

Uppsala universitet  
Institutionen för informationsteknologi  
Beräkningsvetenskap

## Tentamen i *Numeriska metoder och simulering* 5.0 hp, 2018-10-25

**Skrivtid:** 08<sup>00</sup> – 13<sup>00</sup>

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

- (a) Skriv ett program i Matlab för att med `ode45` lösa differentialekvationen  $P'(t) - (k + 0.1 \sin(t))P(t)(C - P(t)) = 0$  för  $0 \leq t \leq 100$ , med  $P(0) = 10$ . Konstanterna  $k$  och  $C$  är modellparametrar och deras värden ska matas in interaktivt av användaren när programmet körs.
- (b) Skriv ett skript i Matlab, som beräknar  $\int_a^b f(x) dx$  med Monte Carlo-metod. Funktionen  $f(x)$  antas vara given som en matlab-funktion i filen `f.m`. Integrationsgränserna  $a$  och  $b$  ska användaren mata in interaktivt när programmet körs.

**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) Det som uppstår när vi försöker representera ett tal vars absolutbelopp är mindre än det till beloppet minsta normaliserade flyttalet.
  - (b) Upprepad stokastisk simulering samt statistik på de samlade resultaten.
  - (c) Den numeriska lösningen går mot den exakta lösningen då steglängden  $h$  går mot noll.
  - (d) En algoritm utan slumpmoment, så att utdata 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) Ställ upp Trapetsmetoden för begynnelsevärdesproblemet

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

Använd steglängden  $h = 0.1$  och *förklara* (utan att genomföra några beräkningar) hur det går till att utifrån den formel du har ställt upp räkna ut ett närmevärde till  $y(0.1)$ .

- (b) Fördelningsfunktionen för likformigt fördelade slumpstal på intervallet från  $a$  till  $b$  är:

$$F(x) = \begin{cases} 0 & \text{om } x < a \\ (x - a)/(b - a) & \text{om } a \leq x < b \\ 1 & \text{om } x \geq b \end{cases}$$

Givet detta funktionsuttryck kan den kontinuerliga versionen av algoritmen *Inverse Transform Sampling* användas för att härleda en formel för generering av likformigt fördelade slumpstal i intervallet från  $a$  till  $b$ . Genomför denna härledning och redovisa den resulterande formeln.

**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 stabilitetsanalys av implicita Eulers metod.
- (b) Du använder en Monte Carlo-metod med 50000 punkter för att beräkna en integral. Exekveringstiden blir ca 2 sekunder. Nu vill du göra en ny körning där du ökar antalet punkter så att felet halveras. Hur mycket längre kommer exekveringstiden då att bli? 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 löst ett ODE-system som modellerar reaktioner mellan tre kemikalier  $A$ ,  $B$  och  $C$  i laboratoriemiljö där stora mängder av kemikalierna är blandade i en glasbehållare. Vi såg att för vissa värden på modellparametrarna  $s$ ,  $q$  och  $w$  blev problemet styvt. Nu vill du göra en serie simuleringar där modellparametrarna har sådana värden. Du väljer mellan metoderna Euler bakåt och Trapetsmetoden. Vilken av dessa är lämpligast för dessa simuleringar och varför?
- (b) Beskriv en situation där reaktioner mellan tre kemikalier skulle beskrivas bättre med en stokastisk modell än med ett ODE-system och förklara varför ODE-modellen inte skulle vara lämplig i den situationen.

## Uppgift som testar måluppfyllelse för betyg 4

6. I den här uppgiften ska du simulera hur encelliga bakterier konsumerar glukos. Molekylära receptorer i bakteriens cellmembran "fångar" glykosmolekyler och ser till att de förs in i cellen. Om vi har en miljö med

stor mängd bakterier och hög koncentration av glukos kan bakteriepopulationens konsumtion av glukos beskrivas av följande ODE-system<sup>1</sup>:

$$\begin{aligned}\frac{dc(t)}{dt} &= -k_1rc(t) + (k_{-1} + k_1c(t))x(t) \\ \frac{dx(t)}{dt} &= k_1rc(t) - (k_{-1} + k_2 + k_1c(t))x(t)\end{aligned}$$

där  $c(t)$  är koncentrationen av "fria" glukosmolekyler på cellernas utsida och  $x(t)$  är koncentrationen av receptorer som har glukos bundet till sig. Modellparametern  $r$  är den totala koncentrationen av receptorer (med eller utan glukos). De övriga modellparametrarna ( $k_{-1}$ ,  $k_1$  och  $k_2$ ) är reaktionshastigheter för de kemiska reaktioner som äger rum när glukos fångas in och förs in i cellerna.

Din uppgift är att skriva ett program i Matlab för att simulera den ovan beskrivna processen. Begynnelsevärden ska vara givna för  $t = 0$  och simuleringen ska pågå till  $t = T$ . Sluttiden  $T$ , begynnelsevärdena, samt värden på modellparameterarna ska matas in interaktivt när programmet körs.

Resultatet av simuleringen ska presenteras grafiskt. Den figur som programmet genererar ska ha rubrik och axlarna i diagrammet ska ha etiketter. Vidare ska det finnas ledtext som talar om vad varje funktionskurva i diagrammet representerar.

För att lösningen ska godkännas ska du också ge argument för att ditt program skulle vara lämpligt för parametervärden som gör att modellen inte är styv. (OBS! Det räcker inte att *påstå* att programmet är lämpligt för sådana fall, utan påståendet måste underbyggas med övertygande argumentation/förklaring.)

## Uppgift som testar måluppfyllelse för betyg 5

### 7. Metoden

$$\begin{aligned}s_1 &= f(t_k, y_k) \\ s_2 &= f(t_k + h, y_k + hs_1) \\ s_3 &= f(t_k + h, y_k + \frac{h}{2}(s_1 + s_2)) \\ y_{k+1} &= y_k + h \left( \frac{2}{3}s_1 + \frac{1}{6}s_2 + \frac{1}{6}s_3 \right)\end{aligned}\tag{1}$$

---

<sup>1</sup>Se avsnitt 7.1 i Leah Edelstein-Keshet, "Mathematical Models in Biology", SIAM, 2005

har noggrannhetsordning 1. Notera att  $s_1$  och  $s_2$  ingår i Heuns metod och att i  $s_3$  ingår det  $y_{k+1}$ -värde som Heuns metod skulle ge. Metoden (1) och Heuns metod utgör därför ett par av Runge-Kutta-metoder av samma typ som de par som ingår i Matlabs `ode45` respektive `ode23`.

Skriv baserat på (1) och Heuns metod en matlabfunktion `ode12`, som fungerar enligt samma princip som `ode45` och `ode23`. Signaturen till `ode12` ska vara: `ode12(odefun,tspan,y0,tol)`, där de tre första inparametrarna har samma betydelse som i Matlabs inbyggda ODE-lösare och `tol` är toleransen för det absoluta felet i den numeriska lösningen.

För godkänd lösning krävs att funktionen `ode12` i allt väsentligt fungerar i princip som `ode45` och `ode23` (men den behöver inte vara lika genomarbetad som dessa).

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

$$fl(x) = \hat{m} \cdot \beta^e, \quad \hat{m} = \pm(d_0.d_1d_2, \dots, d_{p-1}), \quad 0 \leq d_i < \beta, \quad d_0 \neq 0, \quad L \leq e \leq U,$$

där  $\beta$  betecknar bas och  $p$  precision.

Ett flyttalssystem definieras  $FP(\beta, p, L, U)$ .

Maskinepsilon (avrundningsenheten)  $\epsilon_M = \frac{1}{2}\beta^{1-p}$  och kan definieras som det minsta tal  $\epsilon$  sådant att  $fl(1 + \epsilon) > 1$ .

### 2. Linjära och icke linjära 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*

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

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

*Allmän feluppskattning*

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

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

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

där  $\Delta x = x - \hat{x}$  och  $\Delta b = b - \hat{b}$ .

*Normer (vektor- respektive matrisnorm)*

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

### 3. Approximation

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

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

*Minstakvadratapproximationen* till punktmängden  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$  med ett  $n$ :egradspolynom  $p(x) = a_0 + a_1x + \dots + a_nx^n$  kan formuleras som ett överbestämt ekvationssystem  $Ax = b$ , där  $A$  är  $m \times n$ ,  $m > n$ . Minstakvadratlösningen kan fås ur normalekvationerna

$$A^T Ax = A^T b$$

### 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):

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

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  $h_k = x_{k+1} - x_k$

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

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

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

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  $\epsilon$  är en övre gräns för absoluta felet i varje funktionsberäkning.

*Simpsons formel*

Beräkning på ett dubbelintervall med steglängd  $h$

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

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

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

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 trunckeringsfelet 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

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



## 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.

Kumulativ 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 slumpvariabeln  $X$ :  $\mu = \frac{1}{N} \sum_{i=1}^N x_i$ .

## 9. Taylorutveckling

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

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