

Prosjekt 3 FYS3150 H13

Jostein Granheim Trøyflat

Link til prografil: <https://github.com/josteingt/myfiles>

Innledning

I dette prosjektet er målet å simulere bevegelsen av planetene i solsystemet. Dette gjøres ved å løse newton's andre lov for alle planetene samtidig. Runge kutta 4 brukes for å løse problemet. I prosjektet brukes gjennomgående astronomiske enheter for avstand, masse og tid. 1 AU brukes for lengde og er avstanden fra jorden til sola, masse angis i solmasser og tid måles i år.

Teori

Newton's andre lov vil for hver enkelt planet være på formen

$$\begin{aligned}\frac{d^2x}{dt^2} &= \frac{\sum F_x}{m} \\ \frac{d^2y}{dt^2} &= \frac{\sum F_y}{m}\end{aligned}\tag{1}$$

Disse andre ordens differensialligningene kan for hver koordinat skrives om til to koblede første ordens differensialligninger.

$$\begin{aligned}\frac{dx}{dt} &= v \\ \frac{dv_x}{dt} &= \frac{\sum F_x}{m}\end{aligned}\tag{2}$$

For et system med flere planeter vil alle første ordens ligningene for alle planetene være koblet og må derfor løses samtidig. I dette prosjektet brukes runge kutta 4 som metode for å løse dette settet med koblede differensialligninger. Runge kutta metoder kan brukes til å løse differensialligninger på samme form som i ligning 3.

$$\frac{dy}{dt} = f(t, y)\tag{3}$$

Den generelle ideen i runge kutta metoder er gitt ved ligning 4. For å finne neste steg beregnes integralet over et lite tidsintervall. Integralet kan beregnes numerisk med forskjellige metoder, og hvilken metode som benyttes bestemmer orden av runge kutta metoden. Siden det i dette prosjektet brukes runge kutta 4 vil et tidssteg ha samme form som i ligningene i 5

$$y(t+h) = y(t) + \int_t^{t+h} f(t, y) dt\tag{4}$$

$$\begin{aligned}y_{i+1} &= y_i + \frac{1}{6}(k1 + k2 + k3 + k4) \\ k1 &= hf(t_i, y_i) \\ k2 &= hf(t_i + h/2, y_i + k1/2) \\ k3 &= hf(t_i + h/2, y_i + k2/2) \\ k4 &= hf(t_i + h, y_i + k3)\end{aligned}\tag{5}$$

Fordi ligningene må løses samtidig må $k1$, $k2$, $k3$ og $k4$ for alle planetene beregnes før det er mulig å ta et tidssteg. I algoritmen under er det bare

et steg for 1 planet. Ideen er å oppdatere posisjonene til planetene også for halvstegene slik at det blir lettere å beregne krefter for halvstegene. Dette betyr at initialverdiene må lagres for å kunne ta hele steget til slutt.

```
void RK4(){
    //storing initial values
    initxvel = xvel
    inityvel = yvel
    initxpos = xpos
    initypos = ypos

    //calculate forces for initial values
    force(&forcex,&forcey)

    //find k1s
    k1xvel = h*forcex
    k1yvel = h*forcey
    k1xpos = h*xvel
    k1ypos = h*yvel

    //updating values
    xvel = inintxvel + h/2*k1xvel
    yvel = inintyvel + h/2*k1yvel
    xpos = initxpos + h/2*k1xpos
    ypos = initypos + h/2*k1ypos

    //calculating new forces for the halfstep positions
    force(&forcex,&forcey)

    //finding k2s
    k2xvel = h*forcex
    k2yvel = h*forcey
    k2xpos = h*xvel//velocities are updated to halfstep
    k2ypos = h*yvel

    //updating values
    xvel = initxvelh + h/2*k2xvel
    yvel = inityvel + h/2*k2yvel
```

```

xpos = initypos + h/2*k2xpos
ypos = inintypos + h/2*k2ypos

//calculating new forces for the halfstep positions
force(&forcex,&forcey)

//finding k3s
k3xvel = h*forcex
k3yvel = h*forcey
k3xpos = h*xvel//velocities are updated to halfstep
k3ypos = h*yvel

//updating values
xvel = initxvelh + h*k3xvel//takes the whole step
yvel = inityvel + h*k3yvel
xpos = initypos + h*k3xpos
ypos = inintypos + h*k3ypos

//calculating new forces for the endpoint
force(&forcex,&forcey)

//finding k4s
k4xvel = h*forcex
k4yvel = h*forcey
k4xpos = h*xvel//velocities are updated to endpoint
k4ypos = h*yvel

//making the entire step
xvel = initxvel + 1/6*(k1xvel + k2xvel + k3xvel + k4xvel)
yvel = inityvel + 1/6*(k1yvel + k2yvel + k3yvel + k4yvel)
xpos = initxpos + 1/6*(k1xpos + k2xpos + k3xpos + k4xpos)
ypos = initypos + 1/6*(k1ypos + k2ypos + k3ypos + k4ypos)
}

```

For å utvide algoritmen til å klare flere planeter er hovedforskjellen at flere initialverdier må lagres og flere k må lagres før det er mulig å ta et tidssteg. Utfordringen ligger i å kunne beregne krefter på hver enkelt planet ikke bare for inintialverdier, men også for halvsteg. I programmet er derfor

en solsystem klasse som inneholder alle planetene. I denne klassen ligger metoden for runge kutta 4 og metoden for beregne krefter. Kreftene kan da beregnes for halvstegene ved å oppdatere posisjonene til hver enkelt planet for halvstegene. Men fordi posisjonene til planetene blir forandret underveis må initalposisjonene lagres før bereningene starter slik at det er mulig å ta et helt tidssteg til slutt. I tillegg til en solsystem klasse er det en planet klasse som samler informasjon om hver enkelt planet og har metode for å sette initialverdier til planetene.

På grunn av enhetene som brukes vil verdien for gravitasjonskonstanten forandres, verdien kan bestemmes ved å se på jorda i sirkelbane rundt sola.

$$\begin{aligned}
 m_{j\text{ord}} \frac{v^2}{r} &= F \\
 \frac{v^2}{r} &= \frac{GM_{\text{sol}}m_{j\text{ord}}}{r^2m_{j\text{ord}}} \\
 v^2r &= GM_{\text{sol}} \\
 M_{\text{sol}} &= 1 \\
 r &= 1 \\
 v &= \frac{2\pi r}{1\text{year}} \\
 v^2 &= 4\pi^2 \\
 G &= 4\pi^2
 \end{aligned} \tag{6}$$

Fordi det ikke er noen ytre krefter på solsystemet (som blir tatt hensyn til) vil energien være bevart.

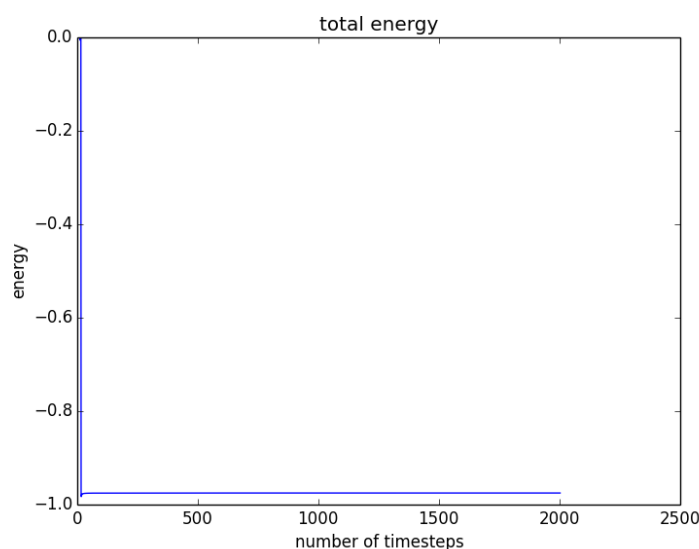
Resultat

For system bestående av få planeter med ”pene” initialverdier som gir sirkelbevegelse holder energien seg konstant, som tyder på at programmet fungerer fint for disse systemene. Med flere planeter begynner energien å oscilere, men den har ingen tydelig tendens til å gå opp eller ned i verdi over lang tid. Bevegelsene ser fortsatt ut til å være fornuftige i plot selv om energien oscilerer.

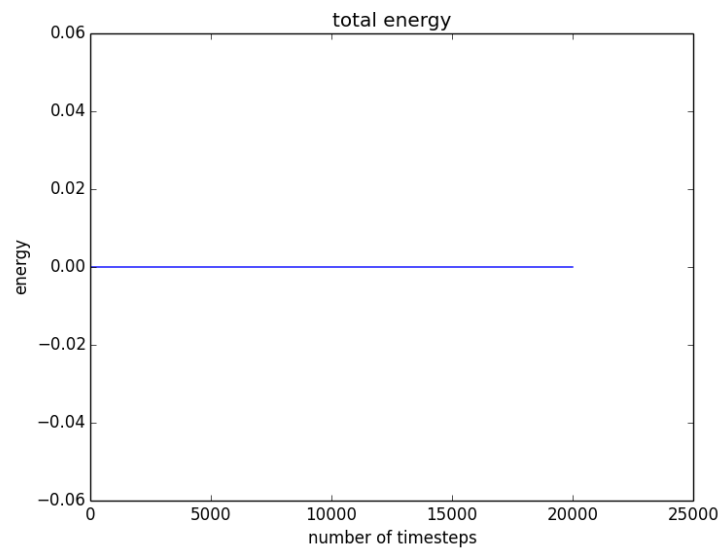
For det enkleste systemet med bare jorden og solen, vil systemet holde seg stabilt ned til en steglengde på 0,01 år som vist i figurene 1 og 2. 0,01 år tilsvarer 3,6 grader av sirkelbevegelsen til jorden. Ved å legge til jupiter med initialhastighet som gir sirkelbevegelse vil systemet også være stabilt for 0,01 år i steglengde. Men ved initialhastighet som gir eliptisk bane vil energien begynne å oscilere som vist i figur 3, selv om bevegelsen fortsatt ser fornuftig ut i et plot.

Kjøring av hele solsystemet krever en steglengde på 0,001 år for å være stabilt, og her vil energien oscilere selv for initialhastigheter som skulle tilsi sirkelbevegelse. Initialverdier som ikke gir sirkelbevegelse gir en mindre periodisk oscilerende energi. Og om man setter flere planeter til å ha eliptiske baner er det overhengende fare for at en planet unnslipper solsystemet.

Figur 1: Energiavvik fra startenergi for steglengde lik 0.1 år, kjørt i 10000 år, jord og sol



Figur 2: Energiavvik fra startenergi for steglengde lik 0.01 år, kjørt i 10000 år, jord og sol



Figur 3: Energiavvik fra startenergi for steglengde lik 0.01 år, kjørt i 1000 år, jord, sol og jupiter med initialhastighet som gir eliptisk bane

