(Se romstofftidforkurs.cdu.no)

15.01 c) Fart er definert som den deriverte til posisjonsfunksjonen, altså stigningstallet til posisjonsgrafen.

d) ca. 36 $\frac{m}{s}$

g) 36,2 $\frac{m}{s}$

h) Positiv retning for fallet er oppover i stedet for nedover.

i) De beregnede fartsverdiene plottes nå midt mellom de to tidspunktene som de er beregnet fra.

k) Det gir et mer korrekt bilde av fartsendringen.

15.02 a) Bruk programmet ballposisjon.py fra 15.01 som modell.

b) Bruk programmet ballfart1.py fra 15.01 som modell.

a)

```
# Program hundremetern.py
import numpy as np
import matplotlib.pyplot as plt
```

← info, navn på program (kommentarlinje)

(programmets kode)

kode som importerer pakken kalt numpy (numerical python) inn i Python og gjør den tilgjengelig gjennom navnet np.

importerer plottepakken og døper den plt

```
# Leser t- og s-verdier inn i lister
```

```python
data = np.loadtxt('hundremeter.txt')  # laster inn tekstfil med tid og posisjon under løpet.
t = data[:,0]  # innleste t-verdier [s] i sekunder
s = data[:,1]  # innleste s-verdier [m] i meter

# Tegner posisjonsgrafen
plt.figure(1)
plt.plot(t,s)
plt.title('Posisjon')
plt.xlabel('$t$ / s')
plt.ylabel('$s$ / m')
```

15.03 Når Luftmotstanden som virker på en ball er proporsjonal med kvadratet av ballfarten, kan vi skrive uttrykket for ballens akselerasjon når den faller vertikalt slik: $a = -g + rv^2$.
Forklar hvordan vi kommer frem til uttrykket.

s412

$$\sum F = ma$$

$-G + L = ma$ der $G = mg$ og $L = kv^2$

$$-mg + kv^2 = ma$$

$$\underline{a = -g + \frac{k}{m} v^2}$$

15.04c) Vi lar programmet dele bevegelsen i ørsmå deler der aks. er tilnærmet konstant i hver del (tidsintervall).
Dermed kan vi bruke bev. likningene med konst. aks. for hver del.

15.05 a) Når r øker, ser vi at akselerasjonen raskere går mot null, farten gå raskere mot en konstant maksfart og posisjonsgrafen blir raskere en lineær funksjon.

$a[i+1] = -g + r*v[i]**2$ dvs. $a = -g + \frac{k}{m} \cdot v^2$

$v[i+1] = v[i] + a[i]*dt$

$s[i+1] = s[i] + v[i]*dt$

(Økt r betyr økt luftmotstand relativt til farten legemet har)

b) ca. $14{,}0 \frac{m}{s}$, ca. $9{,}9 \frac{m}{s}$

15.06 På grunn av luftmotstanden avtar aks. både i x- og y-retning med tiden. Grafene med og uten luftmotstand likner hverandre, men maks høyde er mindre og det tar kortere tid før makshøyen nås. Kastlengden (når $y=0$) er også mindre.

15.07 a) Når vinkelen avtar, avtar makshøyden for begge grafene. Kastlengden (når $y=0$) øker når startvinkelen endres fra 60° til 50°, endres lite fra startvinkel 50° til 40°, men avtar når startvinkelen reduseres videre til 30° og 20°.

b) ca 45° (41m) ca 45° (32m)

c) Endrer i linje 55 og 56 for å utvide koordinatsystemet.

15.08 Bare grafen for kast med luftmotstand endres. Kastet blir kortere og får lavere ~~høyd~~ makshøyde når luftmotstanden øker.

15.09 c) Vi ser at amplitude og periode er lik for begge grafene. Men grafen fra programmet viser fire hele perioder, mens grafen i eksempel 13.11 viser to og en halv periode.

15.10 a) Gjør de nødvendige endringene i program linje 11, 12 og 19 slik:

```
11  m = 0.15          # masse [kg]
12  k = 2.0           # fjærkonstant [N/kg]
19  x[0] = 0.060      # posisjon startverdi [m]
```

b) 1,7 s 6,0 cm $22 \frac{cm}{s}$ $80 \frac{cm}{s^2}$

6,0 cm $0,0 \frac{cm}{s}$ $-80 \frac{cm}{s^2}$

−2 cm $20 \frac{cm}{s}$ $25 \frac{cm}{s^2}$

c) Svarene stemmer med svarene i oppgave 13.26

15.11 Svingningene starter nå i likevektsstillingen med den oppgitte farten, og vi får startfarten som fartsamplitude. Fortegnet til farten bestemmer om startretningen er positiv eller negativ. Ellers er svingningene a og b like og c og d like.

15.12 c) Endre 4-tallet i linje 20 til et 2-tall.
d) Du kan for eksempel erstatte linje 10 med

```
n = int(input("Skriv inn verdien for n"))
```

15.13 Du kan erstatte linje 15 med:

```
for i in range(n):
```

15.14 c) Endre 1-tallet i linje 21 til et 3-tall

d) Endre programmet slik fra og med linje 3:

```
# Definer størrelser
T = float(input("Skriv inn temperaturen i kelvin"))

def p(V)                 # trykk [Pa]
    n = 1.00             # stoffmengde [mol]
    R = 8.31             # gasskonstanten [J/molK]
    return n*R*T/V

V1 = float(input("Skriv inn V1 i m^3"))
V2 = float(input("Skriv inn V2 i m^3"))
V = V1               # aktuell verdi volum [m^3]
N = 10000            # antall intervaller
```

15.15 a) Adiabatlikningen $p_1 V_1^\gamma = p_2 V_2^\gamma$ kan vi også skrive slik for enhver verdi av p og V når startverdiene er p_0 og V_0:

$$p(V)V^\gamma = p_0 V_0^\gamma$$

Vi dividerer likningen på begge sider med V^γ og får

$$p(V) = p_0 V_0^\gamma / V^\gamma$$

b) Du kan endre programmet slik fra og med linje 3:

```
# Definerer størrelser
def p(V):               # trykk [Pa]
 V1 = 0.010             # startverdi volum [m^3]
 g = 1.67               # Adiabatkonstanten
 p1 = 400000            # Starttrykk [Pa]
 return p1*V1**g/V**g

V1 = 0.010              # startverdi volum [m^3]
V2 = 0.030              # sluttverdi volum [m^3]
V = V1                  # aktuell verdi volum [m^3]
N = 10000               # antall intervaller
...
```