Algoritme for Eulers metode

for $i = 0, 1, 2, 3, \ldots, N-1$:

1. Bruk de forrige resultatene x_i og v_i for å regne ut akselerasjonen: $a_i = F(x_i, v_i, t_i)/m$.
2. Regn ut den nye farten: $v_{i+1} = v_i + a_i \Delta t$.
3. Regn ut den nye posisjonen: $x_{i+1} = x_i + v_i \Delta t + \frac{1}{2} a_i \Delta t^2$.

Algoritme for Eulers metode

for $i = 0, 1, 2, 3, \ldots, N-1$:

1. Bruk de forrige resultatene x_i og v_i for å regne ut akselerasjonen: $a_i = F(x_i, v_i, t_i)/m$.
2. Regn ut den nye farten: $v_{i+1} = v_i + a_i \Delta t$.
3. Regn ut den nye posisjonen: $x_{i+1} = x_i + v_i \Delta t + \frac{1}{2} a_i \Delta t^2$.

$\Downarrow$

```
for i in range(N):
    a[i] = F(x[i], v[i], t[i])/m
    v[i+1] = v[i] + a[i]*dt
    x[i+1] = x[i] + v[i]*dt + 0.5*a[i]*dt**2
```

$t_i \Rightarrow$ `t[i]` $\quad$ $v_i \Rightarrow$ `v[i]` $\quad$ $r_i \Rightarrow$ `r[i]`

```
# Import everything we need
from pylab import *

# Define physical parameters used in simulation
m = ...
g = ...

# Define time constants and number of steps
dt = ...
N = ...

# Define empty arrays
t = zeros(N+1)
a = zeros(N+1)
v = zeros(N+1)
x = zeros(N+1)
```

```
# Define inital conditions
v[0] = ...
x[0] = ...

# Define the force
def F(x, v, t):
    return ...

# Euler method loop
for i in range(N):
    t[i+1] = t[i] + dt
    a[i] = F(...)/m
    v[i+1] = ...
    x[i+1] = ...

# Plot results
plot(...)
xlabel(...)
ylabel(...)
grid()
show()
```

Luftmotstand

$$F_d = -\frac{1}{2}\rho C_D A|v|v.$$

Fritt Fall	
m	90 kg
g	9.81 $\mathrm{m/s^2}$
ρ	1 $\mathrm{kg/m^3}$
C	1.4
A	0.7 $\mathrm{m^2}$

Under fallskjerm	
C_p	1.8
A_p	44 $\mathrm{m^2}$

Oppgave 1

Skriv koden for å simulere en fallskjermhopper i fritt fall, ikke tenk på fallskjermen enda. Anta at hoppet starter ved 4000 meters høyden, og anta at hopperen starter med null hastighet mot bakken. Simuler fallskjermhopperen i 1 minutt.

(a) Plot absoluttverdien av farten, $|v|$.
Hint: Du kan bruke `abs(v)` for absoluttverdien.

(b) Hva er terminalhastigheten (maksfarten) til hopperne? Hvor lang tid tar det ca før hopperen når den farten?

(c) Plot fallskjermhopperens høyde over bakken mot tid.

(d) Hvor høyt over bakken er fallskjermhopperen etter 1 minutt i fritt fall? Hvor lang tid vil det ta før hopperen treffer bakken med denne farten?

La oss snakke om g-krefter

La oss snakke om g-krefter

Tross navnet, er g-krefter et mål på *akselerasjon*, ikke krefter. Det er definert som akselerasjonen fra summen av alle kontaktkrefter (altså *ikke* gravitasjon) målt i antall g.

La oss snakke om g-krefter

Tross navnet, er g-krefter et mål på *akselerasjon*, ikke krefter. Det er definert som akselerasjonen fra summen av alle kontaktkrefter (altså *ikke* gravitasjon) målt i antall g.

Til vanlig føler vi $1g$, fra kontaktkraften fra gulvet eller stolen. Hvis du føler $0g$ er du i fritt fall, det føler du for eksempel i stupet på en berg-og-dalbane.

La oss snakke om g-krefter

Tross navnet, er g-krefter et mål på *akselerasjon*, ikke krefter. Det er definert som akselerasjonen fra summen av alle kontaktkrefter (altså *ikke* gravitasjon) målt i antall g.

Til vanlig føler vi $1g$, fra kontaktkraften fra gulvet eller stolen. Hvis du føler $0g$ er du i fritt fall, det føler du for eksempel i stupet på en berg-og-dalbane.

Vi kan lett regne ut g-kreftene iløpet av bevegelsen

```
gforces = a/g + 1
plot(t, g)
```

Oppgave 2

La oss ta hensyn til fallskjermen. Vi simulerer de første 60 sekundene likt som istad. Når tiden når 60 sekunder, 'løser' vi ut fallskjermen ved å endre parametrene C_D og A.

(a) I Euler-løkka, legg inn en `if`-test som endrer C_D og A når tiden når 60 sekunder. Simuler hopperen i 2 minutter.
Hint: `if 60. < t[i] < 60. + dt:`

(b) Plot hastigheten og høyden over bakken.

(c) Hvor høyt over bakken er fallskjermhopperen etter 2 minutter? Ca hvor lenge til er det hun treffer bakken?

(d) Hva er terminalhastigheten med fallskjermen utløst?

(e) Plot g-kreftene som hopperen føler iløpet av hoppet. Diskuter med sidemannen og forklar endringene i g-kreftene. Er det realistisk? Hvorfor/Hvorfor ikke?

Oppgave 3

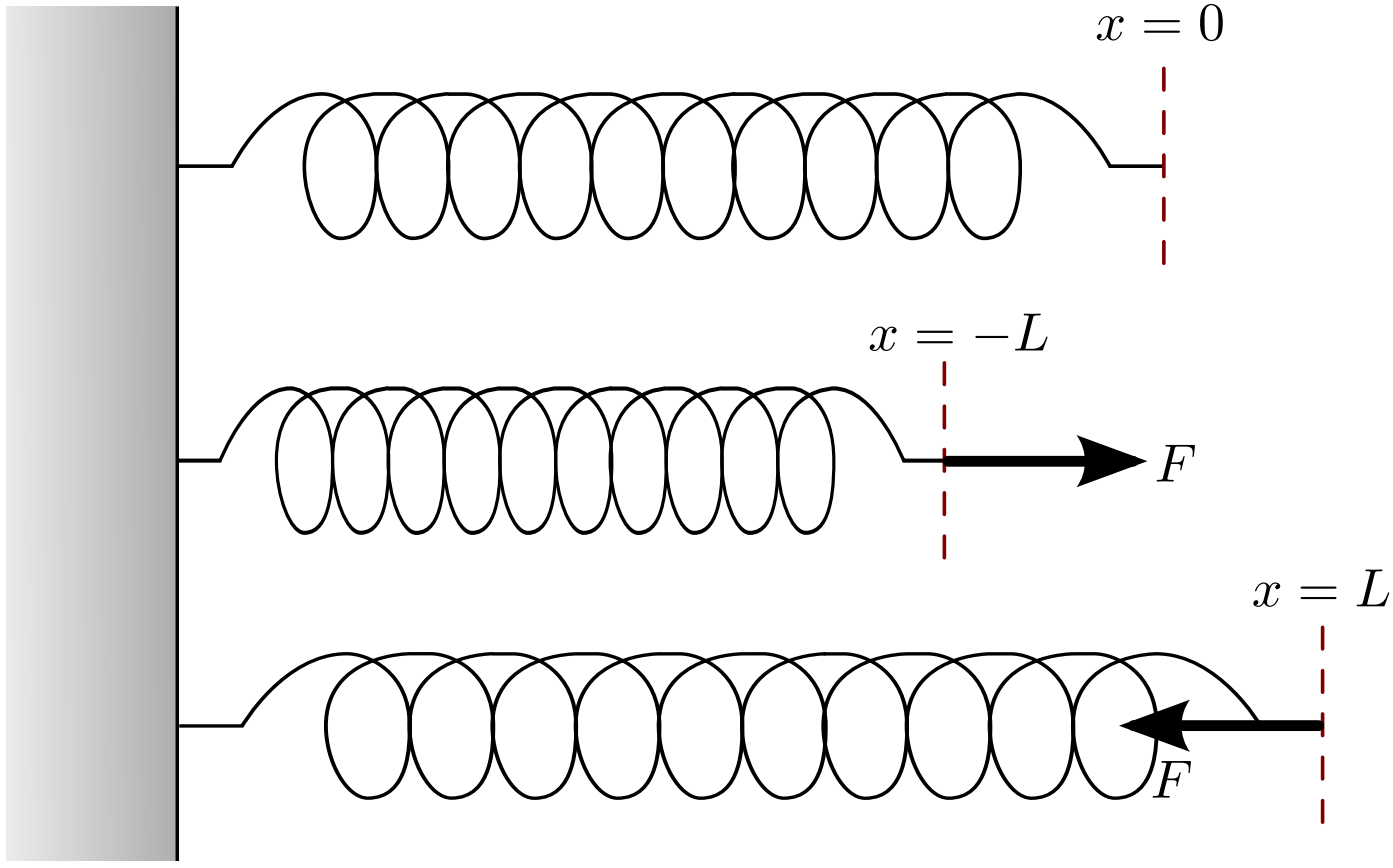
I forrige oppgave ble g-kreftene *altfor* store. De ville garantert drept enhver hopper. Problemet er at vi 'løste' ut fallskjermen altfor fort. Istedet for å endre parameteren rett fra en verdi til en annen, la oss endre dem over et par sekunder:

```
if 60. < t[i] < 65.:
    Cd += (Cd_p - Cd)/(5/dt)
    A  += (A_p  - A)/(5/dt)
```

(a) Endre koden som foreslått. Simuler helt til fallskjermhopperen når bakken.

(b) Hva er nå maks g-krefter hopperen opplever?

Fjærkraft



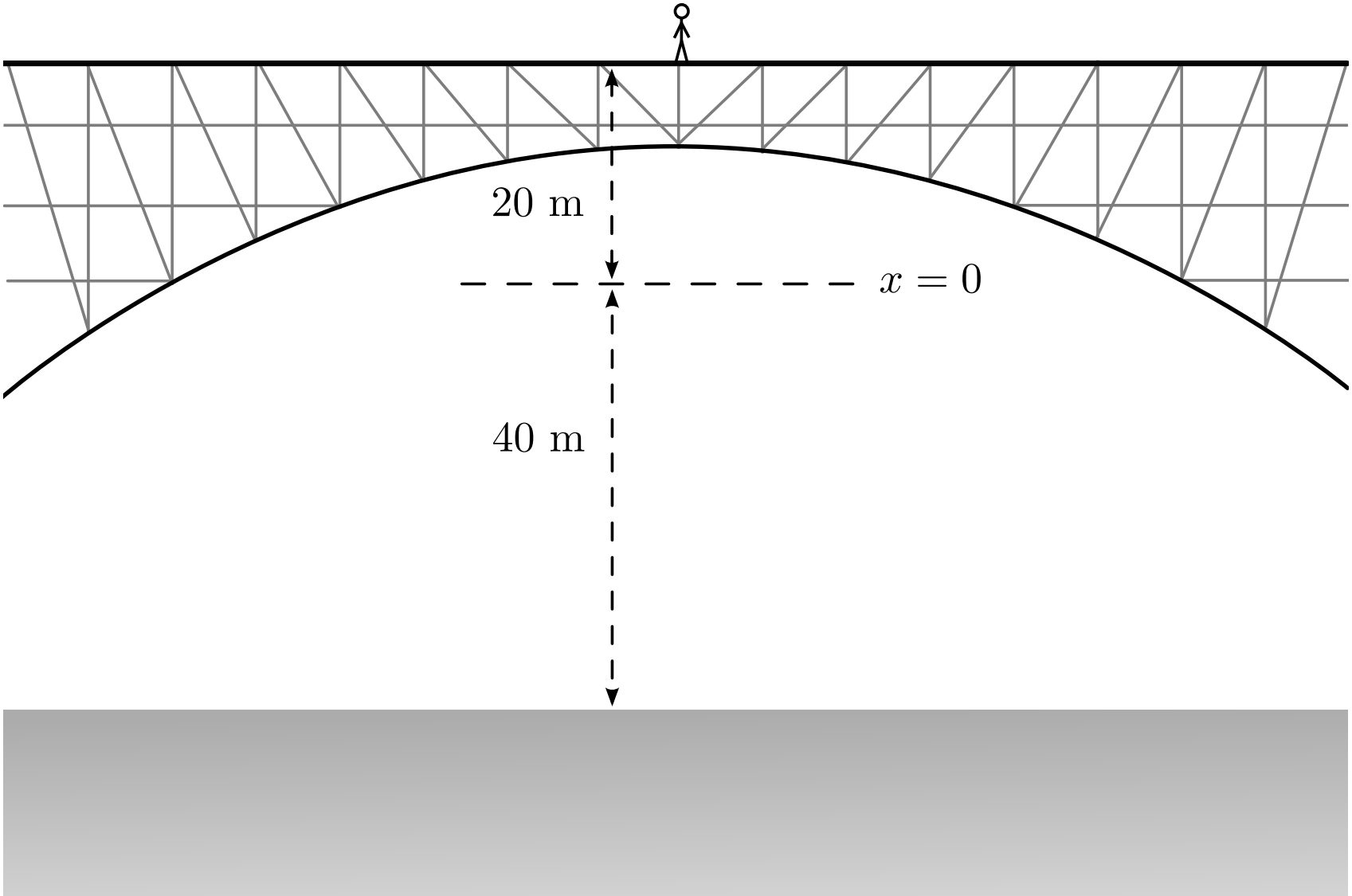
$$F_k = -kx \qquad \text{(Hookes lov)}$$

Strikkkraft

Strikkhopp er med en strikk, og ikke en fjær. En strikk ligner mye på en fjær, men virker bare 'en vei'.

$$F_k = \begin{cases} -kx & \text{if } x > 0 \\ 0 & \text{if } x < 0. \end{cases}$$

Strikkhopp



Oppgave 4

Kopier fallskjermhopp programmet ditt, og endre det så det gjelder for strikkhopp istedet. Gjett på en fjærstivhet, reduser massen til det du selv veier. Start hoppet fra 20 meter over likevektspunktet, anta at elven er 60 meter under brua (altså starter vannet på -40 meter).

(a) Prøv deg frem til du finner en fjærstivhet som gjør at du akkurat 'toucher' vannet.

(b) Hvor mange sekunder tar det fra du hopper fra brua til du treffer vannet?

(c) Simuler bevegelsen i flere minutter, ser bevegelsen realistisk ut? Hva kan gjøre at det ikke er helt realistisk? Diskuter med sidemannen.

(d) Plot g-kreftene. Sammenlign med fallskjermhopp