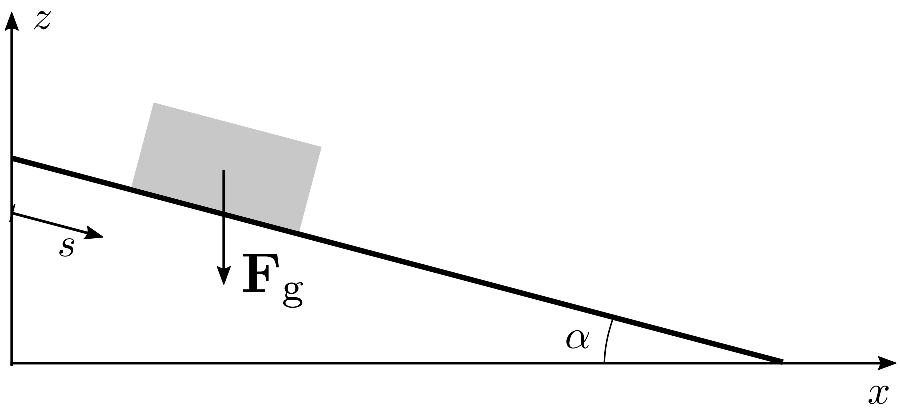
Kloss på skråplan – numerisk løsning av Newtons 2. lov



Figur 1: Kloss på skråplan. Planet danner vinkelen α med x-aksen. Tyngdekraften virker i negativ z-retning.

I denne øvingen skal vi bruke et meget enkelt eksempel til å illustrere hva det innebærer å løse et fysisk problem *numerisk*.

En kloss med masse m befinner seg på et skråplan som danner vinkelen α med x-aksen. La oss se helt bort fra friksjon og dermed gjøre problemet så enkelt som over hodet mulig. Tyngdekraften mg virker i negativ z-retning, som vist i figur 1. Denne kan dekomponeres i en kraft $mg \cos\alpha$ normalt på skråplanet og en kraft $mg \sin\alpha$ langs skråplanet. Ettersom klossen ikke har noen akselerasjon normalt på skråplanet, kan vi uten videre konkludere med at normalkraften fra skråplanet på klossen er $N = mg\cos\alpha$. Vi innser dessuten at det ikke virker andre krefter langs skråplanet, slik at Newtons 2. lov gir at klossen får den konstante akselerasjonen $a = g \sin\alpha$ nedover skråplanet.

Dette er et eksempel på et *eksakt* (evt *analytisk*) løsbart problem. Hvis vi nå kjenner for eksempel klossens posisjon s_0 og hastighet v_0 ved tidspunktet $t = 0$, kan vi bestemme dens posisjon og hastighet ved et senere tidspunkt med formlene

$$s(t) = s_0 + v_0 t + \frac{1}{2} g t^2 \sin\alpha \quad , \quad v(t) = v_0 + gt\sin\alpha.$$

Rent matematisk har vi her løst differensialligningen (Newtons 2. lov) $F = ma$ med $F = mg\sin\alpha$ og $a = d^2s/dt^2$, samt benyttet oss av at $v = ds/dt$.

Oppgaven handler om å løse ligningen

$$\frac{d^2s}{dt^2} = a$$

(med $a = g \sin \alpha$, en konstant) numerisk ved bruk av Eulers metode. Det lønner seg da å splitte ligningen – som er en andre ordens differensialligning – i to første ordens differensialligninger slik:

$$\frac{ds}{dt} = v,$$
$$\frac{dv}{dt} = a$$

Eulers metode går ut på å approksimere (tilnærme) de deriverte slik: $ds/dt \approx (s_{n+1} - s_n)/\Delta t$. Her er Δt tidsdifferansen mellom steg $n+1$ og n, dvs $\Delta t = t_{n+1} - t_n$, med $t_n = n\,\Delta t$. Høyresiden av de to ligningene over, hhv v og a, evalueres i tidssteg nr n.
a) Vis at de to differensialligningene kan skrives som

$$s_{n+1} = s_n + v_n \Delta t,$$
$$v_{n+1} = v_n + a\Delta t$$

Disse ligningene kan nå løses numerisk, som vist i følgende eksempler:

Python:

```python
import numpy as np

alpha = np.radians(60.0)  # Helningsvinkelen i radianer
g = 9.81  # Tyngdens akselerasjon
nsteps = 100  # Antall tidssteg
dt = 0.1  # Tidssteglengde
v0 = 1.0  # Starter med hastigheten 1.0 m/s
s0 = 0.0  # Starter ved posisjon s = 0.0

s = np.zeros(nsteps)
v = np.zeros(nsteps)

# Sett inn startbetingelser
s[0] = s0
v[0] = v0

# Akselerasjon langs planet
a = g*np.sin(alpha)
for n in range(0, nsteps-1):
    s[n+1] = s[n] + v[n] * dt
    v[n+1] = v[n] + a * dt
```

Matlab:

```
alpha = deg2rad(60.0);  # Helningsvinkelen i radianer
g = 9.81;  # Tyngdens akselerasjon
nsteps = 100;  # Antall tidssteg
dt = 0.1;  # Tidssteglengde
v0 = 1.0;  # Starter med hastigheten 1.0 m/s
s0 = 0.0;  # Starter ved posisjon s = 0.0

s = zeros(nsteps);
v = zeros(nsteps);

# Sett inn startbetingelser
s(1) = s0;
v(1) = v0;

# Akselerasjon langs planet
a = g*sin(alpha);
for n 1:nsteps-1
    s(n+1) = s(n) + v(n) * dt;
    v(n+1) = v(n) + a * dt;
end
```

b) Lag, med utgangspunkt i et av disse eksemplene, et fullstendig program i Python eller i Matlab, som løser bevegelsesligningene for en kloss på et skråplan. Plott den numeriske løsningen og den analytiske løsningen i samme figur for å sammenligne dem. Varier tidssteget Δt, men endre antall tidssteg slik at hver kjøring av programmet beregner posisjonen fram til samme maksimale tidspunkt. Hvordan varierer den numeriske feilen (dvs avviket fra den eksakte løsningen) med tidssteget?