1 Innlevering 4 - Numerisk integrasjon

Standardnormalfordelingen er gitt ved

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Standardnormalfordelingen er et eksempel på en sannsynlighetstetthetsfunksjon, og for TMA4400 holder det å vite at sannsynligheten for å havne mellom a og b er gitt ved

$$\frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2}\, dx.$$

Det er bare ett problem - standardnormalfordelingen kan ikke antideriveres. Under er kode for å beregne arealet ved riemannsummer.

[ ]:
```
import numpy as np

#integrasjonsgrenser
a=0
b=1

#gitter
n=1000
x=np.linspace(a,b,n+1)
h=b/n

#venstre riemannsum
f=h*(np.sum(np.exp(-x**2/2))-np.exp(-b**2/2))/np.sqrt(2*np.pi)

print(f)
```

1. Modifiser koden til å beregne arealet ved trapesmetoden, kjør alt med $a = 0$, $b = 1$, sammenlikne begge metoder med den korrekte sannsynligheten

   $$\frac{1}{\sqrt{2\pi}} \int_0^1 e^{-x^2/2}\, dx \approx 0.3413447460685429485852.$$

   Kan du forklare forskjellen? (Fun fact: Det er denne beregningen som forteller deg at om du har IQ på 115 er du smartere enn om lag 84 prosent av befolkningen. Mer om dette i TMA4240/4245.)

2. Man kan bruke polynominterpolasjon til å finne bedre teknikker for numerisk integrasjon. Utled Simpsons metode ved å finne lagrangepolynomene til gitteret

   $$\left\{a, \frac{a+b}{2}, b\right\}$$

   og integrere dem. Test Simpsons metode på arealet over og forklar forskjellen.

De numeriske integrasjonsrutinene gir oss et sett med korresponderende teknikker for å løse ordinære differensiallikninger numerisk. La oss si vi ønsker å løse initialverdiproblemet

$$\dot{x} = f(x) \qquad x(0) = 0$$

på intervallet $[0, T]$. Vi deler tidsintervallet inn i en partisjon med $n + 1$ punkter

$$t_k = kh \qquad 0 \leq k \leq n$$

der $h = T/n$, og integrerer differensiallikningen fra ett gitterpunkt til neste:

$$x\left(t_{k+1}\right) - x\left(t_k\right) = \int_{t_k}^{t_{k+1}} f\left(x\left(t\right)\right)\ dt$$

Hvis vi setter inn en enkelt rektangulær riemannboks (venstre, høyre eller sentrert) eller trapesmetoden for integralet på høyresiden, og så setter inn approksimasjonen $x_k \approx x\left(t_k\right)$ overalt, får vi nå fire metoder på rappen:

$$\begin{aligned}
\text{Fremlengs Euler:} \qquad & x_{k+1} = x_k + hf\left(x_k\right) \\
\text{Baklengs Euler:} \qquad & x_{k+1} = x_k + hf\left(x_{k+1}\right) \\
\text{Midtpunktmetoden:} \qquad & x_{k+1} = x_k + hf\left(\frac{x_k + x_{k+1}}{2}\right) \\
\text{Trapesmetoden:} \qquad & x_{k+1} = x_k + \frac{h}{2}\left(f\left(x_k\right) + f\left(x_{k+1}\right)\right)
\end{aligned}$$

Under finner du kode for fremlengs Euler på det logistiske problemet

$$\dot{x} = a\left(1 - x/k\right)x \qquad x(0) = x_0 \qquad a > 0,\ k > 0.$$

[ ]:
```
import matplotlib.pyplot as plt

#parametre
a=1 #vekstfaktor
k=1 #bærekapasitet

#tidsakse
T=10
N=10000
h=T/N
t=np.linspace(0,T,N+1)

#løsning
x=np.zeros(N+1)
x[0]=.1
for i in range(N):
    x[i+1]=x[i]+h*x[i]*(1-x[i])
```

```
plt.plot(t,x)
plt.savefig('logistisk.png')
```

3. Implementér de forskjellige metodene for det logistiske problemet og sammenlikne med den analytiske løsningen

$$x(t) = \frac{kx_0e^{at}}{k - x_0 + x_0e^{at}}.$$

Kan du forklare forskjellene? (Velg vekstfaktoren a og bærekapasiteten k selv. For midtpunkt- og trapesmetoden kan en numerisk likningsløser komme til nytte.)