Kunnskap for en bedre verden

Institutt for matematiske fag

Eksamensoppgave i **TMA4106 Matematikk 2 for MTELSYS**

Faglig kontakt under eksamen: Morten Andreas Nome

Tlf: 90849783

Eksamensdato:

Eksamenstid (fra–til): 09:00 - 13:30

Hjelpemiddelkode/Tillatte hjelpemidler: Alle hjelpemidler tillatt. Samarbeid er juks.

Annen informasjon:
Denne eksamenen består av 10 delpunkt som alle teller like mye. Alle svar skal begrunnes, og veien til svaret er viktigere enn svaret. Husk derfor å skrive alle steg i beregningene dine. Samarbeid er juks. Lykke til.

Målform/språk: bokmål

Antall sider: 8

Antall sider vedlegg: 0

Kontrollert av:

Informasjon om trykking av eksamensoppgave	
Originalen er:	
1-sidig ☐	**2-sidig** ☒
sort/hvit ☒	**farger** ☐
skal ha flervalgskjema ☐	

Dato Sign

Oppgave 1

a) Oppgaveteksten hinter om at matrisen har et ikketrivielt nullrom. Dette betyr at en av egenverdiene må være null, og følgelig kommer det karakteristiske polynomet til å være lett å faktorisere. Derfor går vi på med friskt mot, og beregner (må nok til med *abc*-formelen her)

$$\det\begin{pmatrix} 1-\lambda & 2 & 3 \\ 2 & 3-\lambda & 5 \\ 3 & 5 & 8-\lambda \end{pmatrix} = -\lambda^3+12\lambda^2+3\lambda = -\lambda\left(\lambda-6+\sqrt{39}\right)\left(\lambda-6-\sqrt{39}\right)$$

som gir egenverdier 0, og $6 \pm \sqrt{39}$. En egenvektor tilfredsstiller likningen

$$A\mathbf{x} = \lambda\mathbf{x}$$

og dersom $\lambda = 0$, blir denne likningen

$$A\mathbf{x} = \mathbf{0}.$$

Derfor er nullrommet til en matrise det samme som egenrommet til egenverdien 0.

b) Løsningen til et slikt 3×3-system er

$$\mathbf{y}(t) = c_1 e^{\lambda_1 t}\mathbf{x}_1 + c_2 e^{\lambda_2 t}\mathbf{x}_2 + c_3 e^{\lambda_3 t}\mathbf{x}_3$$

Siden egenverdiene til matrisen i denne oppgaven er forskjellige, vet vi at egenvektorene er lineært uavhengige, og siden vektoren

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

både er initialkrav og egenvektoren til $\lambda = 0$, må den korrekte løsningen være

$$\mathbf{y}(t) = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}.$$

Oppgave 2

a) Denne tar vi med normallikningene. Vi beregner

$$\begin{pmatrix} 1 & 4 & 9 & 16 & 25 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 979 & 225 & 55 \\ 225 & 55 & 15 \\ 55 & 15 & 5 \end{pmatrix}$$

og

$$\begin{pmatrix} 1 & 4 & 9 & 16 & 25 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 155 \\ 45 \\ 15 \end{pmatrix}$$

Likningssystemet

$$\begin{pmatrix} 979 & 225 & 55 \\ 225 & 55 & 15 \\ 55 & 15 & 5 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 155 \\ 45 \\ 15 \end{pmatrix}$$

gir $a = -5/7$, $b = 30/7$, $c = -2$, slik at projeksjonen blir

$$\begin{pmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \end{pmatrix} \begin{pmatrix} -5/7 \\ 30/7 \\ -2 \end{pmatrix} = \frac{1}{7} \begin{pmatrix} 11 \\ 26 \\ 31 \\ 26 \\ 11 \end{pmatrix}$$

Her er pythonkode for å beregne alt:

```
import numpy as np

x=[1,2,3,4,5]
y=[2,3,4,5,1]

A=[[1,1,1],[4,2,1],[9,3,1],[16,4,1],[25,5,1]]
B=np.dot(np.transpose(A),A)
z=np.dot(np.transpose(A),y)
c=np.linalg.solve(B,z)
```

b) Her kan vi bygge videre på forrige oppgave. Et kvadratisk regresjonspolynom er et polynom på formen $ax^2 + bx + c$, og likningene blir

$$\begin{aligned} a + b + c &= 2 \\ 4a + 2b + c &= 3 \\ 9a + 3b + c &= 4 \\ 16a + 4b + c &= 5 \\ 25a + 5b + c &= 1 \end{aligned}$$

Nå ser vi at a, b og c fra forrige oppgave er koeffisientene i det kvadratiske regresjonspolynomet vi er ute etter. Følgende kode produserer korrekt plot:

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

x=[1,2,3,4,5]
y=[2,3,4,5,1]

A=[[1,1,1],[4,2,1],[9,3,1],[16,4,1],[25,5,1]]
B=np.dot(np.transpose(A),A)
z=np.dot(np.transpose(A),y)
c=np.linalg.solve(B,z)

t=np.linspace(1,5,100)
p=c[0]*t**2+c[1]*t+c[2]

plt.figure()
plt.scatter(x,y)
plt.plot(t,p)
plt.savefig('reg.png')
```

Oppgave 3

a) Dersom f er reell, kan vi beregne

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-int}\,dt = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)\overline{e^{int}}\,dt = \overline{\frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{int}\,dt} = \overline{c_{-n}}.$$

b) Denne kan vi raskt ta med en kjent og kjær fourierrekke vi har beregnet tidligere. Heavisidefunksjonen

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

sin fourierrekke er

$$u(t) \sim \frac{1}{2} + \frac{2}{\pi}\sum_{n=1}^{\infty} \frac{1}{2n-1}\sin(2n-1)t.$$

Men vår funksjon kan skrives $f(t) = 2u(t) - 1$, så fourierrekken til f blir

$$f(t) \sim \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{1}{2n-1} \sin(2n-1)t.$$

Det er selvfølgelig også mulig å regne ut dette fra scratch. Følgende kode produserer korrekt plot:

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

#fourierrekken skal ha K ledd
K=10

#vi plotter med finhet paa N punkt
N=1000

#t-aksen
x=np.linspace(-2*np.pi,2*np.pi,num=N)

#heavisidefunksjonen
f=-np.ones(N)
for i in range(N):
    if x[i] > 0:
        f[i]=1.0

#trunkert fourierrekke
z=0
for i in range(K):
    z=z+4.0/(np.pi*(2*i+1))*np.sin((2*i+1)*x)

plt.figure()

#plotte heaviside og trunkert fourierrekke
plt.plot(x,f)
plt.plot(x,z)

#korrekt utsnitt
plt.axis([-np.pi,np.pi,-2,2])

#vise plot
plt.savefig('nesten_heaviside')
```

Oppgave 4

a) Hvis du har et system, sier superposisjonsprinsippet at en lineærkombinasjon av to løsninger også må være en løsning. I et vektorrom V må vi ha at

$$a\mathbf{x} + b\mathbf{y} \in V \quad \text{dersom} \quad \mathbf{x},\ \mathbf{y} \in V$$

og dette er nettopp superposisjonprinsippet. Systemet i oppgaven har ikke denne egenskapen, for summen av to løsninger er ikke nødvendigvis en løsning. Anta at vi har to løsninger (y_1, z_1) og (y_2, z_2), og sett summen $(y_1 + y_2, z_1 + z_2)$ av dem inn i likningen $\dot{z} = zy - z$. Venstre- og høyresiden er ikke like:

$$\frac{d}{dt}(z_1 + z_2) = \dot{z}_1 + \dot{z}_2 = z_1 y_1 - z_1 + z_2 y_2 - z_2 \neq (z_1 + z_2)(y_1 + y_2) - (z_1 + z_2),$$

så med andre ord er ikke $(y_1 + y_2, z_1 + z_2)$ en løsning.

b) Symplektisk Euler blir

$$\begin{aligned} y_{n+1} &= y_n + h\left(y_n\left(\frac{y_n - 0.2}{y_n + 0.2}\right)\left(1 - \frac{y_n}{4}\right) - y_n z_n\right) \\ z_{n+1} &= z_n + h\left(z_n y_{n+1} - z_n\right) \end{aligned}$$

Følgende kode produserer korrekt plot:

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


plt.figure()
#plt.xlim([0,9])
#plt.ylim([0,12])

T=400.0
N=40000
h=T/N

t=np.zeros(N+1)
y=np.zeros(N+1)
z=np.zeros(N+1)


y[0]=1
z[0]=1

K=4.0
epsilon=.2

for i in range(N):
    y[i+1]=y[i]-h*(y[i]*(y[i]-epsilon)/(y[i]+epsilon)*(1-y[i]/K)-y[i]*z[i])
    z[i+1]=z[i]-h*z[i]*(y[i+1]-1)
    t[i+1]=t[i]+h

plt.plot(t,y)
plt.plot(t,z)
plt.savefig('lotkavolterra_aug.png')
```

Denne modellen spår at populasjonene kommer til å stabilisere seg i bestemte proporsjoner. For akkurat disse verdiene av K og ϵ blir det dobbelt så mange byttedyr (y) som rovdyr (z).

Oppgave 5 Vi begynner med å skrive

$$i = e^{\pi i/2+2k\pi} \quad k \in \mathbb{K},$$

slik at

$$\sqrt{i} = \sqrt{e^{\pi i/2+2k\pi}} = \left(e^{\pi i/2+2k\pi}\right)^{1/2} = e^{\pi i/4+k\pi}$$

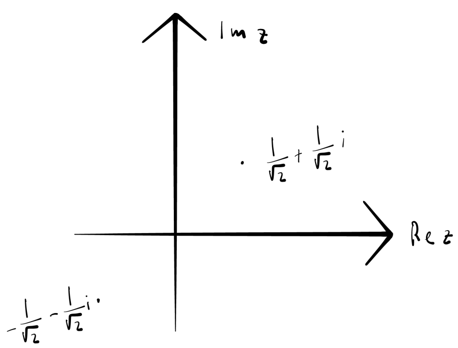
Vi vet av erfaring at to etterfølgende verdier av k gir alle røttene som er å finne, så vi kan like gjerne velge $k = 0$ og $k = 1$, som gir

$$e^{\pi i/4} = \cos\frac{\pi}{4} + \sin\frac{\pi}{4} = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$$

og

$$e^{5\pi i/4} = \cos\frac{5\pi}{4} + \sin\frac{5\pi}{4} = -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i$$

Her er de:



Oppgave 6 Vi begynner med å finne den homogene løsningen. Likningen

$$\ddot{y} + y = 0$$

har vi løst så mange ganger nå, så jeg regner med alle vet at løsningen er

$$y(t) = c_1 \cos t + c_2 \sin t.$$

Så til den inhomogene. Siden høyresiden er $\sin t$, tenker man nok at partikulærløsningen også bør være på formen $A \sin t$, men dette går ikke så bra, for $A \sin t$ er jo en homogen løsning, og vi vil kun få $0 = 0$ om vi prøver denne strategien.

Hvis vi slår opp på s. 5 i Magnus' notat (etter eksempel 18), ser vi at vi bør gjette på at

$$y_p(t) = t(c_3 \cos t + c_4 \sin t).$$

Vår voldsomme erfaring og intuisjon forteller oss at sinusen er unødvendig, og den enkleste måten å vise at vi har rett på, er å prøve det ut. Vi setter derfor $c_4 = 0$, deriverer to ganger og får

$$\ddot{y}_p(t) = -2c_3 \sin t - c_3 t \cos t$$

slik at

$$\ddot{y}_p(t) + y_p(t) = c_3 \left(-2 \sin t - t \cos t + t \cos t\right) = -2c_3 \sin t.$$

Hvis vi sammenlikner dette med påtrykket i likningen, ser vi at $c_3 = -1/2$, slik at løsningen blir

$$\ddot{y}(t) + y(t) = \sin t \qquad y(0) = \dot{y}(0) = 0$$

med penn og papir.

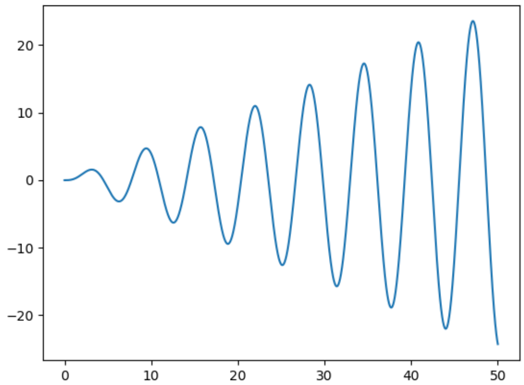
$$y(t) = y_h(t) + y_p(t) = c_1 \cos t + c_2 \sin t - \frac{1}{2} t \cos t$$

Kravet $y(0) = 0$ gir at $c_1 = 0$, og kravet $\dot{y}(0) = 0$ gir $c_2 = 1/2$, så løsningen blir

$$y(t) = y_h(t) + y_p(t) = \frac{1}{2}\left(\sin t - t \cos t\right).$$

Her er et plot av løsningen:

Dette er et klassisk eksempel på resonans. Perioden til de homogene løsningene er 2π, og påtrykket har nøyaktig den samme perioden. Resultatet blir en svingning som stadig forsterkes. Når dette skjer i et PA-anlegg, kalles det feedback, og hvis man bygger en bro med uheldige resonansfrekvenser, kan følgene bli katastrofale: `https://www.youtube.com/watch?v=j-zczJXSxnw`.



20
10
0
−10
−20
0
10
20
30
40
50