

Tykillä ampuminen 2

Mallinnettaessa heittoliikettä, kuten esimerkiksi tykillä ampumista, keskeisinä vaikuttavina tekijöinä ovat painovoima sekä ilmanvastuksen aiheuttama nopeudelle vastakkaissuuntainen voima.

Painovoima on $\overline{G} = m\overline{g}$, missä $\overline{g}$ on maan vetovoiman aiheuttama kiihtyvyys ja m heitettävän kappaleen massa. Ilmanvastusvoima on verrannollinen nopeuden neliöön, ja se voidaan mallintaa lausekkeella

$$\overline{F} = -bv^2\hat{v} = -bv\overline{v},$$

missä b on ilmanvastuskerroin, v skalaarinen nopeus, $\overline{v}$ nopeus vektorina ja $\hat{v}$ nopeuden suuntainen yksikkövektori.

Yleinen liikeyhtälö $\overline{F} = m\overline{a}$ saa tällöin muodon

$$m \frac{\partial^2 \overline{r}}{\partial t^2} = m\overline{g} - b \left| \frac{\partial \overline{r}}{\partial t} \right| \frac{\partial \overline{r}}{\partial t},$$

missä on merkitty $\overline{a} = \frac{\partial^2 \overline{r}}{\partial t^2}$ sekä $\overline{v} = \frac{\partial \overline{r}}{\partial t}$.

Tarkastellaan tilannetta, jossa tykillä ammutaan poispäin liikkuvaan maaliin. Mikä on oikea ampumiskulma, kun 25 metrin korkeudella sijaitsevalta tykkilavetiltä ammutaan 10 kg kranaatti nopeudella 1000 m/s kohti maalia, jonka etäisyys ampumishetkellä on tasan 5 km? Maali liikkuu 15 m/s tykistä poispäin veden pintaa pitkin, ja ilmanvastuskerroin kranaatille on $b = 10^{(-4)}$.

Jaetaan yllä esitetty toisen kertaluvun vektorimuotoinen differentiaaliyhtälö kahteen eri osayhtälöön, z-suuntaan ja x-suuntaan, ja ratkaistaan yhtälöt. Laskujen aluksi on syytä hävittää mahdollisista aiemmista laskuista jääneet muuttujat.

```
> restart;
```

Määritellään differentiaaliyhtälöryhmä ja sen tuntemattomat muuttujat.

```
> ryhmä:= m*diff(x(t), t$2)=-b*diff(x(t), t)*sqrt(diff(x(t),  
t)^2+diff(z(t), t)^2), m*diff(z(t), t$2)=-m*g-b*diff(z(t),  
t)*sqrt(diff(x(t), t)^2+diff(z(t), t)^2);
```

$$\text{ryhma} := m \left(\frac{d^2}{dt^2} x(t) \right) = -b \left(\frac{d}{dt} x(t) \right) \sqrt{\left(\frac{d}{dt} x(t) \right)^2 + \left(\frac{d}{dt} z(t) \right)^2},$$

$$m \left(\frac{d^2}{dt^2} z(t) \right) = -m g - b \left(\frac{d}{dt} z(t) \right) \sqrt{\left(\frac{d}{dt} x(t) \right)^2 + \left(\frac{d}{dt} z(t) \right)^2}$$

> **muuttujat := x(t), z(t);**

muuttujat := x(t), z(t)

Määritellään vakiot ja alkuehto. Ammuksen lähtökulma olkoon θ . Sisällytetään ehtoon sekä ammuksen alkunopeus jaettuna nopeuskomponentteihin että lähtökorkeus.

> **g := 9.81: b := 10^(-4): m := 10: v0 := 1000:**

> **alkuehto := x(0)=0, z(0)=25, D(x)(0)=v0*cos(theta),
D(z)(0)=v0*sin(theta);**

alkuehto := x(0) = 0, z(0) = 25, D(x)(0) = 1000 cos(θ), D(z)(0) = 1000 sin(θ)

Differentiaaliyhtälöryhmä ei ole ratkaistavissa alkeisfunktioiden avulla, joten ryhmää ei voi ratkaista *Maplen* avulla θ :n funktiona. Suljetussa muodossa olevan ratkaisun sijaan joudutaan turvautumaan arvaukseen. Arvataan jokin kulma (radiaaneissa), ja katsotaan, miten hyvin laukaus osui.

> **theta := 0.03;**

$\theta := 0.03$

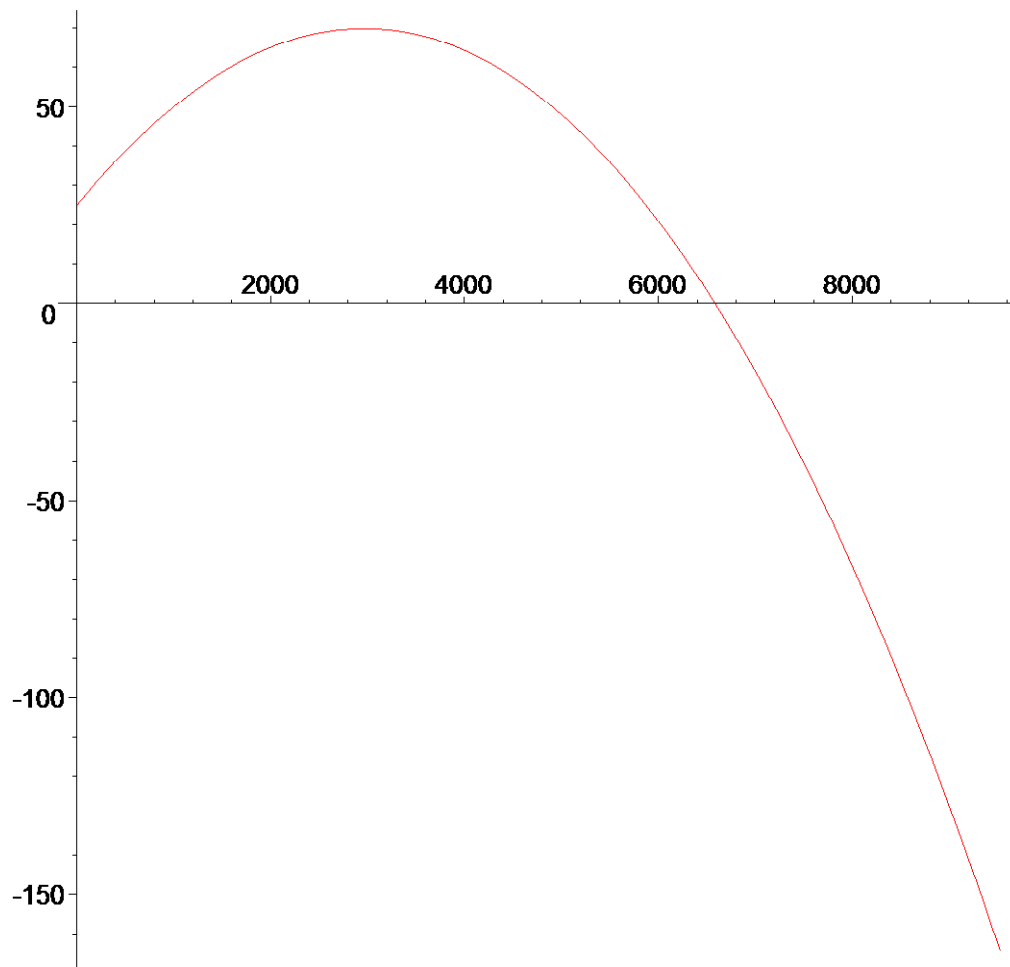
> **rtk1 := dsolve({ryhma, alkuehto}, {muuttujat}, numeric,
startinit=true, output=listprocedure);**

*rtk1 := [t = (proc(t) ... end proc), x(t) = (proc(t) ... end proc),
 $\frac{d}{dt} x(t) = (\text{proc}(t) \dots \text{end proc})$, z(t) = (proc(t) ... end proc), $\frac{d}{dt} z(t) = (\text{proc}(t) \dots \text{end proc})$
]*

Poimitaan ratkaisusta x- ja z-suuntia kuvaavat komponentit listaksi ja piirretään kuvio.

> **lontorata1 := subs(rtk1, [x(t), z(t)]):**

> **plot([lontorata1[], 0..10]);**



Katsotaan miten lähelle laukaus osui. Huomaa, että ratkaisussa on huomioitu myös kohteen liike.

```
> lentoaika1:= fsolve(lentorata1[2](t)=0, t=0..10);
                        lentoaika1 := 6.811463394
> osmaet1:= lentorata1[1](lentoaika1)-(5000+15*lentoaika1);
                        osmaet1 := 1484.431854
```

Ammus osuu veteen noin puolitoista kilometriä liian kauaksi. Lähtökulma oli siis liian suuri. Arvataan uudestaan tarkoituksena haarukoida oikea tulos arvausten väliin. Tämän jälkeen voidaan systemaattisesti hakea oikea kulma.

```
> theta:= 0.01;
                        θ := 0.01
> rtk2:= dsolve({ryhma,alkuehto}, {muuttujat}, numeric,
                startinit=true, output=listprocedure);
```

```

rtk2 := [ t = (proc(t) ... end proc), x(t) = (proc(t) ... end proc),
           $\frac{d}{dt}x(t) = (\text{proc}(t) \dots \text{end proc})$ , z(t) = (proc(t) ... end proc),  $\frac{d}{dt}z(t) = (\text{proc}(t) \dots \text{end proc})$ 
        ]
> lentorata2:= subs(rtk2, [x(t), z(t)]):
> lentoaika2:= fsolve(lentorata2[2](t)=0, t=0..10);
                    lentoaika2 := 3.499905031
> osumaet2:= lentorata2[1](lentoaika2)-(5000+15*lentoaika2);
                    osumaet2 := -1612.618768

```

Nyt laukaus jäi liian lyhyeksi. Saimme kuitenkin haarukoitua osumisvälin. Löytääksemme tarkan ampumiskulman, teemme organisoidun kokeilun.

Tehdään pieni **for**-silmukka, joka tulostaa osumapaikan 21 laskupisteessä 0.001 radiaanin välein. Laskentamenettely hieman poikkeaa edellä olevista, koska Maplen uudemmissa versioissa fsolve-funktio ei kaikissa tapauksissa toimi oikein. Tämän takia käytössä on seuraava erikseen ohjelmoitu proseduri nollakohta, jossa funktion nollakohdan hakemiseen käytetään *regula falsi* -menetelmää.

```

> nollakohta:=proc(fkt,x1,x2)
    local a,b,c,fa,fb:
    a:=x1: b:=x2: fa:=fkt(a): fb:=fkt(b):
    if fa*fb>0 then print(FAIL); return end if:
    c:=(a+b)/2:
    while abs(a-c)>0.00000001 do
        a:=c: fa:=fkt(a):
        c:=a-(a-b)/(fa-fb)*fa:
    end do:
    c;
end proc:
> i:= 0:
for theta from 0.01 by 0.001 to 0.03 do
    rtk[i]:= dsolve({ryhma, alkuehto}, {muuttujat}, numeric,
startinit=true, output=listprocedure);
    lentorata:= subs(rtk[i], [x(t), z(t)]);
    lentoaika:= nollakohta(lentorata[2],0.,10.);
    osumataul[i]:= lentorata[1](lentoaika)-(5000+15*lentoaika);
    kulmataul[i]:= theta;
    i:= i+1;
od:

```

```

kulmat:= [seq(kulmataul[k], k=0..20)]:
etaisyydet:= [seq(osumataul[k],k=0..20)];

```

```

etaisyydet := [-1612.618760, -1475.385567, -1335.351170, -1192.699248, -1047.610328,
-900.2601212, -750.8182643, -599.4473395, -446.3022613, -291.5298303, -135.2685688,
22.35134319, 181.2079169, 341.1872060, 502.1829512, 664.0962402, 826.8351781,
990.3144468, 1154.454953, 1319.183384, 1484.431852]

```

Etsitään lähintä osumaa vastaava indeksi ja tämän avulla ampumiskulma, osumaetaisyys ja lentoaika.

```

> member(min(map(x->abs(x), etaisyydet)[]), etaisyydet, 'ind'):
lahin:=[kulmat[ind], etaisyydet[ind]];
lahin := [0.021, 22.35134319]

```

Sama uudestaan uudella kapeammalla haarukalla ja 0.0001 radiaanin askelvälillä.

```

> i:= 0:
for theta from 0.020 by 0.0001 to 0.022 do
rtk[i]:= dsolve({ryhma, alkuehto}, {muuttujat}, numeric,
startinit=true, output=listprocedure);
lontorata:= subs(rtk[i], [x(t), z(t)]);
lentoaika:= nollakohta(lontorata[2],0.,10.);
osumataul[i]:= lontorata[1](lentoaika)-(5000+15*lentoaika);
kulmataul[i]:= theta;
i:= i+1;
od:

```

```

kulmat:= [seq(kulmataul[k], k=0..20)]:
etaisyydet:= [seq(osumataul[k],k=0..20)];

```

```

etaisyydet := [-135.2685688, -119.5656401, -103.8492586, -88.11954534, -72.37662850,
-56.62062526, -40.85166864, -25.06987092, -9.27535822, 6.53175518, 22.35134319,
38.18329350, 54.02748394, 69.88380224, 85.75213014, 101.6323504, 117.5243548,
133.4280249, 149.3432505, 165.2699182, 181.2079169]

```

```

> member(min(map(x->abs(x), etaisyydet)[]), etaisyydet, 'ind'):
lahin:=[kulmat[ind], etaisyydet[ind]];
lahin := [0.0209, 6.53175518]

```

Osuma on siis runsaan kuuden metrin päässä ja vastaava ampumiskulma on asteissa

```

> convert(lahin[1], degrees):
evalf(%);

```

1.197481792 *degrees*

>

Tehtäviä

Piirrä lähimmäksi osuvan ammuksen lentorata. Määritä radan lakikorkeus ja ammuksen lentoaika.

Suurena ilmanvastuskerrointa, ja tutki, miten se vaikuttaa radan muotoon.

Linkit

tykillä ampuminen huomioimatta ilmanvastusta (tykki1.mws)

JP & SKK & MS 12.07.2001 & 08.07.2005