

Lösningar till tentamen i *Numeriska metoder och simulering* 5.0 hp, 2016-01-16

Uppgifter som testar måluppfyllelse för betyg 3

1. (a) En funktion som beräknar ODE-problemets högerled:

```
function yprim = myODE( t, y)
    yprim = cos(t+y);
end
```

Exempel på ett skript som löser problemet med de indata som angavs i uppgiften:

```
[t, y] = ode45(@myODE, [0 2], 0.5);
plot(t,y)
```

- (b) Exempel på ett program enligt uppgiften:

```
N = input('Antal simuleringar av aktiekurs? ');
x0 = input('Dagens aktiekurs? ');
T = input('Antal dagar simuleringen ska omfatta? ');
resultat = zeros(1,N);

for i = 1:N
    resultat(i) = aktiekurs(x0,T);
end

kurs = mean(resultat);
disp(['Genomsnittlig aktiekurs efter '...
    num2str(T), ' dagar: ' ...
    num2str(kurs) ' kronor' ])
```

2. (a) Stokastisk modell

- (b) Differensmetod
- (c) Maskinepsilon
- (d) Konvergens

3. (a)

$$y_{i+1} = y_i + (h/2) (\cos(t_i + y_i) + \cos(t_{i+1} + y_{i+1}))$$

(b) Givet: tillstånd x och sluttid T

```
t = 0;
while t < T
    Slumpa tid tau till nästa reaktion
    Slumpa vilken nästa reaktion ska bli
    Uppdatera tillståndet enligt nästa reaktion
    t = t + tau
```

4. (a) Se föreläsningssanteckningar.

(b) Euler framåt tillämpad på testekvationen ($y'(t) = \lambda y(t)$) är:

$$y_{k+1} = y_k + h\lambda y_k. \text{ Omformulering ger att } y_{k+1} = y_k(1 + \lambda h).$$

Kom ihåg att i testekvationen står y för störningen. För stabilitet krävs att störningen avtar med tiden, det vill säga att $|y_{k+1}| < |y_k|$. Med insättning av ovanstående uttryck för y_{k+1} blir kravet att $|y_k(1 + \lambda h)| < |y_k|$. Slutsatsen blir att stabilitetsvillkoret för Euler bakåt är:

$$|1 + \lambda h| < 1$$

5. (a) Om man tänker på de modeller som vi har använt i kursen, så är exekveringstiden för att simulera kemiska reaktioner kortare om man använder en deterministisk ODE-modell än om man gör stokastisk simulering. Om det slumpmässiga inslaget är försumbart är därför en deterministisk modell för de kemiska reaktionerna lämpligast. Under de förutsättningar som anges i uppgiften kan man anta att det vid varje tidpunkt sker många reaktioner mellan A- och B-molekyler (och inte enstaka reaktioner vid slumpmässiga tidpunkter som fallet kan vara om det är glest mellan molekylerna, exempelvis inuti en cell). Därför *kommer* slumpmässigheten att vara försumbar i det scenario som uppgiften avser. Slutsatsen blir att det är lämpligast med en deterministisk modell i detta fall.

- (b) Båda metoderna har noggrannhetsordning 2, så om man enbart ser till trunkeringsfelet skulle båda metoderna behöva använda ungefär samma antal beräkningspunkter för att uppfylla noggrannhetskravet. Men eftersom Heuns metod är explicit och ODE-modellen är styv skulle Heuns metod av stabilitetsskäl behöva ta väsentligt kortare steg (flera beräkningspunkter) än vad som krävs för att få ett tillräckligt litet trunkeringsfel. Trapetsmetoden är implicit och drabbas inte av stabilitetsproblem, utan kan ta steg av den storlek som räcker för att trunkeringsfelet ska bli mindre än toleransen. Därmed kommer Trapetsmetoden att behöva betydligt färre beräkningspunkter än Heuns metod i detta fall. Det kommer mer än väl att uppväga att den implicita metoden kräver mera arbete per beräkningspunkt än den explicita metoden. Slutsatsen blir att den av de två metoderna som kommer att behöva kortast exekveringstid för att genomföra simuleringen med önskad noggrannhet är Trapetsmetoden. Därför är den lämpligare än Heuns metod i detta fall.

Uppgift som testar måluppfyllelse för betyg 4

6. Nedan visas ett program som genomför simulering enligt uppgiften.

Högerledet i ODE-systemet beskrivs i funktionen `kemiODE`:

```
function yprim = kemiODE(t,y)

global s q w
yprim = [s*(y(2) - y(1)*y(2) + y(1) - q*y(1)^2);
         (y(3) - y(2) - y(1)*y(2))/s;
         w*(y(1) - y(3)) ];
```

Följande skript genomför simuleringen:

```
% Sätt start- och sluttid för simuleringen
starttid = 0;
sluttid = 10;
tidsintervall = [starttid sluttid];

% Sätt begynnelsevärden
alfa0 = input('Begynnelsevärde på alfa ? ');
beta0 = input('Begynnelsevärde på beta? ');
```

```

gamma0 = input('Begynnelsevärde på gamma? ');
y0 = [alfa0; beta0; gamma0];

% Sätt värden på parametrarna s, q, w
global s q w
s = input('Värde på s? ');
q = input('Värde på q? ');
w = input('Värde på w? ');

% Lös ode:n genom anrop av ode15s
[t, y] = ode15s(@kemiODE, tidsintervall, y0);

% Rita upp lösningen
plot(t,y);
xlabel('tid');
ylabel('Koncentration');
title('Koncentration av kemikalier');
legend('alpha','beta', 'gamma');

```

Programmet anropar Matlabs inbyggda ODE-lösare `ode15s`. Den använder en implicit metod och är därför lämplig när ODE-systemet är styvt.

Uppgift som testar måluppfyllelse för betyg 5

7.
 - Ett program som genomför simuleringen på det sätt som nämns i uppgiften kan se ut som programmet ovan (uppgift 6), men där anropet till `ode15s` byts mot följande:


```

if isStiff(@kemiODE)
    theta = ...
else
    theta = ...
end
[t, y] = thetmethod(@kemiODE, tidsintervall, y0, theta);

```
 - Att ODE-systemet är styvt innebär att en explicit metod behöver ta mycket korta steg av stabilitetsskäl. Det är därför olämpligt med en explicit metod om systemet är styvt. Valet $\theta = 0$ ger en explicit metod (Euler framåt) och är därför direkt olämpligt om

ODE-systemet är styvt.

- Valet $\theta = 0.5$ ger en metod av noggrannhetsordning 2 (övriga val av θ ger noggrannhetsordning 1). För att visa detta undersöker vi lokala trunkeringsfelet τ .

$$\begin{aligned}\tau &= y(t_i + h) - y(t_i) - h[(1 - \theta)y'(t_i) + \theta y'(t_i + h)] = \\ &= y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{6}y'''(t_i) + \mathcal{O}(h^4) - y(t_i) - h[(1 - \theta)y'(t_i) + \\ &\quad \theta(y'(t_i) + hy''(t_i) + \frac{h^2}{2}y'''(t_i) + \mathcal{O}(h^3))] = \\ &= h^2\left(\frac{1}{2} - \theta\right)y''(t_i) + h^3\left(\frac{1}{6} - \frac{\theta}{2}\right)y'''(t_i) + \mathcal{O}(h^4).\end{aligned}$$

Bästa val av θ är $\theta = 0.5$, ty då försvinner h^2 -termen. Då får vi $\tau = \mathcal{O}(h^3)$ och noggrannhetsordning 2. För $\theta \neq 0.5$ fås $\tau = \mathcal{O}(h^2)$ och noggrannhetsordning 1.

- När problemet inte är styvt kan alternativet $\theta = 0$ vara lämpligt. Det ger en explicit metod av noggrannhetsordning 1. Övriga val av θ , med undantag för $\theta = 0.5$, ger implicita metoder av noggrannhetsordning 1. Metoder av samma noggrannhetsordning bör behöva ungefär lika många tidssteg för att lösa uppgiften med önskad noggrannhet. Eftersom en explicit metod kräver mindre arbete per tidssteg än en implicit metod bör därför $\theta = 0$ vara lämpligare än alla de θ -värden som ger implicita metoder av noggrannhetsordning 1. Det enda återstående alternativet utöver $\theta = 0$ är då $\theta = 0.5$. Med $\theta = 0.5$ får vi en implicit metod, som alltså kräver betydligt mera arbete per beräkningspunkt än den metod vi får med $\theta = 0$. Å andra sidan har metoden med $\theta = 0.5$ noggrannhetsordning 2, vilket innebär att den för önskad noggrannhet kan ta färre steg än den metod som har noggrannhetsordning 1. Det är därför inte självklart vilken av de två metoderna $\theta = 0$ respektive $\theta = 0.5$ som är lämpligast.