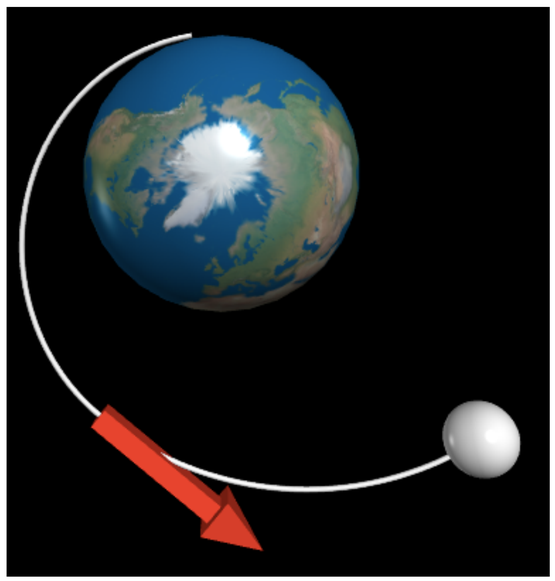
Satellitters – bevægelse i et tyngdefelt

Introduktion

Modellen er afslutningen på et forløb om newtons mekanik i 1.g, uden differentialregning. Eleverne laver simulation af bevægelse i tyngdefelt. Forløbet er lavet i Glowscript som er en del af Python og kan findes her, kortlink.dk/x5kw. Figuren viser hvordan en raketaffyring kan bringe en sattellit i kredsløb om Jorden.



Mål med aktiviteten

Fysikfaglige mål var at træne elevernes forståelse for kinetisk, potentiel og mekanisk energi. Mere specifikt at kunne forklare graferne for de tre energiformer ved bevægelse i et ikke homogent tyngdefelt. Eleverne fik derudover en anderledes introduktion til Newtons tyngdelov, hvor de selv prøvede den af.

CT-faglige mål var at eleverne skulle styrkes i

- Trinvis tilgang til løsning af fysikproblemer.
- Læse kode.
- Rette i kode.

Beskrivelse af aktiviteten

Modulerne:

1. Introduktion, Former og Bevægelse
2. Newtons tyngdelov og bevægelse
3. Newtons tyngdelov, den generelle
4. Jorden og Månen, tur retur.

Første modul gik med at lege med former og bevægelse i programmet. Eleverne blev fortrolige med orienteringen i programmet og brugte while loops til at gøre simuleringen dynamisk.
Andet modul gik med at bevægelse i homogent tyngdefelt hvor de numerisk gik fra a til v til x. Eleverne skulle også kommentere på (t,Ekin),(t,Epot),(t,Emek) grafer som dynamisk blev skabt.
Tredje modul Den generelle tyngdelov blev præsenteret og eleverne arbejde med lodret og vandret kast samt forsøgte at få en sattellit i orbit.
Fjerde modul afsluttede eleverne med at simulere Apollo 8 missionens rundtur om Månen

Centralt loop i simuleringen

```
# loopet som laver simulationen
while t<60*60*4:
  rate(500)
  r=sat.pos-Earth.pos
  F=-G*mSat*mEarth/(mag(r)**3)*r
  if (t>(60*60) and a==0):
    F = F + push*norm(sat.v)
    arrow(pos=sat.pos, axis=norm(sat.v)*1e7, color=color.red)
    a=1
  sat.v=sat.v+F/mSat*dt
  sat.pos=sat.pos+sat.v*dt
```

Simuleringen er over 4 timer, jordtid, men kan indstilles med rate() til hurtigere computertid, her ca. 10 sekunder.
Tyngdekraften beregnes i linje 4 og hastighed og position opdateres i linje 9 og 10 med v = v+F/m·dt, x = x+v·dt.
Efter en time bliver en raket affyret og kraften opdateres med F = F + push*norm(sat.v), norm(sat.v) giver en enhedsvektor i satellittens bevægelsesretning.

Didaktiske overvejelser

Elevresponse var generelt positivt. På spørgsmål om de havde fået bedre styr på Ekin og Epot og dets grafer var svaret klart ja. Hvorvidt eleverne nu følte de kunne programmere i Python var mere usikkert. De fleste mente at de kunne læse koden og lave mindre rettelser til den, mens de færreste følte sig sikre nok til selv at starte fra bar bund. Eftersom det heller ikke var meningen var der generel tilfredshed blandt eleverne og læreren.

Fagligt virker dynamiske visualiseringer hvor man kan følge systemet udvikle sig mens man ser graferne blive tegnet. Det er vigtigt at simuleringen kører så langsomt at eleverne kan nå at se på graferne mens den kører.

Eleverne fandt ret hurtigt ud af at rette i koden og virkede ikke skræmte af syntaksen. De kunne sagtens have brugt mere tid med bare at tegne figurer og få dem til at bevæge sig end vi brugte. Det visuelt skabende virker motiverende, hvad enten det er en vektor de får til at roterer som viserne på et ur eller en satellit de får til at bevæge sig omkring Jorden.

Forsøget på at simulere Apollo 8 missionen var ret svær. Det er svært at ramme Månen og bremse rigtigt så Månens svage tyngdefelt kan hiver satellitten rundt. Det lykkedes for nogen, men en del syntes at der gik for meget finpudsning af parametre i den.

Timerne havde et klart fysikfagligt mål med fokus på Månen, men næste gang kunne jeg godt tænke mig at give eleverne lidt mere frihed til selv at vise noget de synes er sjovt. Det kunne ske ved at lade sidste modul være mere kreativt.

Kreditering

Python-modellen og undervisningsmaterialet er udviklet af Mads Peter H. Steenstrup, Rysensteen gymnasium, i forbindelse med deltagelse i udviklingsprojektet Computational Thinking i Matematik og Naturfag i skoleåret ‘18/’19. Projektet køres i samarbejde mellem Danske Science Gymnasier og Center for Computational Thinking & Design, Aarhus Universitet.