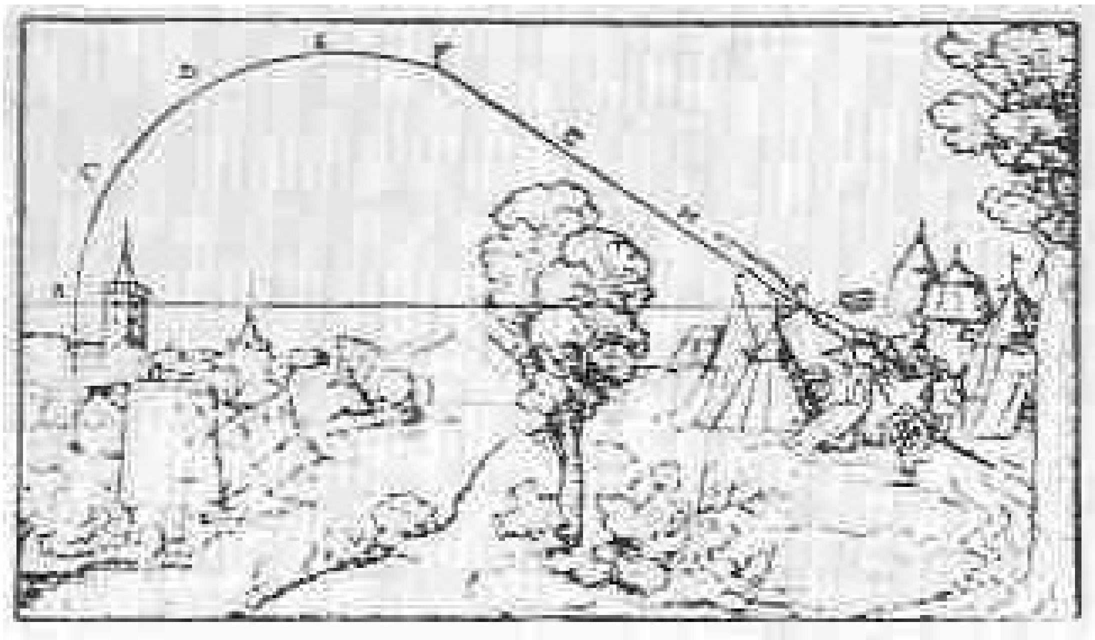
Bevægelse med luftmodstand



Banekurve beskrevet af Albert af Sachsen. Kilde: Fysikhistorie.dk.

Problemformulering

At bestemme stedbevægelsen for et legeme, der bevæger sig i tyngdefeltet og som er påvirket af luftmodstand.

Problemstillinger

1. Bevægelsesligningerne for et legeme, der bevæger sig i tyngdefeltet kan udledes analytisk i tilfældet, hvor kun tyngdekraften påvirker bevægelsen.
2. I virkeligheden er der luftmodstand at tage hensyn til, og dette fænomen, skal man tage hensyn til, når bevægelsen af et legeme - f. eks. en vulkansk bombe - skal beregnes.
3. I visse tilfælde betyder luftmodstanden mere end i andre, fordi masse/overfladearealet af et legeme ikke er konstant.
4. Endelig påvirker den geometriske form af legemet også bevægelsen.

Illustration 1: Vulkanske bombers spor kan ses på billedet. Kilde: https://krakatoavolcanoeruption.weebly.com/background.html.

Teori

I FysikABbogen 2 kap. 5-6 er der angivet analytiske udtryk for specielle bevægelser, for eksempel bevægelse i tyngdefeltet uden luftmodstand samt jævn cirkelbevægelse og harmonisk bevægelse. Her skal du undersøge bevægelsen for et legemes bevægelse i tyngdefeltet under hensyntagen til luftmodstanden.

Du kan ikke løse problemet analytisk, så du bliver nødt til at løse opgaven numerisk. Dertil kan du benytte TI nSpire eller Modellus.

Bevægelsesligningerne kan opstilles ved hjælp af definitionerne af hhv. hastighed og acceleration:

$$\vec{v}=\frac{d\vec{r}}{dt}\wedge\vec{a}=\frac{d\vec{v}}{dt}\Leftrightarrow d\vec{r}=\vec{v}\cdot dt\wedge d\vec{v}=\vec{a}\cdot dt\Rightarrow\Delta\vec{r}\approx\vec{v}\cdot\Delta t\wedge\Delta\vec{v}\approx\vec{a}\cdot\Delta t \tag{1}$$

Derudover skal man anvende Newton's 2. lov til at finde udtryk for accelerationen på legemet.

<u>Eksempel:</u>
Ved opstillingen af bevægelsesligningerne på en form, som en computer kan forstå, kan man ikke regne med differentielle størrelser, så derfor må man lave en approksimation af udtrykkene for sted, hastighed og acceleration. De er udtrykt i sidste udtryk i (1).

Vi betragter et en-dimensionalt problem, nemlig en kugle der falder i tyngdefeltet uden luftmodstand. Her vil Newton's 2. lov give os et udtryk for accelerationen, som i dette tilfælde er konstant og nedadrettet. Hastighedstilvæksten i det lille tidsrum, Δt, bliver så

$\Delta v = F_{res}\cdot\Delta t/m$.

Hastighedsændringen efter tidsrummet Δt bliver så

$v(t+\Delta t)= v(t)+\Delta v$

Tilsvarende får man for stedkoordinaten

$\Delta y= v\cdot\Delta t$
$y(t+\Delta t) = y(t)+\Delta y$

Prøv at sammenligne udtrykkene ovenfor med definitionerne af hastighed og acceleration.

Ved at lade tiden vokse, kan man altså bestemme stedbevægelsen. I nSpire opskriver man bevægelsesligningerne på følgende måde:

```
F:=-m·g {Positiv retning er valgt opad.}
dv:=F·dt/m
v:=v+dv
dy:=v·dt
y:=y+dy
t:=t+dt
```

Bevægelse med luftmodstand

Når en partikel bevæges i tyngdefeltet, vil den blive påvirket af tyngdekraften samt luftmodstanden. Kraften fra luften er afhængig af hastighedens størrelse. Hvis hastigheden er 'lille' er gnidningskraften proportional med hastigheden og hvis hastigheden er 'stor' er kraften proportional med *kvadratet* på hastigheden. (Dvs. for store hastigheder stiger gnidningskraften relativt set meget voldsommere end for små hastigheder.)

Vi vil kun betragte en kugles bevægelse i tyngdefeltet, hvor hastigheden er 'stor.' Her viser det sig, at man kan opskrive gnidningskraften på følgende vis:

$$\vec{F}_{luft} = -\frac{1}{2} \cdot c_w \cdot \rho_{luft} \cdot A \cdot v^2 \cdot \vec{e} \qquad (2)$$

Ovenfor er A et tværsnitsareal af kuglen, $c_w = 0{,}40$ (formfaktoren kaldes også for c_w-værdien og den varierer afhængig af legemets form, (se illustration 3 for forskellige former, vulkanske bomber kan have), ρ = 1,25 kg/m^3 (lufts massefylde), og v er kuglens fart. Luftmodstandsskraften peger i øvrigt hele tiden modsat bevægelsesretningen. $\vec{e}$ angiver en retningsvektor for stedbevægelsen. Illustration 2 viser kræfterne, som påvirker kuglen.

Illustration 2. Sporet af et flyvende legeme med kræfterne indtegnet.

Find enhedsvektorens 1. og 2. koordinater udtrykt ved trigonometriske funktioner og v_x- og v_y-koordinaterne, som antages kendte. (**Hint:** Indtegn hastighedsvektoren i illustration 2 og find vinklen mellem v_x og v eller v_y og v. Hvilken retning peger hastigheden imod i forhold til luftmodstandskraften?)

Hvis du er helt ny i programmering, så hop nu til Appendixet og læs det og løs de øvelser, der står der. Hvis du kender nSpire BASIC, kan du læse videre herunder.

Simulationens kode

Lodret fald.

a) Betragt en partikel med massen, m, der er placeret i højden, y_0, over et valgt nulpunkt. Skitser situationen, indtegn et valgt koordinatsystem og tegn de kræfter, der virker på partiklen.
b) Start nSpire op. Vælg programmodulet, giv programmet et passende navn. (Se evt appendix) og skriv startværdierne for acceleration, hastighed, position og tid samt konstanterne.
c) Opstil derefter udtryk for accelerationen, hastigheden og positionen. Beregningerne skal gentages mange gange, så udtrykkene skal ligge i en *For-EndFor*-løkke.
d) Husk at sørge for at koordinater og hastigheder efter hver eneste løkkekørsel skal gemmes i nogle generelle variable, så tallene gemmes til udskrivningen.

e) Tegn følgende grafer: En (tid, sted)-graf, en (tid, fart)-graf samt en (tid, acceleration)-graf.

Generelt kast.

Betragt situationen fra før, men med den tilføjelse, at partiklen nu kan tilføres en starthastighed i en vilkårlig retning i (x, y)-planet.

a) Generaliser modellen fra før, så den nu tager højde for bevægelse i både vandret og lodret retning. Find maksimum-højden på en (tid, sted)-graf og sammenlign resultatet med en teoretisk udregning.
b) Tilføj luftmodstanden i din formel for $\vec{F}_{res}$
c) Betragt en vulkansk *bombe*, der vejer 1000 g, har en densitet på 1800 kg/m^3 og som skydes ud fra toppen af en 800 m høj vulkan med farten 30 m/s med en vinkel på 10° i forhold til lodret. Vælg selv en passende form for stenen, beregn dens tværsnitareal og find en værdi for formfaktoren. Beregn dens bane.

Rapportering

Udskrift af

- modellen for det generelle kast
- værdier
- graf

skal vedlægges et kort diasshow på Google-docs, og der skal følge forklaringer med på modellerne.

Ekstra - kontrol af beregningerne

Du testede dit program ved at sammenligne kastelængde og stighøjde i tilfældet hvor, der ikke er luftmodstand. Man kan også delvist teste tilfældet med luftmodstand inkluderet. Det kan du gøre her.

Betragt et lodret fald - det kunne være en muffinform, der lades falde. Hvis man vælger den positive retning til at pege nedad, bliver bevægelsesligningen $\frac{dv}{dt} = g - \frac{\rho_{luft} \cdot c_w \cdot A \cdot v^2}{2 \cdot m}$. Løsningen til denne ligning er

$$v(t) = v_\infty \cdot \left(\frac{e^{\frac{2 \cdot g}{v_\infty} \cdot (t - t_0)} - 1}{e^{\frac{2 \cdot g}{v_\infty} \cdot (t - t_0)} + 1} \right) = v_\infty \cdot \tanh\left(\frac{g}{v_\infty} \cdot (t - t_0)\right), hvor\ v_\infty = \sqrt{\frac{2 \cdot m \cdot g}{\rho_{luft} \cdot c_w \cdot A}}.$$

Juster dit program til at simulere et frit fald, og beregn også den teoretiske hastighed til hver tidspunkt ved hjælp af formlen ovenfor. Sammenlign v(t) med de hastigheder, dit program integrerer sig frem til.

Hvis du vil være ekstra særlig omhyggelig kan du også teste stedfunktionen for det lodrette fald. Forskriften står nedenfor.

$$s(t)=\frac{v_\infty^2\cdot\log\left(\cosh\left(\frac{g\cdot t}{v_\infty}\right)\right)}{g}, \text{ hvor } v_\infty=\sqrt{\frac{\rho_{luft}\cdot c_w\cdot A}{2\cdot m\cdot g}}.$$

God arbejdslyst.

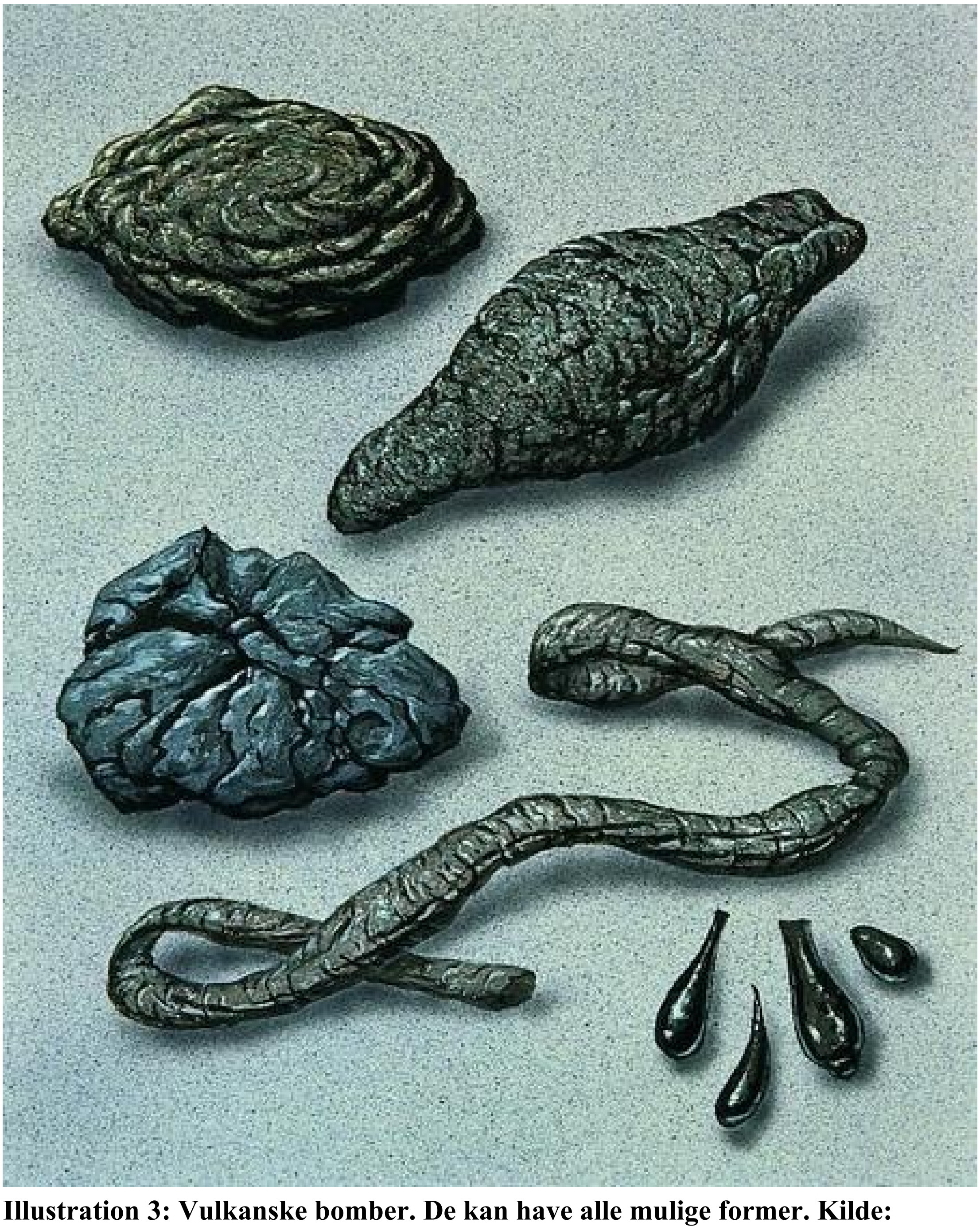


Illustration 3: Vulkanske bomber. De kan have alle mulige former. Kilde: Denstoredanske.dk.

Appendix

Start af programmeringsmodulet

Når nSpire startes op, skal man tilføje et programmeringsvindue, hvor selve programmet kaldes.

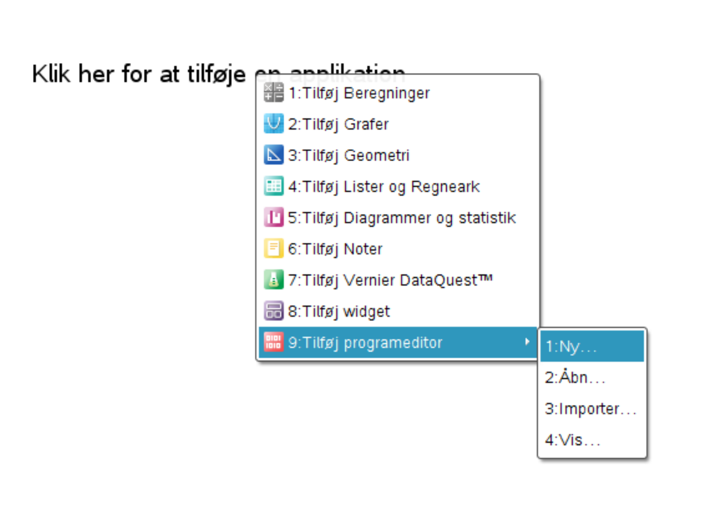


Illustration 4

Når programmet er skrevet og testet, kan man køre programmet fra et *almindeligt beregningsvindue*, og så kan grafer mv bagefter tegnes på normal vis.

Eksempler på kode

Programkode kan nemt blive totalt rodet, så det kan være en god ide, at krydre programmet med kommentarer, der forklarer, hvad der foregår. Det gør man ved at trykke på *Handlinger-Indsæt kommentar*. (Genvejstast: *Alt+1+8.)* Derved fremkommer tegnet ©, og kommentaren kan skrives efter tegnet.

Skal man gentage en beregning mange gange med en løbende variabel, er en løkke en god måde, at gøre det på. Herunder er der et eksempel på en beregning med en indledende kommentar.

```
© Herunder beregnes en serie funktionsværdier.
j:=1
For i,1,10,0.5
 x[j]:=i
 f[j]:=5·i+10
 j:=j+1
EndFor
```

Øvelse

1. Åbn et programvindue og giv det navnet *eksempel*.
2. Prøv at taste programlinierne ovenfor ind i programmet.
3. Højreklik i vinduet og vælg *kontroller syntaks og gem.* Melder maskinen om fejl, så sammenlign programteksten nøje med det, du har skrevet.
4. Åbn et beregningsvindue på en ny side og kør programmet ved at skrive *eksempel()*. (Uden punktummet.)

Ovenfor beregnes f(x)=5x+10 for x ∈ {1; 1,5; 2; 2,5 ... 10}. Alle værdierne gemmes i variablerne x og f. Bemærk, at x og f er vektorer, dvs. de indeholder mange værdier.

Øvelse

1. Åbn programmet eksempel, som du lavede i forrige øvelse.
2. Ret programmets første linie, så der nu står ***Define*** *eksempel(a,b)*.
3. Tilføj linien *Delvar x,f*. Dermed slettes hukommelse med gamle værdier. (Det svarer til, at du tager et nyt stykke papir, så der ikke står en masse gamle ting på dit ark.)
4. Ret beregningen af f[x] til *f[j]:=a·i+b*
5. Højreklik og kontroller syntaks og gem.
6. Prøv nu i et beregningsvindue at skrive *eksempel(1,2).* Når programmet er færdigt, så skriv *f* og tryk på Enter. Nu kan du se funktionsværdierne.
7. Prøv at indtaste kommandoen *eksempel(2,10)*.'
8. Skriv igen *f* og tryk på Enter. Kan du forstå, hvad maskinen regner ud?

Til beregninger kan man bruge nSpires indbyggede funktioner, men bemærk, at hvis man bruger solve()-funktionen, så gemmes resultatet som en tekst, og man kan altså ikke regne videre med resultatet. Derfor er nsolve()-funktionen mere egnet til at løse en ligning, hvis man får behov for det.

Hvis man vil standse en kørsel, når en variabel får en bestemt værdi, kan man lave en forespørgsel på værdien af variablen og så bede programmet vælge en eller flere handlinger.

For eksempel kan man skrive følgende kode

```
If y<0 Then
 Goto Afslutning
EndIf
..
..
Lbl Afslutning
DelVar x,y
```

Ovenfor ønsker programmøren at afslutte et program, hvis y-variablen bliver negativ. Hvis y ikke er negativ fortsætter programmet efter EndIf-sætningen.

Lbl er en forkortelse for *Label* og DelVar sletter nogle midlertidige variable fra hukommelsen.

Man kan læse mere om programmering på følgende link:
http://tibasic.wikia.com/wiki/TI-Basic_Nspire_Programming