Aalto-yliopisto, Matematiikan ja Systeemianalyysin laitos **Apiola/Kuortti**

Mat-1.C Matemaattiset ohjelmistot

Harjoitus 1 12-14.3.2012

LT – Luentotehtävä, tehdään yhdessä luennolla.
HT – Harjoitustehtävä, käydään keskustellen läpi ma 19.3. rastilista.
DokuT – Kirjallinen työ, m-tiedosto lähetetään Juhalle ma 19.3. klo 14.00 mennessä. Jaottelu `%%`-merkein, saa selostaa ja kommentoida muutamalla sanalla. Pääasia, että testattu. Saa laittaa kommenttia työmäärästä, ongelmista ym. Saat myös lähettää publish-pdf-tiedoston, mutta Juha ei sitä vaadi.

Sovitaan ke HT-tehtävien minimimäärä.

Doku-tehtäviä on 4, joista tulisi valita 2. Ne ovat kaikki lyhyitä ja tarkasti neuvottuja.
(Jos haluat tehdä enemmän, ilmoita rastilistalla, mutta älä lähetä kahta enempää Juhalle (ettei vaadi palkankorotusta).)

Keskiviikon istunnon luento-osuudessa keskitytään niihin asioihin, joita näissä tehtävissä tarvitaan.

Ylivoimaisen esteen tapauksessa voidaan neuvotella erikoisjärjestelystä kaikissa kysymyksissä.

LT **1.** Olkoon `z = [0 -1 2 4 -2 1 5 3]`, ja `J = [5 2 1 6 3 8 4 7]`.
Mitä syntyy seuraavilla Matlab-komennoilla (sijoitetaan tilan säästämiseksi useita samalle riville.)

```
>> x = z', A = x*x', s = x'*x, w = x*J,
>> length(x), length(z)
>> size(A), size(x), size(z), size(s)
```

Vihje: Suorita doc length, doc size, tai etsi Matlabin Help index:n avulla (lisä)tietoa komennoista.

LT **2.** Määrittele vektorit

```
x = [1 2 3 4 5]
y = [0 2 4 6]
z = [-4 -2 0 2 4 ]
```

Kokeile seuraavia laskutoimituksia/komentoja (kirjoita mieluummin omille riveilleen), ja selvitä virheelliset.

```
x.*z, x*z' ,x*z
x.^2, x^2
s1=sqrt(x*x')
s2=sqrt(sum(x.^2))
s3=norm(x)
[s1 s2 s3] % helppo verrata
help norm
```

HT **3.** Määrittele matriisit

$$u = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 6 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 6 \\ 1 & 4 & 9 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 4 & 5 & 6 \\ 2 & 1 & 0 & -1 \\ 5 & 4 & 2 & 0 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

Selvitä (ilman Matlabia), mitkä seuraavista laskutoimituksista on määritelty, ja kerro sanallisesti, mitä ne tekevät. Tarkista MATLAB:lla.

```
A*C   C*A    C^2    C.^2   A^2    A.^2
```

HT **4.** Kirjoita MATLAB-skripti, joka laskee ja piirtää seuraavat funktiot:

a) $y = 5\cos(3\pi x)$. Laske arvo 101:ssä tasavälisessä pisteessä välillä $0 \leq x \leq 1$.

b) $y = \frac{1}{1+x^2}$ välillä $-5 \leq x \leq 5$.

c) $y = \frac{\sin(7x)-\sin(5x)}{\cos(7x)+\cos(5x)}$. Laske arvo 200 tasavälisessä pisteessä välillä $-\frac{\pi}{2} \leq x \leq \frac{\pi}{2}$. Käytä `axis` komentoa asettaaksesi näytettävät akselit väleille $-2 \leq x \leq 2$ ja $-10 \leq y \leq 10$.

Vihje: Jako- ja kertolaskujen tapauksessa ole tarkkana: haluatko matriisioperaation vai alkioittaisen operaation? Alkioittaiset operaatiot erotetaan matriisioperaatioista operaattorin eteen sijoitettavalla pisteellä. Esimerkiksi `.*` on alkioittainen kertolasku, `*` matriisien kertolasku.

Trigonometriset funktiot toimivat MATLABissa alkioittain, ja löytyvät loogisilla nimillä. (`cos, acos, sin` jne.)

Tasavälisiä pistejoukkoja luodaan komennolla `linspace`, tai vaihtoehtoisesti MATLABin kaksoispiste-notaatiolla. Tutustu kummankin dokumentaatioon, ja päätä kumpaa kannattaa tässä tilanteessa käyttää.

HT **5.** Esitä yhden rivin Matlab-komento, jolla saat selville vektorin tai matriisin niiden alkioiden lukumäärän, jotka ovat > 5.

Testaa ainakin näille:

a) `A=1:10`

b) `B=[1 2 3;4 5 6;7 8 9]`

c) `C=10*rand(6,6)`

d) `D=ones(4,4)`

HT **6. HT**

a) Ratkaise lineaarinen yhtälöryhmä ja tarkista tulos kertolaskulla.

$$\begin{cases} 4x - 5y = 11 \\ 2x + y = 9 \end{cases}$$

Vihje: Matlab: "Matriisijako": `A\b`
Maple: `LinearSolve` (ei tarvitse ekalla viikolla)

b) Tiedetään, että Celsius-asteiden ja Fahrenheit-asteiden välillä on lineaarinen yhteys:

$$C = aF + b.$$

Lisäksi tiedetään, että vesi jäätyy 32 F:ssa ja -40 on sama kummassakin asteikossa. Johda kaava. Tarkoitus on kirjoittaa kertoimien a ja b määräämiseksi lineaarinen yhtälösysteemi, joka ratkaistaan Matlab:n takakenolla (\).

c) Muodosta matriisi, jonka 1. sarake on C-asteet -50:sta 5:n asteen välein 100 :aan ja toinen sisältää vastaavat F-asteet.

Vihje: Tarkan rationaalilukukaavan saat komentamalla `format rat`. Tee m-tiedosto kommentteineen.
Huomaa, että taulukkoa ei ole mukavaa katsoa koknaisuutena, esim. 10 ekaa riviä näet näin: `taulukko(1:10,:)` (eikö vain?).
Hivelevää on myös mennä "Workspace-ikkunaan" ja kaksoisklikata taulukko-ikonia.

Kokeile sen ajamista myös pdf:ksi `publish(Fahrenheit,pdf)`-komennolla (jos skripti on `Fahrenheit.m`), kunhan ensin testailet sen kuntoon.

HT **7.** Kirjoita skripti, jolla vertaat `inv(A)*b`:n ja `A\b`:n tehokkuus- ja tarkkuuseroja.

Tee välillä $[0, 10]$ olevista satunnaisista kokonaisluvuista koostuva matriisi $n \times n$-matriisi A, ja muodosta sen rivisummista koostuva sarakevektori b. Olkoon `u=ones(n,1)` .

Mieti, miksi u on yhtälön $Ax = b$ ratkaisu. Vahvista tämä Matlab-laskulla:

```
%%
n=5
A=floor(10*rand(n))
b=sum(...)
b=b'
u=ones(n,1)
[A*u b]
x=A\b
%%
```

Jatka nyt skriptiäsi (copy/paste) ja aloita $n = 200$. Muista päättää puolipisteeseen (;) [muuten kärsit].

Suorita ajanmittausta varten:

```
tic,x=A\b;toc
tic,y=inv(A)*b;toc
```

Vertaa suoritusaikoja.

Virheiden vertaamiseksi laske $\max_i |x_i - u_i|$ ja $\max_i |y_i - u_i|$

Voit myös käyttää `norm`-funktiota (`help norm`)

Kokeile vielä arvoja $n = 500, n = 1000, n = 10000$. (Ja muista puolipisteet!!) Kirjoita muutaman virkkeen yhteenveto skriptiisi ja aja vielä pdf:ksi komennolla `publish('tamaskripti','pdf')`.

Hyväksytään myös pelkkä m-tiedosto, joka on testattu toimivaksi.

LT **8.** Tarkastellaan lineaarista 4×6 systeemiä $Ax = c$,

missä $A = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 2 \\ 1 & 1 & 2 & 4 & 1 & 2 \\ 0 & 2 & 1 & 4 & 3 & 1 \\ 1 & -3 & 0 & -4 & -2 & 1 \end{bmatrix}$ ja $c = \begin{bmatrix} 0 \\ 3 \\ 2 \\ 0 \end{bmatrix}$

Tee liitännäismatriisi Ac, jossa c-vektori on liitetty A:n perään. Suorita komento `[R,jb]=rref(Ac)`. Poimi sen avulla 4×4 osamatriisi, joka on yksikkömatriisi. Päättele rangi ja tarkista `rank`-komennolla A:n ja Ac:n rangit.

Lopuksi voit huvitella `rrefmovie`;n parissa.

HT **9.** Monikulmio voidaan esittää 2-rivisenä matriisina, jonka sarakkeet edustavat koordinaattipisteitä. Esim. Kolmio, jonka kärjet ovat pisteissä $(0,0), (1,1), (1,-1)$ voitaisiin esittää matriisina

$$T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}.$$

```
>> plot(T(1,:),T(2,:))
```

piirtää kärkipisteet ja yhdistää ne janoilla tässä järjestyksessä.

(Seuraavassa tehtävässä hiukan enemmän selitystä lineerikuvausperiaatteesta.)

Jos tähän sovelletaan lineaarikuvausta, eli kerrotaan matriisilla A, saadaan kuvan kärkipisteet $S = AT$.
Muodosta jokin/joitakin 2×2-matriiseja A. Piirrä kolmio T ja sen kuva `S` eri ruutuihin. Voit kokeilla myös rinnakkain (`subplot`) ja päällekkäin `hold on` ja myös alueen täyttöä `fill`

```
» S=A*T; plot(S(1,:),S(2,:)).
```

Jos haluaisit iteroida, voisit tehdä komentoikkunassa ↑-tyyliin:

```
>> S=T;
» S=A*S; plot(S(1,:),S(2,:)) % ja tätä riviä ↑-toistaen.
```

DokuT **10.** **HT** Muodosta seuraava matriisi, jonka sarakkeet edustavat tason pisteitä. Piirrä pisteet yhdistysjanoineen:

```
T=[0 0 -1 6 13 12 12 3 3 6 6 0;0 9 8 15 8 9 0 0 5 5 0 0]
plot(T(1,:),T(2,:))
```

Olkoon A 2×2-matriisi. Tason piste (pystyvektori) kuvautuu matriisilla kerrottaessa pisteeksi $y = Ax$. Näin matriisi A määrää tason lineaarikuvauksen.

Piste**joukko** voidaan kuvata kokoamalla pisteet sarakkeiksi matriisiin (kuten `T` yllä) ja kertomalla se kuvausmatriisilla A. Niinpä talon nurkkapisteiden kuvat saadaan suoraan kertomalla "talomatriisi"T A:lla, ts. $S = AT$. Lineaarisuuden nojalla yhdistysjanat kuvautuvat yhdistysjanoille.

Kerro näillä matriiseilla ja piirr̈(ja kerro, mitä näet).

$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $A_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ $A_4 = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$

Miten kuvailisit vastaavia kuvauksia ? (A_4 on engl. nimeltään “horisontal shear”.)

HT **11.** Muodosta Matlab:lla polynomien $p(x) = x^3 + 2x - 1$ ja $q(x) = 2x^5 + 3$ summa. Määritä $p(x), q(x)$ ja $(p + q)(x)$ vektorin `x=[-2 -1 0 1 2]` pisteissä.

Lado nämä arvot 3-sarakkeiseksi matriisiksi PQS.
Miten voit kätevästi tarkistaa, että laskut menivät oikein?

Vihje: `polyval` arvon laskemiseen.

HT **12.**

a) Muodosta Matlab-lauseke, joka laskee kahden polynomin p ja q summan $p+q$. Oletetaan, että p:n asteluku $= n$ ja q:n asteluku $= m$ ja $m \leq n$.

b) Tee sitten yleispätevästi, tarvitsematta tietää mitään asteluvuista.

c) Kirjoita funktiotiedosto `polysum.m`, jolle annetaan argumentteina polynomivektorit ja joka palauttaa summapolynomi(vektorin). Kirjoita alkukommentit, joissa käyttötarkoitus, ohje ja käyttöesimerkki. Se siis tulee näkyviin komentamalla `>> help polysum` . Testaile!

Vihje: b):

```
np=length(p);nq=length(q);m=max(np,nq);
```

Muodosta $2\times m$ nollamatriisi, jonka 1. riville viet p:n “loppuun kiinnittäen” ja 2. riville vastaavasti q:n. (Muista `end`.) Sitten vaan muodostat sarakesummat `sum`-funktiolla, siinäpä se!

c): Ohjelmankehitys kannattaa tehdä skriptitiedostossa, ja kun se toimii, kirjoitetaan siitä funktio.

HT **13.** (a) Määritä funktion $f(x) = x^4 - 7x^3 + 11x^2 + 7x - 10$ nollakohdat, ja ääriarvot. Piirrä kuvaaja.

(b) Määritä funktion maksimi- ja mimimipisteet ja arvot "raakaa voimaa" käyttämällä soveltaen Matlabin tehokkaita vektori- ja datanpoimistyökaluja.

(c) Määritä samalla periaatteella f:n nollakohdat.

Vihje:

`polyval, polyder, roots`

(b) Tarkoitus on oikeastaan muuttaa kuvasta katsominen kvantitatiiviseksi tähän tyyliin:

```
>> x=linspace(a,b,N);
>> y=polyval(...);
>> ymax=...  % suurin y-vektorin arvo
>> maxind=find(...)  % Etsi suurimman arvon indeksi.
>>   % Indeksoi sill\"a x-vektori.
```

(c)

```
>> xpist=linspace(a,b,N); % Sopiva v\"ali ja N.
>> find(...)   % Etsi eka merkinvaihtopiste.
>>          % sill\"a v\"alill\"a on 0-kohta.
>>          % toista nuolinappaimella tai CTR-ENTER
```

Tässä tehdään välin puolittamisen sijasta välin jakaminen esim. 100:aan tai 1000:een osaan. Huomaa, että pitkillä vektoreilla operointi on Matlabilla tehokasta.

HT **14. HT**

a) Piirrä funktiot $\cos t$ ja $\sin t$ samaan kuvaan eri väreillä.

b) Piirrä toiseen kuvaan yksikköympyrä ja säännöllinen $n-$kulmio esim. arvolla $n = 10$. Järjestä sopivilla `axis`-komennoilla skaalat yhtäsuuriksi, jotta ympyrä näkyy ympyränä.

c) Piirrä yksikköympyrän kuva joillain edellä esiintyneillä lineaarikuvauksilla(tai muilla keksimilläsi).

Vihje: Uusi grafiikkaikkuna: `figure`

Muistathan ympyrän luonnollisen parametriesityksen.

Ympyrän data koostuu oikeasti säännöllisesen n-kulmion nurkkapisteistä, missä esim. $n = 100$ (`linspace`:n oletus). Ympyrän kuvan piirtäminen on siten sama homma kuin edellisissä lineaarikuvaustehtävissä.

HT **15.** Olkoon

$$g(x) = \begin{cases} 0, x \leq 0 \\ x, x > 0 \end{cases}$$

Määrittele funktio g Matlab-funktioksi (m-tiedostoon).

Vihje: Voit käyttää funktioita `zeros` ja `max` .

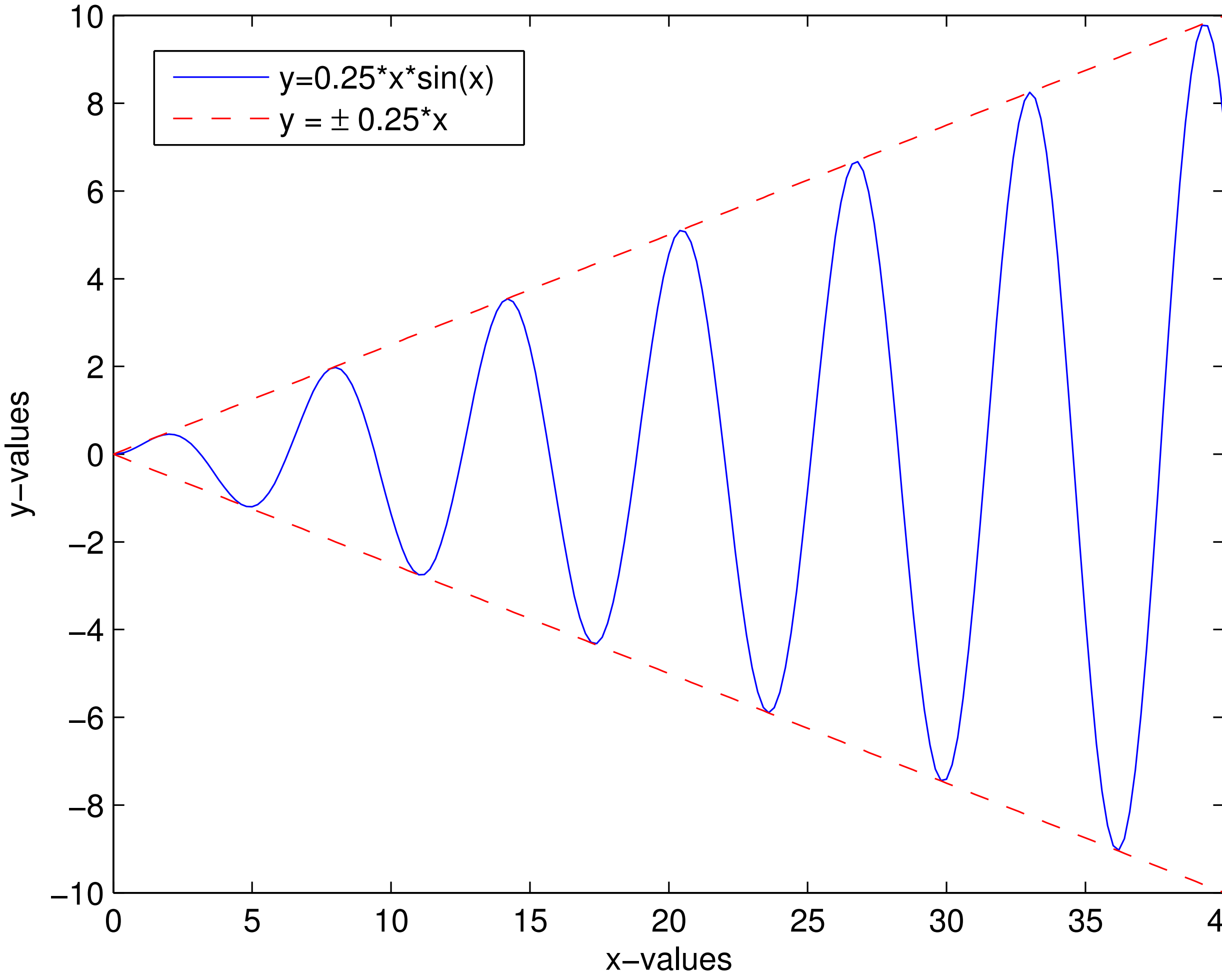
Ehkä vieläkin elegantimmin näin:
Mieti, millä saat aikaan yksikköaskelfunktion (Heavisiden funktion), joka saa negatiivisilla arvon 0 ja positiivisilla 1. (Tähän riittää 3 merkkiä.) Sillä kerrot funktion $y = x$.

HT **16.** Funktion $g(x)$ kiintopiste on piste x_0, jolle pätee $g(x_0) = x_0$. Valistuneen arvauksen kiintopisteen sijainnista voi piirtämällä kuvaajat $y = g(x)$ ja $y = x$ samaan kuvaan, ja arvioimalla käyrien leikkauspistettä graafisesti. Käyttämällä tätä tekniikkaa, arvioi funktion $g(x) = \cos(x)$ kiintopisteen sijaintia.

Vihje: Kaksi käyrää voidaan piirtää samaan kuvaan joko yhdellä `plot` käskyllä : `plot(x1,y1,x2,y2)`, tai vaihtoehtoisesti voidaan käyttää MATLABin `hold` optiota:

```
plot(x1,y1);
hold on
plot(x2,y2);
hold off
```

HT **17.** Piirrä MATLABilla alla oleva kuva.



Vihje: Selityslaatikko luodaan komennolla `legend`, akselikuvaukset komennoilla `xlabel` ja `ylabel`.

DokuT **18.** Laskemme yksikkökolmion T (virittävät pisteet $(0,0),(1,0),(0,1)$) pinta-alan tasaisesti jakautuneilla satunnaisluvuilla Monte-Carlo menetelmää mukaillen:

1. Muodosta N tasaisesti jakautunutta satunnaislukuparia (x, y) yksikköneliöön.
2. Etsi näistä satunnaispisteistä kolmion T sisälle osuvat. Havainnollista tätä piirtämällä T:n sisälle osuvat pisteet ja T:n ulkopuoliset pisteet samaan kuvaan eri väreillä.
3. Approksimoi T:n alaa laskemalla kolmion sisälle osuneiden pisteiden osuus kaikista valituista. Kokeile menetelmän tarkkuutta eri arvoilla N.

Vihje: Muodosta `rand`-funktiolla tasaisesti jakautuneita satunnaislukuja välillä $[0,1]$ vektoreihin x ja y.
Vertaamalla näitä vektoreita saat loogisen vektorin, jolla indeksoimalla saat suoraan poimituksi sisäpisteet (ja ulkopisteet). Loppu onkin jo silkkaa nautiskelua (onhan tämäkin!)

Pisteitä piirretään `plot`-komennolla optioita hyväksikäyttäen: esimerkiksi
`plot(x,y,'ob',xsisa,ysisa,'*r')` piirtää kaikki pisteet sinisinä rinkuloina ja sisäpisteet punaisina tähtinä.

Tiedoksi:

Kysymys siitä, kuuluuko piste(joukko) jonkin monikulmion sisäpuolelle, on monimutkainen ja tärkeä erinäisissä sovelluksissa (vaikkapa FEM). Siihen on funktio `inpolygon`, joka on huomattavan monipuolinen funktio, mutta tämän tehtän kannalta tarpeettoman järeää.

DokuT **19. Palautettava teht.**

Huom: Tehtävänanto on siksi pitkä, että suurin osa on ohjetta. Ratkaisu lyhyt.

Tutkitaan heitetyn pallon lentorataa MATLABilla. Aloita luomalla m-tiedosto johon kirjoitat tarvittavat komennot.

1. Teemme seuraavat lähtöoletukset:

   i Pallon korkeus h heittohetkellä on $1.5m$

   ii Putoamiskiihtyvyys g on $9.8m/s^2$

   iii Pallon vauhti v heittohetkellä on $4m/s$

   iv Pallon etenemisvektorin suunta θ on 45^o

   Kirjoita oletukset skriptiisi.

2. Luo vektori `t`, jossa on 1000 tasaisin välein valittua arvoa väliltä $[0, 1]$.

3. Kuvataan muuttujalla x pallon etäisyyttä heittäjästä (mitattuna maan pinnalla) ja muuttujalla y pallon korkeutta, seuraavat yhtälöt kuvaavat muuttujien riippuvuutta ajasta ja oletetuista parametreista.

   (a)
   $$x(t) = v\cos\left(\theta\frac{\pi}{180}\right)t.\text{Muunnetaan kulma radiaaneiksi}$$

   (b)
   $$y(t) = h + v\sin\left(\theta\frac{\pi}{180}\right)t - \frac{1}{2}gt^2.$$

   Kirjoita annettujen yhtälöiden ja määrittelemiesi arvojen avulla vektorit `x` ja `y`.

4. Arvioidaan hetkeä jolloin pallo putoaa maahan, ja sen lentämää matkaa: etsi ensimmäinen indeksi, jolla pallon korkeus y muuttuu negatiiviseksi (käytä funktiota `find`). Pallon lentämä etäisyys on vektorin x arvo tässä indeksissä, lentoaika on vektorin t arvo tässä indeksissä. Tulosta sekä lentomatka että -aika näkyviin ruudulle.

5. Piirretään pallon lentorata: piirrä kuva, jossa pisteiden x-koordinaatit ovat vektorissa x, ja y-koordinaatit vektorissa y. Tämän jälkeen piirrä nolla-taso näkyviin katkoviivalla.