

Laboratorijske vaje Numerične metode

5. Vaja

B. Jurčič Zlobec¹, A. Perne¹

¹Univerza v Ljubljani,
Fakulteta za Elektrotehniko
1000 Ljubljana, Tržaška 25, Slovenija

Numerične metode FE, Ljubljana, 6. november 2012

Elektroskop

Na dveh vrvicah dolžine $\ell = 12\text{cm}$ visita kroglici enakih mas $m = 1\text{g}$. Glej sliko 1. Kroglici sta prevodni in naelektreni s $Q = 2.5 \cdot 10^{-8}\text{As}$ naboja. Kolikšen je kot (α) med vrvicama?

Influenčna konstanta $\epsilon_0 = 8.854184817 \cdot 10^{-12}\text{ F/m}$
in težni pospešek $g = 9.80665\text{ m/s}^2$.

Sila teže in električna sila:

$$F_e = \frac{Q^2}{4\pi\epsilon_0 d^2}, \quad F_g = mg, \quad d = 2\ell \sin \frac{\alpha}{2}. \quad (1)$$

V ravnovesnem stanju je

$$\sum F_x = 0 \rightarrow \frac{F_e}{F_g} = \tan \frac{\alpha}{2}. \quad (2)$$

Enačba

Pišimo:

$$u = \tan \frac{\alpha}{2}, \quad k = \frac{Q^2}{16\pi\epsilon_0 mg \ell^2}, \quad (3)$$

izraze (1) vstavimo v enačbo (2) izrazimo s k in u iz (3) in dobimo enačbo za u :

$$u^3 - ku^2 - k = 0.$$

Program inicializacija

Definicija konstant.

```
epsilon0=8.854184817*1e-12;  
g=9.80665;  
m=1e-3;  
l=0.12;  
Q=2.5*1e-8;  
k=Q^2/(16*pi*epsilon0*m*g*l^2);
```

Definicija funkcije, za klic z ukazom `eval`.

```
fs='u.^3-k*u.^2-k';  
dfs='3*u.^2-2*k*u';
```

Program inicializacija

Definicija konstant.

```
epsilon0=8.854184817*1e-12;  
g=9.80665;  
m=1e-3;  
l=0.12;  
Q=2.5*1e-8;  
k=Q^2/(16*pi*epsilon0*m*g*l^2);
```

Definicija funkcije, za klic z ukazom `eval`.

```
fs='u.^3-k*u.^2-k';  
dfs='3*u.^2-2*k*u';
```

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

```
iter=5, alfa= 24.64097133 stopinj
```

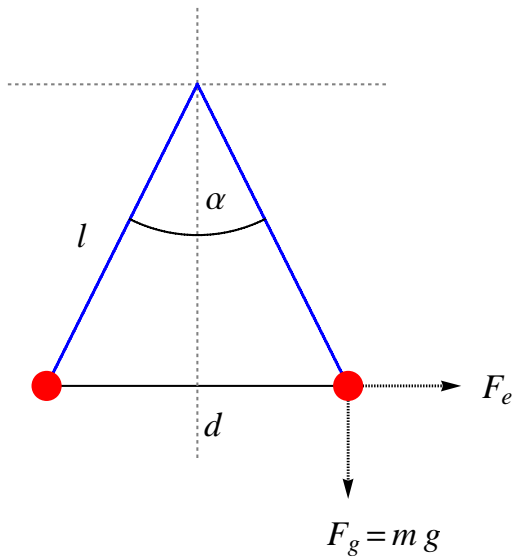
Program

Reševali bomo s pomočjo Newtonove metode.

```
alpha=20*pi/180;  
eps=1e-8;  
u=tan(alpha/2);  
for i=1:100,  
    u1=u-eval(fs)/eval(dfs);  
    if abs(u1-u)<eps, break, end;  
    u=u1;  
end;  
printf('iter=%0d, alfa= %0.8f stopinj\n',...  
i,2*atan(u)/pi*180);
```

iter=5, alfa= 24.64097133 stopinj

Slika 1.



Poševni met z linearnim uporom

Pri danih začetnih pogojih izračunaj domet, to je absciso točke, kjer se pot izstrelka sreča z osjo x .

Podatki:

Sila upora je premosorazmerna hitrosti $\beta \vec{v}$.

Začetni položaj je $(0, 0)$, začetna hitrost je $\vec{v}_0 = (v_x, v_y)$.

Gravitacijski pospešek je g .

Ostali podatki:

Začetna hitrost je $\vec{v}_0 = (2, 4)$, $g = 9.80665$ in $\beta = 2$.

Enačbe

- Newtonov zakon:

$$\vec{F} = m\vec{a} = \vec{F}_g + \vec{F}_u = m\vec{g} - m\beta\vec{v}, \rightarrow \vec{a} + \beta\vec{v} = \vec{g}.$$

- Sistem diferencialnih enačb:

$$\ddot{x}(t) + \beta\dot{x}(t) = 0, \quad \ddot{y}(t) + \beta\dot{y}(t) = -g.$$

- Začetni pogoji: $x(0) = y(0) = 0, \dot{x}(0) = v_x, \dot{y}(0) = v_y.$

- Rešitev:

$$x(t) = \frac{v_x}{\beta} \left(1 - e^{-\beta t} \right),$$

$$y(t) = \frac{1}{\beta^2} \left(g - \beta g t + \beta v_y - e^{-\beta t} (g + \beta v_y) \right).$$

Enačbe

- Newtonov zakon:

$$\vec{F} = m\vec{a} = \vec{F}_g + \vec{F}_u = m\vec{g} - m\beta\vec{v}, \rightarrow \vec{a} + \beta\vec{v} = \vec{g}.$$

- Sistem diferencialnih enačb:

$$\ddot{x}(t) + \beta\dot{x}(t) = 0, \quad \ddot{y}(t) + \beta\dot{y}(t) = -g.$$

- Začetni pogoji: $x(0) = y(0) = 0, \dot{x}(0) = v_x, \dot{y}(0) = v_y.$

- Rešitev:

$$x(t) = \frac{v_x}{\beta} \left(1 - e^{-\beta t} \right),$$

$$y(t) = \frac{1}{\beta^2} \left(g - \beta g t + \beta v_y - e^{-\beta t} (g + \beta v_y) \right).$$

Enačbe

- Newtonov zakon:

$$\vec{F} = m\vec{a} = \vec{F}_g + \vec{F}_u = m\vec{g} - m\beta\vec{v}, \rightarrow \vec{a} + \beta\vec{v} = \vec{g}.$$

- Sistem diferencialnih enačb:

$$\ddot{x}(t) + \beta\dot{x}(t) = 0, \quad \ddot{y}(t) + \beta\dot{y}(t) = -g.$$

- Začetni pogoji: $x(0) = y(0) = 0$, $\dot{x}(0) = v_x$, $\dot{y}(0) = v_y$.

- Rešitev:

$$x(t) = \frac{v_x}{\beta} \left(1 - e^{-\beta t} \right),$$

$$y(t) = \frac{1}{\beta^2} \left(g - \beta g t + \beta v_y - e^{-\beta t} (g + \beta v_y) \right).$$

Enačbe

- Newtonov zakon:

$$\vec{F} = m\vec{a} = \vec{F}_g + \vec{F}_u = m\vec{g} - m\beta\vec{v}, \rightarrow \vec{a} + \beta\vec{v} = \vec{g}.$$

- Sistem diferencialnih enačb:

$$\ddot{x}(t) + \beta\dot{x}(t) = 0, \quad \ddot{y}(t) + \beta\dot{y}(t) = -g.$$

- Začetni pogoji: $x(0) = y(0) = 0, \dot{x}(0) = v_x, \dot{y}(0) = v_y.$

- Rešitev:

$$x(t) = \frac{v_x}{\beta} \left(1 - e^{-\beta t} \right),$$

$$y(t) = \frac{1}{\beta^2} \left(g - \beta g t + \beta v_y - e^{-\beta t} (g + \beta v_y) \right).$$

Enačbe

- Newtonov zakon:

$$\vec{F} = m\vec{a} = \vec{F}_g + \vec{F}_u = m\vec{g} - m\beta\vec{v}, \rightarrow \vec{a} + \beta\vec{v} = \vec{g}.$$

- Sistem diferencialnih enačb:

$$\ddot{x}(t) + \beta\dot{x}(t) = 0, \quad \ddot{y}(t) + \beta\dot{y}(t) = -g.$$

- Začetni pogoji: $x(0) = y(0) = 0, \dot{x}(0) = v_x, \dot{y}(0) = v_y.$

- Rešitev:

$$x(t) = \frac{v_x}{\beta} \left(1 - e^{-\beta t} \right),$$

$$y(t) = \frac{1}{\beta^2} \left(g - \beta g t + \beta v_y - e^{-\beta t} (g + \beta v_y) \right).$$

Reševanje

Rešimo enačbo $y(t_0) = 0$ s pomočjo sekantne metode.

Čas t_0 je čas, ko izstrellek pade na tla.

Domet je enak $d = x(t_0)$.

Začetna približka določimo s poizkušanjem.

```
v=[2,4]; beta=2; g=9.80665;  
x='v(1)/beta*(1-exp(-beta*t))';  
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\  
(g+beta*v(2)))';  
t=[0.5,0.7]; eval(y)
```

```
ans = 0.362325 -0.078431
```

Reševanje

Rešimo enačbo $y(t_0) = 0$ s pomočjo sekantne metode.

Čas t_0 je čas, ko izstrellek pade na tla.

Domet je enak $d = x(t_0)$.

Začetna približka določimo s poizkušanjem.

```
v=[2,4]; beta=2; g=9.80665;
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
t=[0.5,0.7]; eval(y)
```

```
ans = 0.362325 -0.078431
```

Reševanje

Rešimo enačbo $y(t_0) = 0$ s pomočjo sekantne metode.

Čas t_0 je čas, ko izstrellek pade na tla.

Domet je enak $d = x(t_0)$.

Začetna približka določimo s poizkušanjem.

```
v=[2,4]; beta=2; g=9.80665;  
x='v(1)/beta*(1-exp(-beta*t))';  
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\  
(g+beta*v(2)))';  
t=[0.5,0.7]; eval(y)
```

```
ans = 0.362325 -0.078431
```


Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

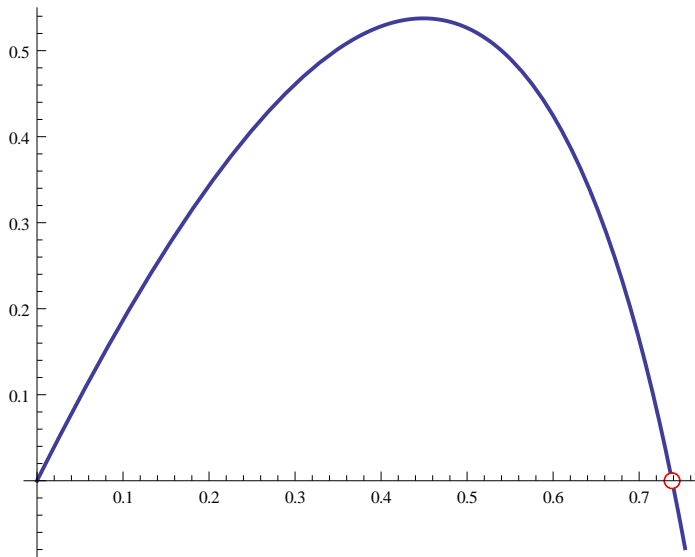
Čas padca projektila in domet

Sekantna metoda, računanje dometa.

```
eps=1e-8;  
for i=1:100,  
    ff=eval(y);  
    tt=t(2)-ff(2)*diff(t)/diff(ff);  
    t(1)=t(2); t(2)=tt;  
    if abs(diff(t))<eps, break, end;  
end;  
domet=mean(eval(x));  
printf('cas=%0.8f, domet=%0.8f\n',mean(t),domet);
```

cas=0.67030559, domet=0.73831432

Slika 2.



Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Strel v tarčo

Streljamo v tarčo, ki je postavljena v točki $(4, 0)$.

Streljamo pod kotom 45° (domet je največji).

Kolikšna mora biti začetna hitrost, da izstrelek doseže tarčo.

- Zapišimo funkcijsko datoteko `domet=strel(hitrost,t)`.
- Funkcija vrne domet izstrelka pri dani začetni hitrosti.
- Komponenti hitrosti sta enaki (kot je 45°).
- Količnik upora $\beta = 2$.
- Oceno časa za sekantno metodo bomo izbrali $t=[0.5, 1]$.
- Zapišimo program, ki bo s pomočjo sekantne metode, ugibal začetno hitrost za predpisan domet.

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```


Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program prvi del

Funkcija bum, datoteka bum.m

```
function domet=strel(hitrost,t)
x='v(1)/beta*(1-exp(-beta*t))';
y='1/beta^2*(g-beta*g*t+beta*v(2)-exp(-beta*t)*\
(g+beta*v(2)))';
v=hitrost*[1,1]/sqrt(2); beta=2; g=9.80665;
eps=1e-8;
for i=1:100,
    ff=eval(y);
    tt=t(2)-ff(2)*diff(t)/diff(ff);
    t(1)=t(2); t(2)=tt;
    if abs(diff(t))<eps, break, end;
end;
domet=mean(eval(x));
```

Program drugi del

Strel v tarčo

```
tarca=4; t=[0.5,1];  
hitrost=[4,4.5]; eps=1e-8;  
for i=1:100  
    kiks=[strel(hitrost(1),t)-tarca,...  
        strel(hitrost(2),t)-tarca];  
    s=hitrost(2)-kiks(2)*diff(hitrost)/diff(kiks);  
    hitrost(1)=hitrost(2); hitrost(2)=s;  
    if abs(diff(hitrost))<eps, break, end;  
end;  
printf('hitrost=%0.8f\n',mean(hitrost));
```

```
hitrost=12.26802407
```

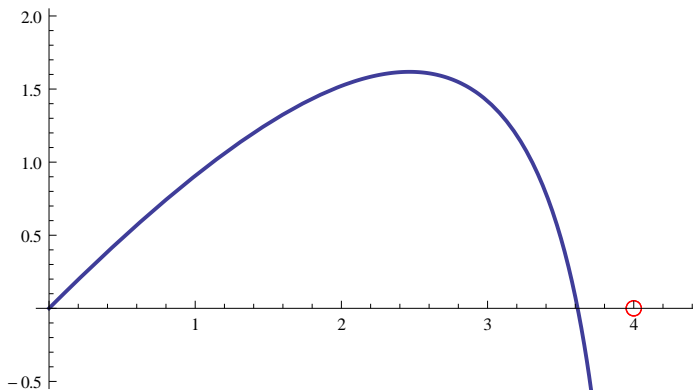
Program drugi del

Strel v tarčo

```
tarca=4; t=[0.5,1];  
hitrost=[4,4.5]; eps=1e-8;  
for i=1:100  
    kiks=[strel(hitrost(1),t)-tarca,...  
        strel(hitrost(2),t)-tarca];  
    s=hitrost(2)-kiks(2)*diff(hitrost)/diff(kiks);  
    hitrost(1)=hitrost(2); hitrost(2)=s;  
    if abs(diff(hitrost))<eps, break, end;  
end;  
printf('hitrost=%0.8f\n',mean(hitrost));
```

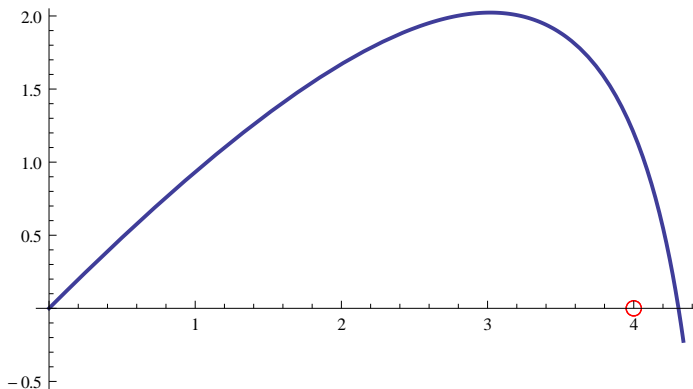
hitrost=12.26802407

Slika 3.



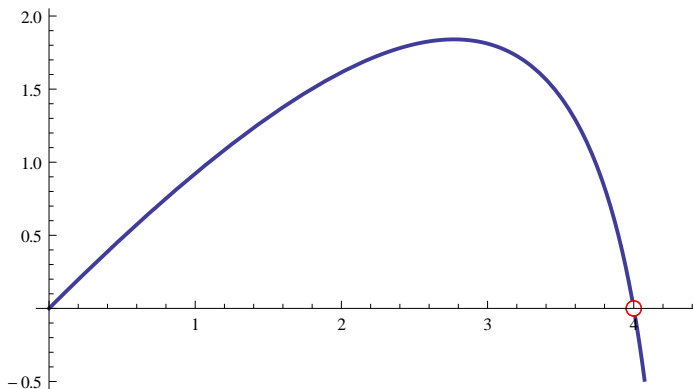
Slika: Strel preblizu

Slika 3.



Slika: Strel predaleč

Slika 3.



Slika: Strel točno v sredino tarče