Bernt Jonas Fløde

Project 3 Computational Physics

12. november 2018

Introduksjon

Dette projektet tar sikte på å simulere planetenes og Solas baner i Solsystemet ved hjelp av den forlengse Euler-metoden og Verlet-integrasjon. Metodene sammenlignes, og vi ser hva ulike tidssteg gjør for nøyaktigheten. Vi finner også ut hvor stor innvirkning det har å anta at Sola står stille i Solsystemets massesentrum. Til slutt ser vi litt på den relativistiske effekten på Merkurs bane.

Metode

Fysikk

Til å finne planetbanene brukes den klassiske newtonske ligninga for gravitasjonell akselerasjon:

$$a_G = \frac{GM_\odot}{r^2},$$

der $M_\odot$ er Solas masse, G er gravitasjonskonstanten og r er avstanden fra Sola til himmellegemet som skal få akselerasjonen sin utregnet. Formelen kan også brukes for å regne ut tyngdeakselerasjonen fra hvilket som helst himmelegeme på et annet, ved å bytte ut solmassen med massen til det legemet som er kilde til gravitasjonsfeltet.

For å gjøre uttrykket relativistisk, gjøres det en liten endring i uttrykket ved å ta med en ekstra faktor:

$$a_G = \frac{GM_\odot}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right]$$

I en del av prosjetet skal det demonstreres sirkulære planetbaner, og for å finne dette brukes det at sentripetalakselerasjonen skal være lik gravitasjonell akselerasjon. Utregninga går slik:

$$a_G = v^2/r = GM_m athrm\odot/r^2 v = (GM_m athrm\odot/r)^{0.5}$$

For å nå unnslippingshastighet, må kinetisk energi være minst like stor som negativ potensiell energi. Dvs.

$$1/2M_{planet}v^2 = GM_m athrm\odot M_{planet}/rv^2 = 2GM_m athrm\odot/AUv = (2/GM_m athrm\odot/AU)^{0.5}$$

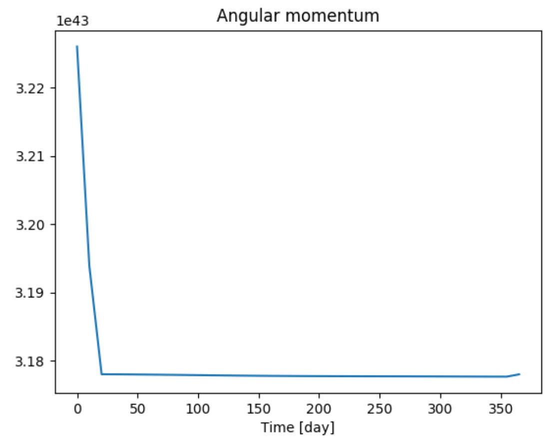
der AU er en astronomisk enhet.

I programmet escape_velocity.py er det lagd en animasjon som viser en planet som akkurat har unnslippingshastigheten, samt den samme banen der gravitasjonsloven er justert slik at 2-eksponenten har forskjellige verdier mellom 2 og 3.

Programmering

Med forlengs Euler metoden enkel. Først regnes akselerasjonen for et gitt tidspunkt ut, så ganges den med et gitt tidssteg, og med dette har vi funnet hastighetsendringa. Tilsvarende gjøres for posisjonsendringa, ved å gange hastighet med tidssteget. I kodeform kan dette skrives

```
for (i = 0; i < N; i++)
    v[i+1] = v[i] + dt * a_G(r[i])
    r[i+1] = r[i] + dt * v[i]
```



Figur 1:

der v er hastighetene, r er posisjonene for hver gitte tidsindeks i, N er antall elementer i listene, dt er tidssteget og a_G er gravitasjonens akselerasjon.

Verlet-integrasjon virker annerledes, og her regnes i utgangspunktet bare posisjonen ut. Her gis den bare i kodeform:

```
current_x = r[0] + v[0]*dt + 0.5*a_G(x[0])*dt**2
for (i = 0; i < N; i++):
    next_x = 2*current_x - prev_x + a_G(current_x)*dt**2
    prev_x = current_x
    current_x = next_x
    x[i] = current_x
```

For å finne hastighetene, må vi her derivere posisjonen numerisk:

```
v[i] = (r[i+1] - r[i-1]) / (t[i+1] - t[i-1])
```

der t er tidspunktet for en gitt tidsindeks i.

I begge tilfellene er det en forutsetning av vi kjenner startposisjon r[0] og starthastighet v[0].

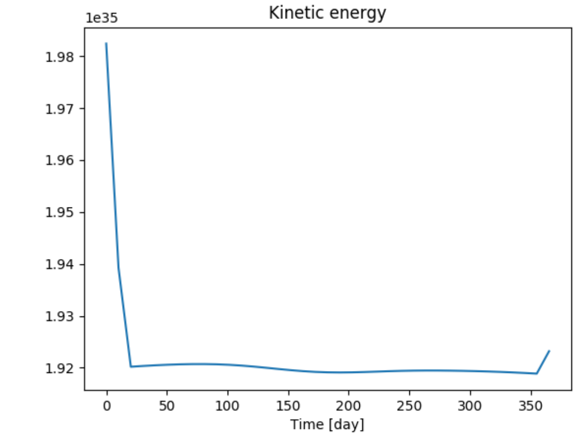
Datakilde

Data for himmelegemenes startposisjon og -hastighet hentes fra NASA: https://ssd.jpl.nasa.gov/horizons.cgitop

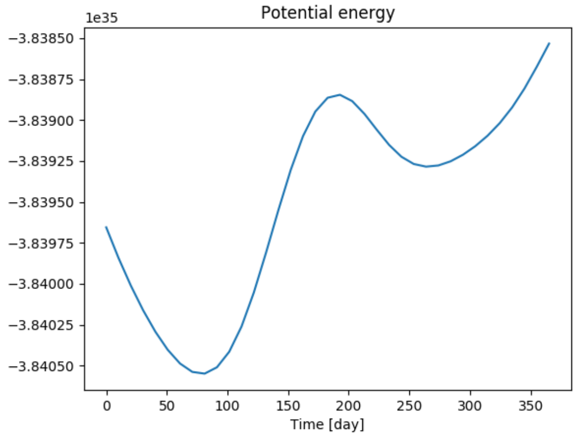
Resultat

I figur 1, 2 og 3 kan vi se at både drivmoment (eng: angulat momentum), kinetisk energi og potensiell energy ligger stabilt sett utfra tallene på y-aksen. Dette gjelder altså for tilnærmet sirkulær planetbane.

Når det gjelder FLOP (flyttallsoperasjoner), viser en rask telling at dette er 6 for framlengs Euler og 5 for Verlet, for hvert iterasjon. Hvis vi tar med hastighetsutregningen med Verlet blir det derimot 9 for Verlet.



Figur 2:



Figur 3: