Tentamen 2023-10-26, lösningsförslag

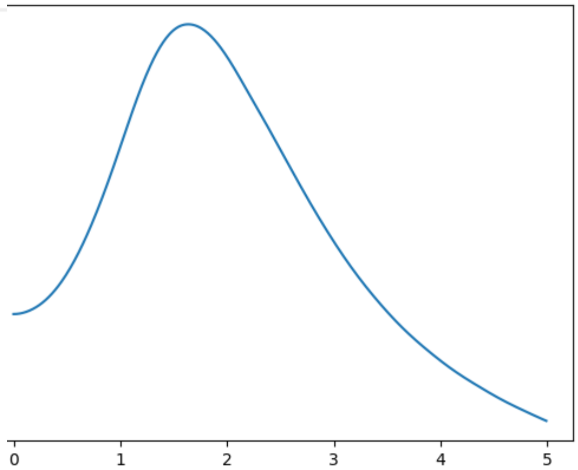
1)

```python
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

def ODE_fun(t, y):
    yprime = np.sin(y*t)
    return yprime

y0 = [1]
tspan = [0, 5]
eval = np.arange(tspan[0], tspan[1], 0.01)
result = solve_ivp(ODE_fun, tspan, y0, t_eval = eval)

plt.plot(result.t, result.y[0])
plt.show()
```



2) $my''(t)+cy'(t)+ky(t)=0 \Rightarrow y''(t)=(-cy'(t)-k\cdot y(t))/m$

skriv om på formen $\bar{y}'(t)=\bar{f}(t,\bar{y})$

$$u_1=y \Rightarrow u_1'=y'$$
$$u_2=y' \Rightarrow u_2'=y'' \Rightarrow \begin{cases} u_1'=u_2 \\ u_2'=(-c\cdot u_2-k\cdot u_1)/m \\ u_1(0)=0,\ u_2(0)=1 \end{cases}$$

```python
import numpy as np
from scipy.integrate import solve_ivp

def ODE_fun(t, u, m,k,c):
    u1, u2 = u
    u = [u2, (-c*u2 - k*u1)/m]
    return u

u0 = np.array([0,1]) # y(0) resp. y´(0)
tspan = [0, 2]
m = 2
k = 100
c = 0.1
eval = np.arange(tspan[0], tspan[1], 0.01)
result = solve_ivp(ODE_fun, tspan, u0, t_eval=eval, args=(m,k,c))
```

3/ a/ En stokastisk process som är "minnesfri" ("memoryless"): Nästa steg kan bestämmas enbart utgående från befintligt steg.

b/ Exempel: stokastisk SIR-modell eller stokastisk Pred-prey

c/ Exempel: Brownsk rörelse

d/ Exempel: "Hur långt i medeltal har en partikel rört sig efter t sekunder?"

4/ a/ Noggrannhetsordning är lutningen på linjen där diskretiseringsfelet dominerar (dvs för $h > 10^{-3}$)

b/ maskinepsilon $\varepsilon_m \approx 10^{-16}$ och är orsaken till att felet inte kan bli $< 10^{-16}$ (i bilden $\approx 10^{-15}$)

c/ man kan se att om felet minskar från t.ex 10^{-1} till 10^{-2} så förbättras felet från 10^{-6} till 10^{-10} $\Rightarrow$ minskning faktor 10 ger minskning av felet med faktor 10^4 $\Rightarrow$ n.o. 4 dvs klassisk Runge-kutta.

5/

```
N = ..., t.ex N = 1000
cost_a = 3
for i = 1, 2, ..., N
    cost_m = normal(μ_p, σ_p)   % normalförd.
    cost_e = el_pris( )
    tot_cost[i] = cost_m + cost_e
end_for
mean_cost = mean(tot_cost + cost_a)
```

6/ ODE:

$$\begin{cases} y_1' = k_1 \cdot y_1 - k_2 y_1^2 - k_3 y_1 \cdot y_2 \\ y_2' = k_3 y_1 \cdot y_2 - k_4 y_2 \end{cases}$$

a/ stokastisk modell:

$$\begin{cases} y_1 \xrightarrow{k_1 y_1} 2y_1 \\ y_1 \xrightarrow{k_2 y_1^2} \emptyset \\ y_1 + y_2 \xrightarrow{k_3 y_1 \cdot y_2} 2y_2 \\ y_2 \xrightarrow{k_4 y_2} \emptyset \end{cases}$$

b/ stochiometrimatris:

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}$$

propensiteter:

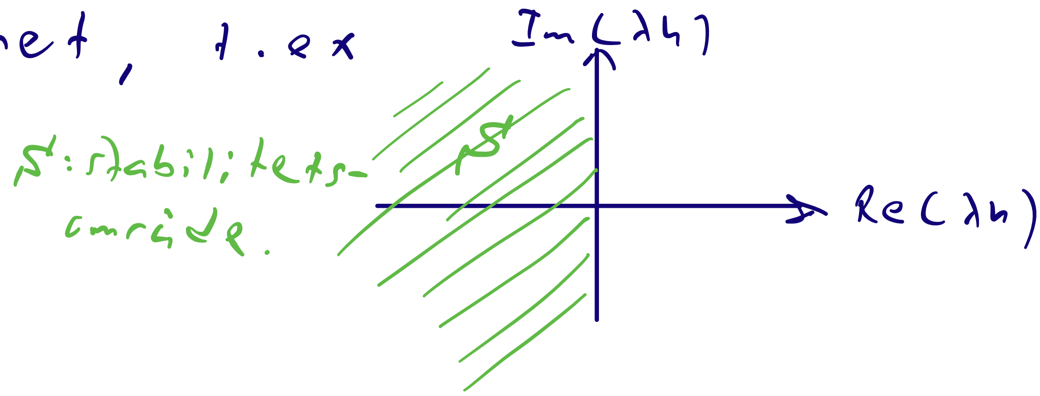
$$p = \begin{pmatrix} k_1 \cdot y_1 \\ k_2 \cdot y_1^2 \\ k_3 \cdot y_1 \cdot y_2 \\ k_4 \cdot y_2 \end{pmatrix}$$

7/ Stabilitetsområden visar för vilka val av steglängd h som metoden är numeriskt stabil (eller instabil). Området härleds genom att applicera en viss metod på testekvationen

$$y'(t) = \lambda \cdot y(t) \quad , \quad Re(\lambda) < 0 .$$

Om numerisk lösning växande för ett visst h, så är metoden instabil för de h-värdena. Man får då ett område i λh-planet.

Ovillkorligt stabil innebär att lösningen är avtagande för alla h (givet $Re(\lambda) < 0$), dvs i hela vänstra talplanet, t.ex



8/ Exakt lösning:

$$y(t_{i+1}) = y(t_i) + h\,y'(t_i) + \frac{h^2}{2} y''(t_i) + \frac{h^3}{6} y'''(t_i) + O(h^4)$$

Numerisk lösning:

$$y_{i+1} = y_i + h \cdot a\, f(t_i, y_i) + h b \cdot f(t_{i+1}, y_{i+1}) = y_i + h \cdot a\, y_i' + h b\, y_{i+1}'$$

$$= y_i + h a\, y_i' + h b\left(y_i' + h y_i'' + \frac{h^2}{2} \cdot y_i''' + O(h^4)\right) =$$

$$= y_i + h y_i'(a+b) + h^2 b y_i'' + \frac{h^3}{2}\cdot b\cdot y_i''' + O(h^4)$$

$y_i = y(t_i)$ (lokalt fel):

$$y(t_{i+1}) - y_{i+1} = \cancel{y_i} + h y_i' + \frac{h^2}{2} y_i'' + \frac{h^3}{6} y_i''' + O(h^4) -$$

$$- \cancel{y_i} - h y_i'(a+b) - h^2 \cdot b y_i'' - \frac{h^3}{2}\cdot b \cdot y_i''' + O(h^4) =$$

$$= h y_i' \underbrace{(a+b-1)}_{=0 \text{ då } a=\frac{1}{2},\ b=\frac{1}{2}} + h^2 y_i'' \underbrace{\left(\frac{1}{2} - b\right)}_{=0 \text{ då } b=\frac{1}{2}} + h^3 y_i''' \left(\frac{1}{6} - \frac{b}{2}\right) + O(h^4)$$

$a = b = \frac{1}{2}$ ger högsta möjliga n.o.:

$$y(t_{i+1}) - y_{i+1} = h^3 y_i''' \left(\frac{1}{6} - \frac{1/2}{2}\right) + O(h^4) = -\frac{1}{12} h^3 y_i''' + O(h^4)$$

9/

Ange vilken typ av metod (kolumn) är lämplig i vilken tillämpning (rad)

	Stokastisk metod	Deterministisk metod
Kemisk reaktion med få molekyler	✔	
Lösa en väderprognosmodell		✔
Matematisk ODE		✔
Optionsprissättning	✔	
Integral 2D		✔
Kemisk reaktion med många molekyler		✔
Integral 20D	✔	
Simulera trafikflöden i en stad	✔	

10/ ODE'n är en styv ODE och explicita metoder får (sannolikt) stabilitets-problem pga små stabilitetsområden. man bör därför välja en implicit metod, som har större stabilitets-områden.

Om man väljer en explicit metod kommer steglängden pressas ned pga instabilitet. Detta leder till väldigt lång beräkningstid.

Notera, detta gäller adaptiva metoder, men alla lösare i t.ex. Scipy är adaptiva.

11/ Testekv. $y' = \lambda y$

$$\begin{cases} k_1 = \lambda y_i \\ k_2 = \lambda(y_i + q\cdot h\cdot k_1) = \lambda(y_i + q\cdot h\cdot \lambda y_i) \\ y_{i+1} = y_i + h\cdot a_1\cdot \lambda y_i + h a_2\cdot\lambda(y_i + q\cdot h\cdot\lambda y_i) = \end{cases}$$

$$= y_i + h a_1 \lambda y_i + h a_2 \lambda y_i + h^2 a_2 \lambda^2 \cdot q \cdot y_i = \left(1 + h\lambda \underbrace{(a_1 + a_2)}_{=1} + \underbrace{a_2 q}_{=\frac{1}{2}}\cdot(h\lambda)^2\right)\cdot y_i$$

$$= \left(1 + h\lambda + \tfrac{1}{2}(h\lambda)^2\right)\cdot y_i \Rightarrow$$

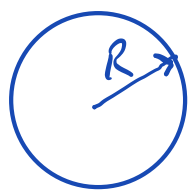
$$\Rightarrow \quad y_k = \left(1 + h\lambda + \tfrac{1}{2}(h\lambda)^2\right)^k \cdot y_0 = \left(1 + z + \tfrac{1}{2}z^2\right)^k \cdot y_0$$

stabilitet då $\left|\tfrac{1}{2}z^2 + z + 1\right| < 1$

$\therefore$ samma stabilitetsvillkor oberoende av val av a_1, a_2, p och q.

stabilitetsvillkoret är $\left|\tfrac{1}{2}z^2 + z + 1\right| < 1$

12/



$$x_i = r_i \cdot \cos(\theta_i)$$
$$y_i = r_i \cdot \sin(\theta_i) \quad , i = 1, \dots, n$$

$\theta_i \in U(-\pi, \pi)$, r_i ges ur radie(R,n)

$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ ges ur x, y = RandomPointsOnDisc(R,n)

a) Algoritm:

- skapa $\theta_i \in U(-\pi, \pi)$. Ges ur $U(0,1)$ genom $\theta = -\pi + 2\cdot\pi\cdot u$ där $u \in U(0,1)$)

  <u>I numpy</u>: theta=-np.pi*2*np.random.rand(n,1)

  eller

  theta=np.random.uniform(-π,π,n)

- skapa r_i: Använd radie(R,n)

  r=radie(R,n)

- Beräkna vektorer x och y. Använd elementvis multiplikation (np.multiply eller * i numpy).

  x=r*np.cos(theta)
  y=r*np.sin(theta)

  eller

  x=np.multiply(r,np.cos(theta))
  y=np.multiply(r,np.sin(theta))

"scriptet" här blir:

```
R= ...
n= ...
x,y=RandomPointsOnDisc(R,n)
plot(x,y) as dots
```

med matplotlib t.ex

```
plt.scatter(x,y)
```

b/ Hitta slumptal ur fördelningen (PDF):

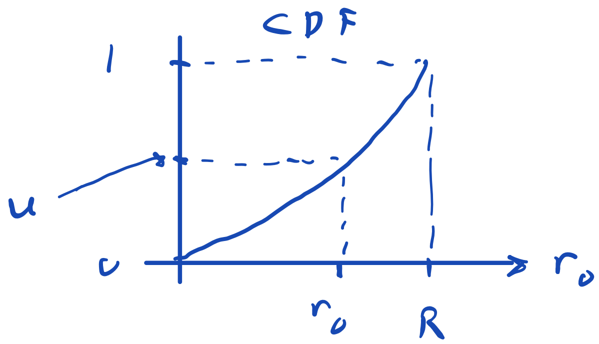
$$f(r) = \begin{cases} 0 & r < 0 \\ 2r/R^2 & 0 \le r \le R \\ 0 & r > R \end{cases}$$

Använd invers transform sampling för att gå från $U(0,1)$ till annan fördelning.

Använd CDF:

$$\int_{-\infty}^{r_0} \frac{2r}{R^2}\,dr = \frac{2}{R^2}\int_0^{r_0} r\,dr = \frac{\cancel{2}}{R^2}\left[\frac{r^2}{\cancel{2}}\right]_0^{r_0} = \frac{1}{R^2}\left(r_0^2 - 0\right)$$

CDF: $F(r_0) = \frac{1}{R^2}\cdot r_0^2$. Notera att $F(r_0)$ går från 0 till 1 för alla $r_0 \le R$



Vi kan nu välja $u \in U(0,1)$ och hitta motsvarande $r_0 \in f(r)$

$$\Rightarrow u = \frac{r_0^2}{R^2} \Rightarrow r_0^2 = u\cdot R^2 \Rightarrow r_0 = +\sqrt{u\cdot R^2} = R\cdot\sqrt{u}$$

I NumPy:

```
def radie(R, n)
    u = np.random.rand(n, 1)
    r = R * np.sqrt(u)
    return r
```

Förslag på kod i numpy (men krävs inte färdig och fungerande kod på tentan)

```python
import numpy as np
import matplotlib.pyplot as plt

def radie(R,n):
    u = np.random.rand(n,1)
    r = R*np.sqrt(u)
    return r

def randomPointsOnDisc(R,n):
    theta = -np.pi + 2*np.pi*np.random.rand(n,1)
    r = radie(R,n)
    x = np.multiply(r, np.cos(theta))
    y = np.multiply(r, np.sin(theta))
    return x, y

R = 5      # Radius
n = 10000 # Number of points
x, y = randomPointsOnDisc(R,n)
plt.scatter(x,y, s=1)
axisticks=np.arange(-R,R+1,step=1)
plt.xticks(axisticks)
plt.yticks(axisticks)
plt.grid()
plt.show()
```