräknar värdet av funktionen g i punkten x. Skriv ett program i Matlab som använder en Monte Carlo-metod för att beräkna $\int_0^1 g(x)\mathrm{dx}$.

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

För att visa att du har nått delmålet behöver du klara minst två av deluppgifterna i uppgift 2.

2. Nedan finner du förklaringar av fyra begrepp som har ingått i kursen. Ange för varje förklaring vilket begrepp det är som avses.

   (a) En algoritm utan slumpmoment, så att utdata beror entydigt på indata.

   (b) Konventionen att i flyttalsrepresentation ha precis en nollskild siffra före "decimalpunkten" (så att flyttalsrepresentationen blir entydig).

   (c) En ODE för vilken en explicit differensmetod behöver ta mycket kortare steg för stabilitet än vad som skulle krävas för tillräcklig noggrannhet.

   (d) En numerisk metod för lösning av ODE, där högerledet i metoden innehåller y_{i+1}, så att man måste lösa en ickelineär ekvation i varje steg.

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

För att visa att du nått delmålet behöver du klara minst en av deluppgifterna i uppgift 3.

3. (a) Ställ upp Heuns metod för ekvationen i uppgift 1. (OBS! Du behöver inte genomföra några räkningar utan uppgiften är att skriva upp *formeln* för Heuns metod när den tillämpas på just denna ekvation.)

   (b) Om vi i uppgift 1b ovan hade velat beräkna en integral över intervallet från a till b , så skulle vi ha behövt generera likformigt fördelade slumptal i detta intervall. Fördelningsfunktionen för likformig fördelning på det intervallet är:

$$F(x) = \begin{array}{cl} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{array}$$

   Givet denna funktion kan algoritmen *Inverse Transform Sampling* användas för att sampla likformigt fördelade slumptal i intervallet. Första steget i algoritmen är att generera ett slumptal u ur den likformiga fördelningen på intervallet 0 till 1. Ett slumptal x ur den likformiga fördelningen på intervallet a till b kan sedan beräknas: $x = a + (b - a)u$. Visa hur man med *Inverse Transform Sampling*, givet $F(x)$ och u, kommer fram till denna formel för x. (*OBS!* Man kan tänka ut formeln utan att använda algoritmen, men här måste du använda algoritmen för att visa att du behärskar den.)

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

För att visa att du nått delmålet behöver du klara minst en av deluppgifterna i uppgift 4.

4. (a) Genomför analys för att visa att implicita Eulers metod har noggrannhetsordning 1.

   (b) Du använder en Monte Carlo-metod med 2000 slumpmässigt valda beräkningspunkter för att beräkna ett närmevärde till en integral. Hur mycket längre skulle exekveringstiden för beräkning av integralen ha blivit om du i stället hade använt 4000 punkter? Kom ihåg att motivera svaret tydligt.

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

För att visa att du nått delmålet behöver du klara minst en av deluppgifterna i uppgift 5.

5. (a) Du ska simulera ett system av kemiska reaktioner i en kemisk-teknisk process, där kemikalierna förekommer i stora mängder och är väl omblandade med varandra. Är det för denna simulering lämpligast att utgå från en deterministisk eller en stokastisk modell? Ge argument för ditt svar. (OBS! Avsikten med den här uppgiften är att testa att du kan argumentera för val av modell, så **det är viktigt att du verkligen åstadkommer en tydlig argumentation, som övertygar läsaren om att det slags modell du förespråkar skulle vara lämpligast.**)

   (b) Du tänker skriva ett program för simulering där du utgår från en ODE-modell. Programmet ska användas i ett projekt där det kommer att göras många olika simuleringar med olika värden på modellparametrarna. Därför är det viktigt att programmets exekveringstid blir så kort som möjligt och samtidigt måste den tolerans som du har valt givetvis uppfyllas. För de parameterkombinationer som ska användas kommer ODE-systemet inte att vara styvt. Vilken av klassiska Runge-Kuttas metod och Heuns metod skulle du välja för denna simulering och varför? (OBS! Avsikten med den här uppgiften är att testa att du kan argumentera för val av metod, så **det är viktigt att du verkligen åstadkommer en tydlig argumentation, som övertygar läsaren om att den metod du förespråkar skulle ge kortare exekveringstid**.)

Uppgift som testar måluppfyllelse för betyg 4

6. På 1920-talet arbetade fysikern Balthasar Van der Pol hos Philips Research Labs. I sitt arbete där kring teori för elektriska kretsar formulerade han en matematisk modell som har blivit känd som Van der Pols ekvation:

   $$x''(t) - \mu(1 - x(t)^2)x'(t) + x(t) = 0,$$

   där $\mu > 0$ är en parameter. För ökande värden på μ blir ekvationen alltmera styv.

   Din uppgift är nu att skriva ett Matlab-program för att lösa Van der Pols ekvation. Begynnelsevärden ska vara givna för $t = 0$ och simuleringen ska pågå till $t = T$. Värdet på parametern μ, sluttiden T samt begynnelsevärden ska matas in interaktivt när programmet körs. Resultatet av simuleringen ska presenteras grafiskt.

   För att lösningen ska godkännas behöver du också ge argument för att ditt program skulle vara lämpligt för parametervärden som gör att modellen är styv. (OBS! Precis som i argumentationsuppgifterna för betyg 3 är det viktigt att du åstadkommer en tydlig argumentation, som gör att läsaren känner sig övertygad om att ditt program skulle vara lämpligt under den givna förutsättningen.)

Uppgift som testar måluppfyllelse för betyg 5

7. I en central för telefonbokning av taxi sitter ett antal personer ("bokare") och hjälper kunder som ringer för att boka taxi. Antalet kunder är mycket större än antalet bokare, så när en ny kund ringer är vanligen alla bokare upptagna. Den första bokare som blir ledig tar hand om den nya kunden.

I en enkel stokastisk modell för telefonbokningen tänker vi oss att varje bokare tilldelas ett effektivitetsvärde, så att bokare nummer i har effektivitetsvärde E_i. Ett exempel: en bokare med effektivitetsvärde 1.9 kommer i genomsnitt att utföra bokningar snabbare än en bokare med effektivitetsvärde 1.2. I vår stokastiska modell tänker vi oss att effektivitetsvärdet E_i genom lämplig omskalning ska översättas till ett värde på sannolikheten P_i för att bokare nummer i är den som först blir ledig och kan ta hand om nästa kund.

Din uppgift är nu att skriva en Matlab-funktion för stokastisk simulering av ett arbetspass i bokningscentralen, baserat på ovanstående modell. Inparametrar till funktionen ska vara antal kunder (m) som ringer under arbetspasset, antal bokare (n; bokarna numreras från 1 till n) samt en vektor med effektivitetsvärdena E_i för de olika bokarna. Utparameter från funktionen ska vara en vektor `antal` med n värden, där `antal(i)` är det antal kunder som bokare nummer i sammanlagt tog hand om under arbetspasset.

Algoritmen för simuleringen ska i grova drag vara följande. Vi tänker oss att kunderna ringer in en i taget. För varje kund: slumpa fram numret i på den bokare som först blir ledig och kan ta hand om kunden. Öka `antal(i)`.

När numret på nästa bokare slumpas fram ska det göras enligt den sannolikhetsfördelning som motsvarar de ovan nämnda sannolikheterna P_i.

För att lösningen ska godkännas ska du inte bara skriva Matlab-funktionen utan även en text där du förklarar och motiverar din lösning.

Uppsala universitet
Institutionen för informationsteknologi
Avd. för beräkningsvetenskap

Blandade formler i Beräkningsvetenskap I och II

1. Flyttal och avrundningsfel

Ett flyttal $fl(x)$ representeras enligt

$$fl(x) = \hat{m} \cdot \beta^e, \quad \hat{m} = \pm(d_0.d_1 d_2, \ldots, d_{p-1}), \quad 0 \leq d_i < \beta, \quad d_0 \neq 0, \quad L \leq e \leq U,$$

där β betecknar bas och p precision.
Ett flyttalssystem defineras $FP(\beta, p, L, U)$.
Maskinepsilon (avrundningsenheten) $\epsilon_M = \frac{1}{2}\beta^{1-p}$ och kan defineras som det minsta tal ϵ sådant att $fl(1 + \epsilon) > 1$.

2. Linjära och ickelinjära ekvationer

Newton-Raphsons metod: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
För system: $x_{k+1} = x_k - [F']^{-1}F(x_k)$, där x_k och $F(x_k)$ är vektorer och F' är Jacobianen.
Fixpunktsiteration för $x = g(x)$: $x_{k+1} = g(x_k)$
Konvergenskvot, konvergenshastighet

$$\lim_{k \to \infty} \frac{|x_{k+1} - x_*|}{|x_k - x_*|^r} = C,$$

där C är en konstant, och r anger konvergenshastigheten ($r = 1$ betyder t ex linjär konvergens).
Allmän feluppskattning

$$|x_k - x^\star| \leq \frac{|f(x_k)|}{\min |f'(x)|}$$

Konditionstalet $\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$ mäter känsligheten för störningar hos ekvationssystemet $Ax = b$. Det gäller att

$$\frac{\|\Delta x\|}{\|x\|} \leq \mathrm{cond}(A) \frac{\|\Delta b\|}{\|b\|},$$

där $\Delta x = x - \hat{x}$ och $\Delta b = b - \hat{b}$.
Normer (vektor- respektive matrisnorm)

$$\| x \|_2 = \sqrt{|x_1|^2 + \ldots + |x_n|^2} \qquad \| x \|_1 = \textstyle\sum_i |x_i| \qquad \| x \|_\infty = \max_i\{|x_i|\}$$
$$\| A \|_1 = max_j(\textstyle\sum_i |a_{ij}|) \qquad \| A \|_\infty = max_i(\textstyle\sum_j |a_{ij}|)$$

3. Approximation

Newtons interpolationspolynom $p(x)$ då vi har n punkter $(x_1, y_1), \ldots (x_n, y_n)$ bygger på ansatsen

$$p(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \ldots + a_{n-1}(x - x_1) \cdots (x - x_{n-1})$$

Minstakvadratapproximationen till punktmängden $(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)$ med ett n:egradspolynom $p(x) = a_0 \cdot 1 + a_1 \cdot x + \ldots + a_n \cdot x^n$ kan formuleras som ett överbestämt ekvationssystem $Ax = b$, där A är $m \times n$, $m > n$. Minstakvadratlösningen kan fås ur normalekvationerna

$$A^T A x = A^T b$$

4. Ordinära differentialekvationer

Eulers metod (explicit Euler): $y_{k+1} = y_k + hf(x_k, y_k)$, n.o. 1
Implicit Euler (Euler bakåt): $y_{k+1} = y_k + hf(x_{k+1}, y_{k+1})$, n.o 1
Trapetsmetoden: $y_{k+1} = y_k + \frac{h}{2}(f(x_k, y_k) + f(x_{k+1}, y_{k+1}))$, n.o. 2
Heuns metod (tillhör gruppen Runge-Kuttametoder):

$$\begin{cases} K_1 = f(x_k, y_k) \\ K_2 = f(x_{k+1}, y_k + hK_1) \\ y_{k+1} = y_k + \frac{h}{2}(K_1 + K_2) \end{cases}$$

n.o. 2
Klassisk Runge-Kutta:

$$\begin{cases} K_1 = f(x_k, y_k) \\ K_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_1) \\ K_3 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_2) \\ K_4 = f(x_{k+1}, y_k + hK_3) \\ y_{k+1} = y_k + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

n.o. 4

5. Numerisk integration

Trapetsformeln
Beräkning på ett delintervall med steglängd $h_k = x_{k+1} - x_k$

$$\int_{x_k}^{x_{k+1}} f(x)\,dx = \frac{h_k}{2}[f(x_k) + f(x_{k+1})]$$

Sammansatt formel på helt intervall $[a\ b]$, då ekvidistant steglängd $h = h_k$:

$$\int_a^b f(x)\,dx \approx \frac{h}{2}[f(x_0) + 2f(x_1) + \ldots + 2f(x_{N-1}) + f(x_N)]$$

Diskretiseringsfelet R på helt intervall $[a \;\; b]$, dvs $\int_a^b f(x)\,dx = T(h) + R$ är

$$R = -\frac{(b-a)}{12} h^2 f''(\xi).$$

Funktionsfelet (övre gräns): $(b-a) \cdot \epsilon$, där ϵ är en övre gräns för absoluta felet i varje funktionsberäkning.

Simpsons formel

Beräkning på ett dubbelintervall med steglängd h

$$\int_{x_k}^{x_{k+2}} f(x)\,dx = \frac{h}{3}[f(x_k) + 4f(x_{k+1}) + f(x_{k+2})]$$

Sammansatt formel på helt intervall $[a\; b]$, då ekvidistant steglängd $h = h_k$:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)]$$

Diskretiseringsfelet R på helt intervall $[a \;\; b]$, dvs $\int_a^b f(x)\,dx = S(h) + R$ är

$$R = -\frac{(b-a)}{180} h^4 f''''(\xi).$$

Funktionsfelet: Samma som för trapetsformeln, se ovan.

6. Richardsonextrapolation

Om $F_1(h)$ och $F_1(2h)$ är två beräkningar (t ex ett steg i en beräkning av en integral eller en ODE) med en metod av noggrannhetsordning p med steglängd h respektive dubbel steglängd $2h$ så är

$$R(h) = \frac{F_1(h) - F_1(2h)}{2^p - 1}$$

en uppskattning av den ledande termen i trunkeringsfelet i $F_1(h)$. Kan även användas för att förbättra noggrannheten i $F_1(h)$ genom

$$F(h) = F_1(h) + \frac{F_1(h) - F_1(2h)}{2^p - 1}.$$

7. Numerisk derivering

För numerisk derivering används s k differensformler

$$\begin{aligned}
f'(x) &\approx \tfrac{f(x+h)-f(x-h)}{2h}, \quad \text{centraldifferens} \\
f'(x) &\approx \tfrac{f(x+h)-f(x)}{h}, \quad \text{framåtdifferens} \\
f'(x) &\approx \tfrac{f(x)-f(x-h)}{h}, \quad \text{bakåtdifferens} \\
f''(x) &\approx \tfrac{f(x+h)-2f(x)+f(x-h)}{h^2}
\end{aligned}$$

8. Monte Carlometoder

Den övergripande strukturen för Monte Carlosimuleringar är

```
Indata N (antal försök)
for i = 1:N
   Utför en stokastisk simulering
   resultat(i) = resultatet av simuleringen
end
Slutresultat genom någon statistisk beräkning,
t ex medelvärdet mean(resultat)
```

Felet i Monte Carlometoder är $\mathcal{O}(\frac{1}{\sqrt{N}})$, där N är antal samplingar.
Kumultativ fördelningsfunktion: $F(x) = \int_{-\infty}^{x} f(y)dy$

Normalfördelning

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Aritmetiskt medelvärde baserat på N realisationer x_i av slumpvariablen X: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

9. Taylorutveckling

Taylorutveckling av $y(x_k + h)$ kring x_k:

$$y(x_k + h) = y(x_k) + hy'(x_k) + \frac{h^2}{2!}y''(x_k) + \frac{h^3}{3!}y'''(x_k) + \mathcal{O}(h^4)$$