

## Uge 4 labøvelse

### Fjederkanon simulation i Python

Daniel Lawther, Edin Ikanovic, Rune Mikkelsen, Jesper Hornebo Jensen  
Hold 4 gruppe 6

Vi simulerer vores fjederkanon eksperiment ved at lave et vPython-program.

Vi kan beskrive bolden som en partikel for at gøre vores beregninger mere simple, idet vi kan se bort fra boldens rotation om sin egen akse. Vi ser også bort fra luftmodstand.

I vores program fra sidste uge spillede massen og størrelsen af bolden ingen rolle. I det nye program har vi defineret en masse og lavet et udtryk for accelerationen ud fra massen. Da massen er konstant spiller dette ikke nogen praktisk rolle (massen går ud i linjen 'netA=..' i vores program). Hvis vi havde gnidningskraften med i vores beregninger så ville massen have en betydning.

Vi opererer her med en konstant tyngdekraft som den eneste kraft. Hvis vi havde en kraft som var beskrevet som en funktion af tiden så kunne vi også skrive denne ind i vores program. X-værdierne ved nedslag lægges ind i et array i programmet, og printes også på skærmen.

Vi har indlagt et plan i vores program, der ligger lige så langt fra boldens startposition i y retningen som højden fra kanonmunden til gulv i vores forsøg. Vi har fået programmet til at stoppe bolden og genstarte bevægelsen når bolden har samme y-koordinat som vores plan.

I første omgang får vi et program, der bare gentager samme beregninger i det antal shots, der er defineret i maxShots=. Derfor får vi altid det samme resultat uanset om vi skyder 10, 100 eller 1000000 gange (og vi har prøvet!).

Vi indlægger derfor en Gauss-usikkerhed i vores nedslagslængde, lig med den målte standardafvigelse i forsøget.

Til sidst får vi programmet til at lave statistiske beregninger på de data, vi indsamlede i vores array. På grundlag af dataen tegnes der også et histogram og et nedslagsplot.