

Laskuharjoitus 4

Tee seuraavat tehtävät MATLABILLA. Testejä varten on kätevää muodostaa satunnaisia matriiseja ja vektoreita komentojen `rand` ja `randn` avulla. Esimerkiksi

- (i) `2*rand(5,1)-1` antaa vektorin jonka alkiot ovat tasajakautuneita satunnaislukuja välillä $[-1, 1]$.
- (ii) `2*randn(5,1)-1` antaa vektorin jonka alkiot ovat normaalijakautuneita satunnaislukuja joitten odotusarvo on -1 ja hajonta on 2 .

Komento `norm` antaa sekä vektorin että matriisin normin. Pelkkä `norm` antaa Euklidisen normin ja `norm(ma,1)` (vastaavasti `norm(ma,inf)`) antaa ykkös-normin (maksiminormin).

Muutamia vinkkejä kuvan piirtämiseen.

- (a) Jos halutaan piirtä käyrä $y = x^3 - 4x - 1$ välillä $[-1, 2]$ niin annetaan komennot `x=linspace(-1,2); y=x.^3 -4*x-1; plot(x,y)`
- (b) Yksikköympyrä tulee seuraavasti: `t=linspace(0,2*pi); plot(cos(t),sin(t))`
- (c) Jos haluaa 2 eri kuvaa samaan, niin piirtämisen välillä annetaan komento `hold`.
- (d) Jos halutaan piirtää pelkät pisteet eikä käyrää joka menee niitten kautta niin annetaan komento `plot(x,y,'*')`.
- (e) Jos v on kompleksinen vektori, niin `plot(v,'*')` antaa vektorin pisteet kompleksitasossa.

1. MATLABISSA polynomi voidaan tulkita vektorina jossa on polynomin kertoimet:

$$p = [1, 2, 3, 4] \longleftrightarrow p = x^3 + 2x^2 + 3x + 4$$

Polynomin nollakohdat saadaan komennolla `roots`. Valitse satunnaisia polynomeja ja pohdi väitettä:

satunnaisen polynomin nollakohdat ovat yksinkertaisia

Komennon `roots` ”käänteisfunktio” on `poly`. Kun sille antaa nollakohdat niin se laskee vastaavan polynomin.

2. Luentorungon lemma 2.5, s. 23 antaa arvion kiekosta jossa polynomin kaikki nollakohdat ovat. Valitse satunnaisia polynomeja, ja tutki tämän arvion hyvyttä. Siis näyttääkö siltä, että nollakohdat ovat enemmän tai vähemmän satunnaisesti kyseisessä kiekossa, vai näyttääkö siltä että paljon pienempikin kiekko tuntuisi riittävän? Huomaa muuten, että lemma pätee myös kompleksikertoimisille polynomeille.
3. Kirjoita editorin avulla tiedosto joka laskee funktion

$$f(x) = \arctan((x-1)^3 - 3x - 1) - x/3 - 4\sin(x)$$

Piirrä kuva. Montako nollakohtaa funktiolla f on? Laske joitain nollakohtia komennolla **fzero**. Piirrä näitä ratkaisupisteitä samaan kuvaan.

4. Yhtälösystemeitä voi ratkaista komennolla **fsolve**. Tarkastellaan systeemiä

$$\begin{cases} 5x_1^2 - x_2^2 = 0 \\ \sin(x_1) + \cos(x_2) - 4x_2 = 0 \end{cases}$$

Montako ratkaisua on? Etsi ratkaisut **fsolven** avulla.

5. Anna komento `opt0=optimset('fsolve')`. Nyt tietorakenteessa `opt0` on kaikki komennon **fsolve** parametrin oletusarvot. Kun sitten halutaan muuttaa vain jotain parametrin arvoa ja pitää muut entisellään niin voidaan antaa esimerkiksi komento `opt=optimset(opt0,'Display','iter')`. Tämän parametrin muuttaminen aiheuttaa sen että algoritmi joka kieroksella antaa tietoa laskennan edistymisestä. Jos sitten halutaan antaa tehtävän Jacobin matriisi eksplisiittisesti, niin pitää antaa

$$\text{opt}=\text{optimset}(\text{opt0},\text{'Display'},\text{'iter'},\text{'Jacobian'},\text{'on'})$$

Kokeile näitä edellisen tehtävän tapauksessa.

6. Ratkaise tehtävä

$$\begin{cases} x_1^3 + x_1^2 x_2 - x_1 x_3 + 6 = 0 \\ e^{x_1} + e^{x_2} - x_3^3 = 0 \\ x_2^2 - 2x_1 x_3 - 4 = 0 \end{cases}$$

Nyt ei ole selvää kuinka monta ratkaisua on. Kokeile eri alkuarvoja. Löytyykö useita ratkaisuja?