MAT-INF 1100 Obligatorisk oppgave 2

Kenneth Ramos Eikrehagen

9. november 2016

Innhold

1 Oppgave 1.

a)
Her brukte jeg at akselerasjon er den deriverte av farten. Det betyr at akselerasjonen er $\frac{fartsendring}{tidsendring}$ altså $a = \frac{\delta x}{\delta t}$ Jeg lagde to arrayer av fart- og tidsverdiene slik at jeg kunne indeksere inn endring i fart over endring av tid og lagret dette i en array for akselerasjon som jeg senere skulle plotte.
b)
Her brukte jeg numerisk integrasjon, nærmere bestemt trapes metoden. Fordi jeg vet at integralet av fart = posisjon, og trapes metoden er en god tilnærming til den virkelige grafen.
c)
Her skulle vi plotte a) og b), jeg laget derfor en klasse der jeg kunne lett hente inn de funksjonene jeg ville bruke og det ble enklere for meg å plotte resultatet. Jeg syntes også koden ser penere ut på denne måten. Jeg plottet også begge grafene i et vindu.
Jeg har lagt ved python koden min:

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

class pos_aks:

    def __init__(self, t, v):
        self.t, self.v = t, v
#a)
    def aks(self):
        t = np.array(self.t); v = np.array(self.v)
        a = np.zeros(len(t))
        for i in xrange(1,len(t)):
            a[i] = (v[i]-v[i-1])/(t[i]-t[i-1])
        return a
```

```
#b)
    def pos(self):
        t = np.array(self.t); v = np.array(self.v)
        s = np.zeros(len(v))

        for i in xrange(1,len(t)):
            s[i] = s[i-1] + (((v[i-1])+(v[i]))/2.0)*(t[i]-t[i-1])

        return s


t = []; v = []

infile = open( running.txt , r )
for line in infile:
    tnext, vnext = line.strip().split( , )
    t.append(float(tnext)), v.append(float(vnext))
infile.close()

#c)
ap = pos_aks(t,v)
a = ap.aks()
p = ap.pos()

plt.subplot(2,1,1)
plt.plot(t,a)
plt.title( Akselerasjon graf )
plt.ylabel( a(t) )
plt.legend([ akselerasjon ])
plt.subplot(2,1,2)
plt.plot(t,p, black )
plt.title( Posisjon graf )
plt.axis([0,7000,0,2.8e4])
plt.xlabel( t )
plt.ylabel( s(t) )
plt.legend([ posisjon ])
plt.show()
```

Den øverste grafen i figur 1 på neste side viser hvordan akselerasjonen er som funksjon av tiden t. Den nederste grafen i figur 1 på neste side viser hvordan posisjonen endrer seg med tiden.

2 Oppgave 2.

a)

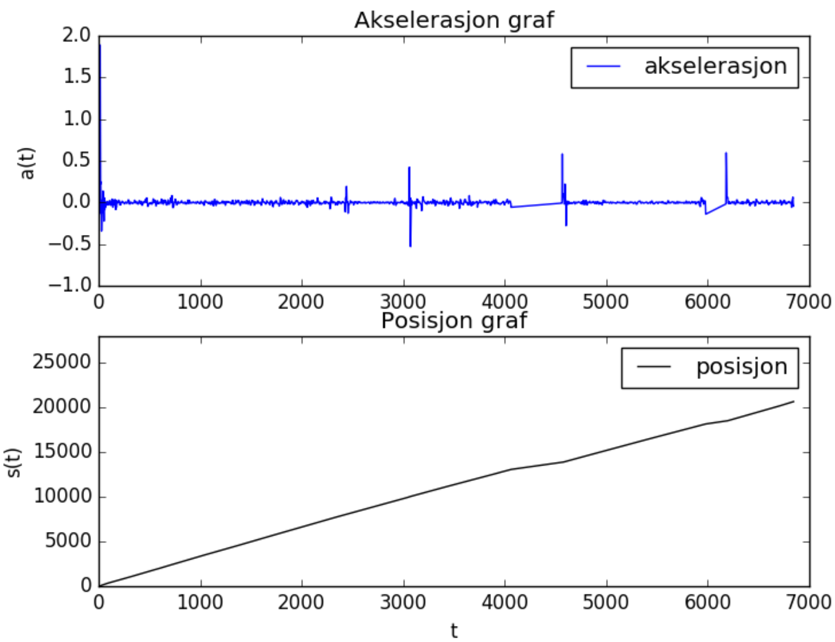
$$x' + x^2 = 1,\ x(0) = 0 \tag{1}$$

Dette kan også skrives som

$$\frac{d}{dt}x(t) = 1 - x(t)^2$$

Deler vi med $1 - x(t)^2$ og tar integralet på begge sider får vi en ligning som ser slik ut:

$$\int \frac{\frac{d}{dt}x(t)}{1 - x(t)^2}dt = \int 1dt$$



Figur 1: Jeg har plottet begge grafene i samme plott

Jeg ser at venste siden blir $t + c_1$, jeg løser så høyre siden og slår sammen:

$$\int \frac{\frac{d}{dt}x(t)}{1 - x(t)^2} dt = \int \frac{dx(t)}{1 - x(t)^2} = artanh(x(t)) + c_2 \tag{2}$$

$$\begin{aligned} artanh(x(t)) + c_2 &= t + c_1 \\ artanh(x(t)) &= t + c_1 - c_2 \\ x(t) &= tanh(t + c) \end{aligned}$$

Putter inn $x(0) = 0$ får at $tanh(c) = 0 \Rightarrow c = 0$, $x(t) = tanh(t)$ Vet at $tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$ kan nå skrive $x(t)$ på en annen måte.

$$x(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$$

Her har jeg brukt at $\int \frac{1}{1-x^2} dx = artanh(x) + c$ som det står om i kap 7.7 side 387 i kalkulus.

b)

Jeg har brukt følgende algoritme til å løse denne oppgaven.

$$n = 5,\ h = \frac{2}{5},\ k = \{0, 1, 2, 3, 4, 5\} \tag{3}$$

$$x_{k+1} = x_k + h \cdot f(t_k, x_k)$$

$$t_{k+1} = a + (k + 1) \cdot h$$

Legger ved python koden:

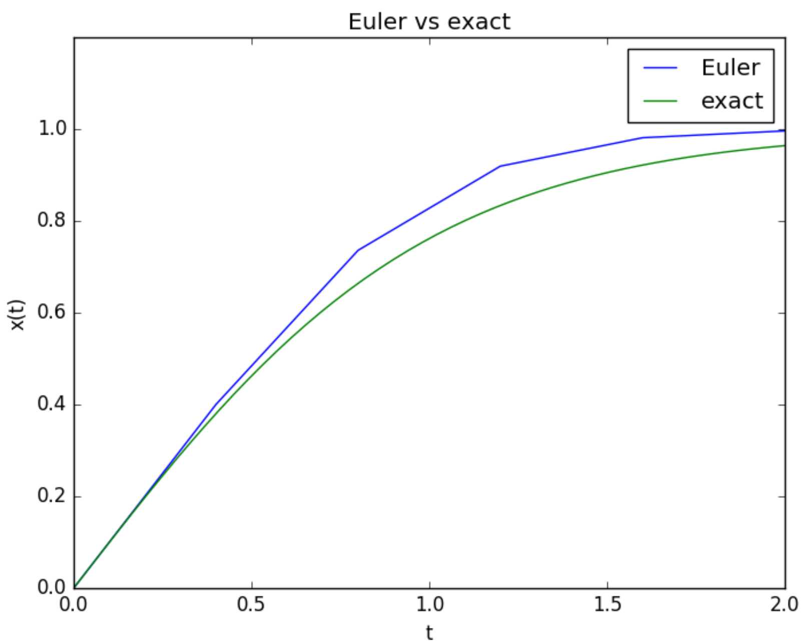
```python
import numpy as np

k = [0,1,2,3,4,5]
n = 5
x = np.zeros(len(k))
t = np.zeros(len(k))

for i in range(1,n+1):
    h = 0.4
    x[i] = x[i-1] + h*(1-x[i-1]**2)
    t[i] = (i)*h
```

På figuren 2 på neste side ser vi hvorden Euler metoden bruker tangenten i x_i for å finne en tilnærming til den eksakte grafen.

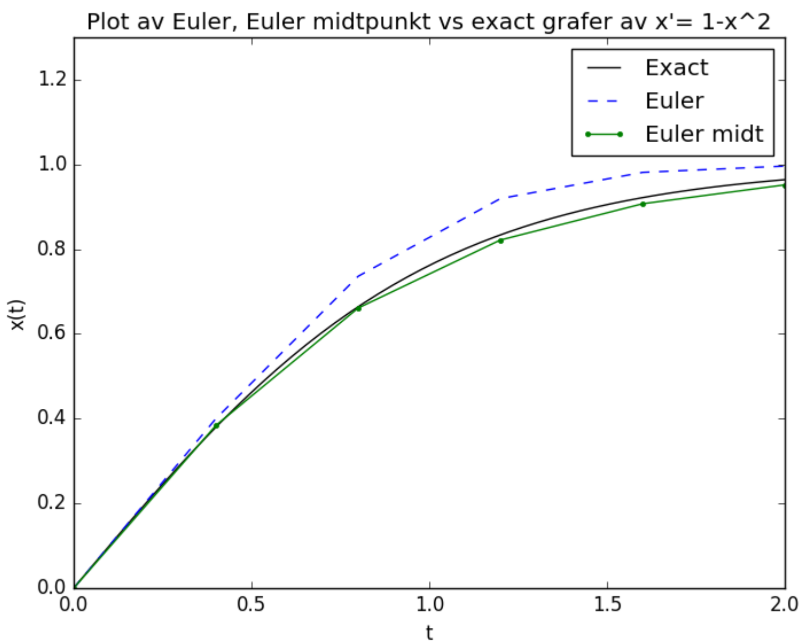
c)

Jeg modifiserte litt på algoritmen jeg brukte i oppgave b).

$$n = 5,\ h = \frac{2}{5},\ k = \{0, 1, 2, 3, 4, 5\} \tag{4}$$



Figur 2: I dette plottet ser vi Euler metoden mot den eksakte løsningen



Figur 3: I dette plottet ser vi Euler metoden og Eulers midtpunkt metode mot den eksakte løsningen

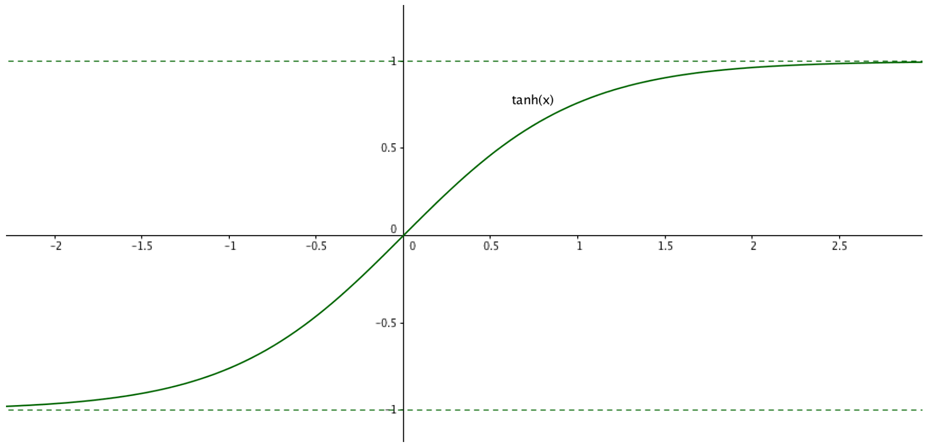
$$x_{k+\frac{1}{2}} = x_k + \frac{h}{2} \cdot f(t_k, x_k)$$
$$x_{k+1} = x_k + h \cdot f(t_K, x_{k+\frac{1}{2}})$$
$$t_{k+1} = a + (k+1) \cdot h$$

Legger ved python koden:

```
import numpy as np

k = [0,1,2,3,4,5]
n = 5
x = np.zeros(len(k))
t = np.zeros(len(k))

for i in range(1,n+1):
    h = 0.4
    xtmp = x[i-1] + (h/2)*(1-x[i-1]**2)
    x[i] = x[i-1] + h*(1-xtmp**2)
    t[i] = (i)*h
```

d)

Jeg ser ut ifra grafene jeg har fått når jeg har plottet at for $0 \leq x(t^*) < 1$ er voksende uansett hvilken verdi t^* har, innenfor de begrensiningene som er satt.



Figur 4: Grafen til den hyperbolske funksjonen tanh(x)

Hvis vi bruker grafen til tanh(t), som vi ser på figur 4, at den går mot 1 for $t^* \in [1, \infty)$