Uppsala universitet
Institutionen för informationsteknologi
Beräkningsvetenskap

Tentamen i *Numeriska metoder och simulering* 5.0 hp, 2019-10-25

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 en uppgift 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 har nått delmålet behöver du klara minst en av deluppgifterna i uppgift 1.

1. (a) Skriv ett program i Matlab för att med `ode45` lösa differentialekvationen $y'(t)\sqrt{t+1} + \sin(t + y(t))^3 = 0$, för $0 \le t \le T$, med $y(0) = a$. Sluttiden T och begynnelsevärdet a ska matas in interaktivt av användaren när programmet körs.

   (b) Antag att du har tillgång till en matlabfunktion `dygnslangd(delta)` som utför stokastisk simulering av dygnsrytm med Gillespies algoritm. Värdet på inparametern `delta` påverkar dygnets längd. Varje gång `dygnslangd` anropas simuleras ett dygn och dygnets längd returneras som utparameter. Skriv ett program i Matlab som använder funktionen `dygnslangd` i en Monte Carlo-metod för att beräkna genomsnittlig dygnslängd då värdet på `delta` är 0.16.

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) Förlust av signifikanta siffror vid subtraktion mellan jämnstora flyttal.

   (b) Att den numeriska metoden i varje steg automatiskt ställer in steglängden så att det uppskattade felet blir lägre än toleransen.

   (c) Den numeriska metoden går mot differentialekvationen då h går mot noll.

   (d) En modell med slumpmoment, så att utdata inte beror entydigt på indata.

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

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

3. (a) Visa med handräkning hur det går till att med explicita Eulers metod, steglängd $h = 0.1$, beräkna ett närmevärde till $y(0.1)$, där $y(t)$ är lösningen till begynnelsevärdesproblemet

$$y'(t) + \frac{2y(t)}{1 + 0.1y(t)} = 0, \qquad y(0) = 10.$$

   (b) I en stokastisk simulering av köbildning vid trafikljus vill du använda *Inverse Transform Sampling* för att slumpa fram en färg, där det finns tre möjliga utfall, 1: röd, 2: gul och 3: grön. Sannolikheterna för de tre utfallen ska vara: $P_1 = 0.45$, $P_2 = 0.10$, $P_3 = 0.45$. Som första steg i algoritmen slumpar du fram ett tal u. Antag att värdet på u blev 0.61. Visa med handräkning hur algoritmen fortsätter i detta fall och vilken färg som blir resultatet. Det måste framgå tydligt hur du kommer fram till slutsatsen.

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 har 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 och exekveringstiden blir ca 1 minut. Nu vill du ha ett noggrannare resultat och gör därför en ny körning, där du fördubblar antalet upprepade stokastiska simuleringar. Ungefär hur lång blir då exekveringstiden. 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 har nått delmålet behöver du klara minst en av deluppgifterna i uppgift 5.

OBS! Med tanke på vad det här delmålet handlar om är det ***viktigt att du verkligen åstadkommer en tydlig argumentation, som övertygar läsaren om att det alternativ du förespråkar skulle vara lämpligast.***

5. (a) I kursen har vi simulerat smittspridning med SIR-modellen. Tänk dig ett scenario där den modellen ska användas för att planera ett vaccinationsprogram, så att vi ska göra ett stort antal simuleringar där den modellparameter som har med vaccination att göra varieras. Vilken av Heuns metod och Trapetsmetoden skulle vara bäst att använda för dessa simuleringar och varför? Vi antar att modellen inte är styv.

   (b) 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?

Uppgifter som testar måluppfyllelse för högre betyg

6. Den här uppgiften har en del för betyg 4 och en annan del för betyg 5.

   För betyg 4 ska du implementera Heuns metod som en funktion `heun` och dessutom skriva ett script som testar funktionen.

   Anropet till `heun` ska vara:

   ```
   [t, y] = heun(odefun, tspan, y0, h)
   ```

   Utparametrarna är desamma som för Matlabs ODE-lösare (`ode45` etc.). Inparametrarna `odefun`, `tspan`, `y0` är samma som motsvarande inparametrar till Matlabs ODE-lösare och `h` är steglängden. (Vi använder konstant steglängd här.)

   Du ska skriva funktionen `heun` och dessutom ett script som testkör funktionen på ett problem med känd lösning. Som testproblem ska du använda $y'(t) = -y(t)$ med begynnelsevärde $y(0) = 1$, så att den exakta lösningen är $y(t) = e^{-t}$.

   Ditt testscript ska lösa testproblemet med `heun` och sedan plotta den numeriska lösningen som diskreta punkter samt visa den exakta lösningen som en heldragen kurva i samma plot. Det ska också finnas förklarande text i figuren, som säger vilken lösning respektive graf visar.

   För betyg 5 behöver behöver du inte implementera Heuns metod. I stället ska du anta att du har *använt* funktionen `heun` i en simulering där du löste ett icke-styvt ODE-problem med steglängden vald så att felet blev i storleksordningen 10^{-4} och att exekveringstiden då blev 20 sekunder. Din uppgift är att svara på frågan: ungefär hur lång skulle exekveringstiden ha blivit om vi i stället hade velat ha ett fel i storleksordningen 10^{-6}? För godkänt krävs att svaret underbyggs med en teoretisk analys.

7. Även den här uppgiften har en del för betyg 4 och en annan del för betyg 5. Läs först nedanstående bakgrundsbeskrivning.

   Bakgrund: Numerisk simulering innehåller flera osäkerhetsfaktorer. Ett exempel är mätfel i begynnelsevärdena. För att få en uppfattning om vilken inverkan dessa mätfel har på den numeriska lösningen kan man göra en *ensemblesimulering*. Det innebär att man gör simuleringen upprepade gånger. I varje ny simulering gör man en slumpmässig störning av de givna begynnelsevärdena/mätvärdena. När alla körningarna är genomförda gör man en statistisk utvärdering av hur väl lösningarna från de olika simuleringarna överensstämde med varandra. I den här uppgiften ska vi tänka oss att resultatet av ensemblesimuleringen är medelvärdet av de olika simuleringarnas lösningsvärde i sluttidpunkten.

För betyg 4 ska du tänka dig att du har gjort en ensemblesimulering där de givna begynnelsevärdena stördes slumpmässigt 500 gånger (och att det i ensemblesimuleringen alltså gjordes 500 enstaka simuleringar). Exekveringstiden blev då 2 minuter. Din uppgift är att svara på frågan: ungefär hur lång skulle exekveringstiden ha blivit om du i stället hade stört begynnelsevärdena ännu flera gånger, så att den övre gränsen för felet i slutresultatet från ensemblesimuleringen hade blivit hälften så stor? För godkänt krävs att svaret underbyggs med en detaljerad motivering.

För betyg 5 ska du implementera ensemblesimulering i Matlab som en funktion `ensemble`. Den ska vara lämplig för problem som *inte* är styva. Vi begränsar oss till skalära ODE-problem och den statistiska utvärderingen begränsas till beräkning av ett medelvärde. Anropet till `ensemble` ska vara:

```
ymean = ensemble(odefun, tspan, y0, tol, nsim, eps)
```

där `odefun`, `tspan`, `y0` är samma som motsvarande inparametrar till Matlabs ODE-lösare och `tol` är toleransen för det relativa felet. Funktionen `ensemble` ska göra `nsim` stycken upprepade simuleringar. I simulering nummer i ska begynnelsevärdet vara $\mathtt{y}_0 + \varepsilon_i$, där ε_i är ett slumptal ur den likformiga fördelningen på intervallet `[-eps, eps]`. Utparametern `ymean` ska vara medelvärdet av de olika simuleringarnas lösningsvärde för den sluttidpunkt som anges i `tspan`.

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)

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

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

$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$, centraldifferens
$f'(x) \approx \frac{f(x+h)-f(x)}{h}$, framåtdifferens
$f'(x) \approx \frac{f(x)-f(x-h)}{h}$, bakåtdifferens
$f''(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2}$

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:

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