Av: Ellen E & Geir Magne Fløe

Opplegg 13 – **Modellering med luftmotstand**

Utleiing av uttrykk

Dersom du slepp eit objekt, t.d. ein ball, vil det bli påverka av tyngdekrafta G og luftmotstanden L. I eit slikt høve vil vi få følgande uttrykk for akselerasjonen (positiv retning er nedover):

$\Sigma F = G - L$ Newtons 2. lov

$G - L = ma$

$mg - kv^2 = ma$ Set $L=kv$^2

$a = g - kv^2 / m$ Deler på m

Programmering

Vi startar med å definere grunnleggande storleikar og initialbetingelsar. Deretter deler vi tida opp i små tidsintervall, opprettar matriser og set initialbetingelsane inn på første plass i matrisene. Til sist bruker vi ei for-lykkje med Eulers metode og plottar grafane.

Puslespel

Set saman kodebitane til eit ferdig program.

```
v[0] = v0
s[0] = s0
t[0] = t0
```
→ Set initialbetingelsar inn i matrisene

```
from pylab import *

g = 9.81
m = 1.0
k = 0.1
```
→ Definerer grunnleggande variablar.

```
plot(t,v)
xlabel("Tid (s)")
ylabel("Fart (m/s)")
grid()
title("Fartsgraf")
figure()

plot(t,a)
xlabel("Tid (s)")
ylabel("Akselerasjon (m/s)")
grid()
title("Akselerasjonsgraf")
```
→ Plottar grafene.

```
a = zeros(N)
v = zeros(N)
t = zeros(N)
s = zeros(N)
```
→ Opprettar matriser

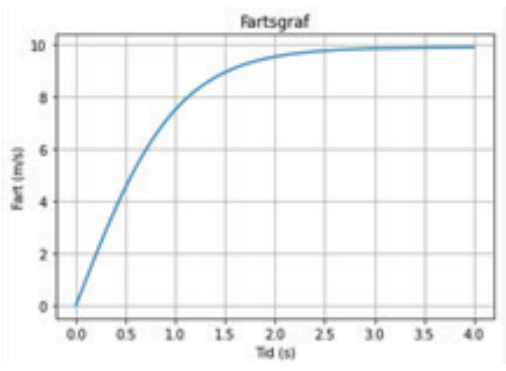
```
N = 1000
tid = 4
dt = tid/N
```
→ Deler tida opp i små tidsintervaller.

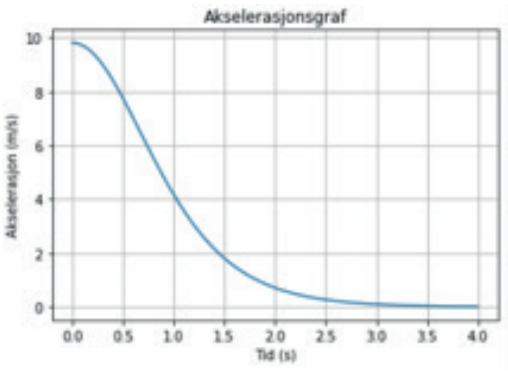
```
v0 = 0
s0 = 0
t0 = 0
```
→ Setter initialbetingelsar.

```
for i in range(N-1):
    a[i]=g-k/m*v[i]**2
    s[i+1]=s[i]+v[i]*dt+0.5*a[i]*dt**2
    v[i+1]=v[i]+a[i]*dt
    t[i+1]=t[i]+dt

a[N-1]=g-k/m*v[i]**2
```

For-lykkje med Eulers metode. Her bereknast først akselerasjonen som har indeks i. Deretter bereknast dei øvrige storleikane som ligg eit tidssteg framfor, og har indeksen i+1. Uttrykka her er bevegelsesformlane frå fysikken og er baserte på verdiane til s, v, og t for indeksen i. I aller siste line fyller vi ut siste plass i akselerasjons-matrisen.



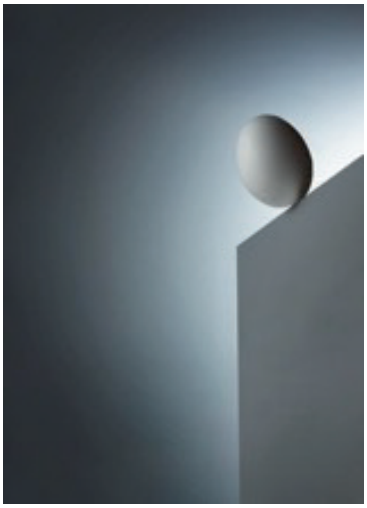


Redd det fallande egget!

- mål tid og akselerasjon med micro:bit og lagre til fil

Oppgåve

Lag noko som forhindrer egget frå å bli knust når det sleppest frå ca. 3 meters høgde. Mål tida og akselerasjonen medan egget fell med å feste ein micro:bit som lagrer data til ei fil, på egget. Dette datasettet med tid og akselerasjon skal de bruke for å modellere egget sin fart med programmering.



Fase 1: Korleis redde egget? Søk gjerne på «egg drop challenge» for tips.

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikn gjerne skisser frå ulike vinklar.
2. Forklar ideen din for dei andre på gruppa. Bruk gjerne skissene i forklaringa.
3. Heile gruppa diskuterer dei ulike ideane, og lagar ein felles plan for bygging.

Fase 3: Gjennomfør planen dykkar for å lage anti-eggknusaren og lag programmet for å måle akselerasjonen og lagre på micro:biten. Sjå tips på side 3.

Gjennomfør fase 4 – 7.

Programmeringsoppgåve med målte verdiar

De skal bruke dei målte verdiane for tida, og den tilhøyrande akselerasjonen for å finne egget sin fart medan det fell.

1. Korleis skal de gjere om akselerasjonsmålingane til rett eining, m/s^2 ?
2. Modeller egget sin fart og lag ein fartsgraf – sjå tips på side 4.
3. Kva for ein momentanfart skulle egget hatt ved bakken ifølge denne modellen?
4. Berekn kva for ein fart egget skulle hatt ved bakken ifølge energibevaring i tyngdefelt.
    a. Kva for nokre antakingar ligg til grunn for formelen de brukte?
    b. Kvifor blir denne verdien så ulik den programmet bereknar?

Programmeringsoppgåver – teoretisk modellering

1. Lag eit program som modellerer akselerasjonen til egget når det fell, og plottar resultatet som ein graf – sjå tips på førre sida.
2. Inkluder dei målte verdiane for akselerasjonen i same graf og diskuter kvifor dei er ulike.
3. Utvid programmet til å modellere farten til egget, og plott resultatet som ein fartsgraf.
4. Inkluder fartsverdiane de berekna i programmeringsoppgåva der de brukte dei målte akselerasjonsverdiane, i same grafen.
    a. På kva for ein måte er grafane ulike?
    b. Kva kunne de ha gjort eksperimentelt for at fartsgrafane skulle blitt likare kvarandre? Er det mogleg?

Skrive til fil på micro:bit med Python Mu

I slutten av kapittel 9 brukte vi datasett i form av tekstfiler for å lage matematiske modellar. Hadde det ikkje vore kult om vi kunne fått micro:biten til å lage slike filer av målingane sine? Da kan vi bruke micro:biten til å samle inn data over ein viss periode, utan å vere kopla til ein datamaskin. Datasettet blir lagra i micro:biten som ei .txt-fil, og vi kan legge henne over i datamaskinen, før vi plottar resultata eller lager ein matematisk modell med Python.

Det aller første vi må gjere, er å opprette ei tekstfil, det vil seie ei fil som sluttar på .txt, og det er enklast å gjere i Mu. Lagre fila i same mappe som resten av Python Mu-filene dine.

Deretter må den fila du har laga, kopierast over på micro:biten. Dette gjer du på same måten som for sonaren.

```
with open ('filnavn.txt', 'w') as fil:
```

Opnar fila som heiter filnavn.txt og tildeler den til eit objekt som vi kallar fil. Sidan vi har 'w' (write) i kommandoen gjerest det klar til å skrive inn i fila.

```
with open ('filnavn.txt', 'r') as fil:
```

Tilsvarande som lina over, men sidan vi har 'r' (read) i kommandoen gjerest det klar til å lese fila. Det er valfritt å ta med 'r'.

```
fil.write(y+'\n')
```

Skriver variabelen y til fila vår. +'n' gjør at det blir et linjeskift etter hvert tall, og det må vi ha for at python skal kunne lese filene.
Må stå med innrykk.

```
y = str(x)
```

Gjer om variabelen x til ein streng (tekstverdi).
Må stå med innrykk.

```
print(z)
```

Skriv variabelen z til skjermen på PC. Må IKKJE stå med innrykk.

```
z = fil.read()
```

Les det som står i fila vår, og legg det inn i variabel z. Må stå med innrykk.

Da gjenstår det å lage sjølve programmet. Under er eit døme der micro:biten måler temperaturen og legg målinga i ein .txt-fil med namn datasett.txt. Det kan vere greit å kunne sjekke korleis datasett.txt-fila ser ut utan å overføre til datamaskinen og opne den, difor er dette og med i puslespel-oppgåva.

Puslespill-oppgave

```
print(innhold)
```

```
t = temperature()
```

```
while not button_a.is_pressed():
```

```
sleep(1000)
```

```
fil.write(str(t)+'\n')
```

```
with open('datasett.txt') as fil:
```

```
with open('datasett.txt', 'w') as fil:
```

```
innhold = fil.read()
```

```
from microbit import *
```

Ekstraoppgåve

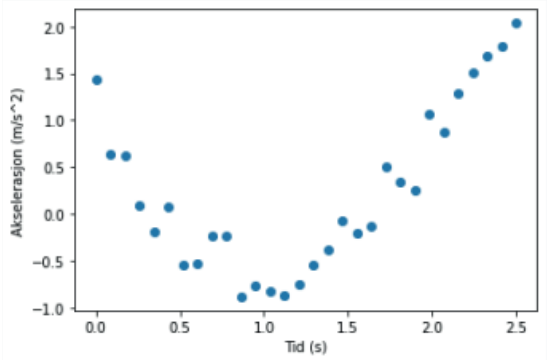
Lag eit program som bruker ein sensor du koplar på micro:biten for å samle inn data som du lagrar i ei fil på micro:biten. Kva må du gjere annleis?

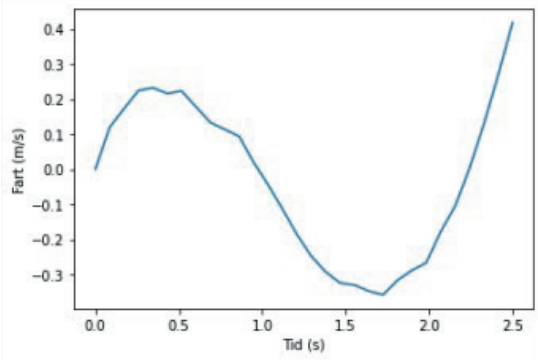
Kva må du forandre i programmet for å lage ei fil med to kolonnar der den første kolonnen er kor lang tid det har gått før målinga er gjort?

Modellering av fart med målt akselerasjon

Dersom vi har eit datasett med målte verdiar for akselerasjonen, kan vi bruke desse verdiane for å rekne ut farten i kvart av punkta. Vi vil da få grafar som vist i figurane nedanfor. Figuren til venstre viser dei målte datapunkta, medan figuren til høgre viser farten. Denne har vi kalkulert numerisk.

For å få til dette, må ein importere eit datasett med akselerasjonsverdiar (a) og tilhøyrande tider (t) i to lister. Deretter bruker vi ei lykkje med rørsleformelen □_(□+1)=□_□+□□ for å berekne fartane.





Med å bruke kodebitane under, kan ein lage eit program som plottar akselerasjonen og farten.

```
figure()
plot(t,v)
xlabel("Tid (s)")
ylabel("Fart (m/s)")
```

Plottar fartsgrafen. Funksjonen «figure» blir brukt når ein alt har plotta ein graf og ønsker å opne eit nytt grafvindu.

Vi deler opp den totale tida i små tidssteg og kallar iet slikt tidssteg «dt».

```
dt = t[N-1] / N
```

```
scatter(t,a)
xlabel("Tid (s)")
ylabel("Akselerasjon (m/s^2)")
```

```
N = len(t)
```

Måler lengda av lista «t», dvs tel talet på observerte data.

Plotter akselerasjonen. «Scatter» blir bruke når ein vil ha punkt, ikkje samanhengande graf.

```
v = zeros(N)
```

Oppretter liste for fartar.

```
for i in range(N-1):
    v[i+1] = v[i] + a[i]*dt
```

Beregner farter ved hjelp av ei løkke.

Diskusjonsoppgåver

1. Samanlikne figurane. Kva hender med grafen til farten når akselerasjonen er positiv? Og negativ?
2. Kva er akselerasjonen i topp- og botnpunktet til farten? Kvifor?