LUENTOMONISTE:

Sovellusohjelmointi MATLAB**-ympäristössä**

Tommi Kärkkäinen

JYVÄSKYLÄN YLIOPISTO
Informaatioteknologian tiedekunta
Tietotekniikan laitos
syksy 2000

Kurssin rakenne

- perusluentoja 8 viikkoa 2 h/viikko = 16 h: 26.10.–14.12. Ag C232.1 (TK, huone Ag C412.1)
- demoja 7 viikkoa 4 h/viikko = 28 h: 30.10.–14.12. (EH, huone Ag C424.1):
    1. maanantai 10–12 Ag B113.1 (5.3.2001 saakka),
    2. torstai 12–14 Ag B211.1
- seminaaritöiden teko, III periodi 2001, viikot 3-8:
    1. pääteohjaus ekana demoaikana maanantaina 10–12 Ag B113.1
    2. yleisohjaus luentoaikaan torstaina 10–12 mutta salissa Ag C 221.1
- seminaaritöiden palautus ja opponointi 6–8 h: viikolla 9, 2001.
- loppukoe 9.3.2001

Kirjallisuutta

[1] MATLAB-OPPAAT: *Getting Started with* MATLAB, *Using* MATLAB, *Using* MATLAB *Graphics,* MATLAB *Application Programmer's Interface Guide, Building GUIs with* MATLAB, MATLAB *5 New Features Guide*, MATLAB *5 Release Notes*, *Using Simulink*, ..., MathWorks, Inc., Natick, MA, 1984–1996.

[2] RUSSEL D. REED AND ROBERT J. MARKS II, *Neural Smithing; Supervised Learning in Feedforward Artificial Neural Networks*, The MIT Press, Massachusetts, 1999.

[3] KAISA MIETTINEN, *Optimointi*, Luentomoniste 33, Jyväskylän yliopisto, matematiikan laitos, 1998.

[4] MIKKO SAARIMÄKI, *Matriisiteoria*, Luentomoniste 31, Jyväskylän yliopisto, matematiikan laitos, 1994.

Sisältö

Luento 1: Johdanto, matriisit ja vektorit

Luento 2: MATLABin grafiikasta

Luento 3: Optimoinnin alkeita

Luento 4: MLP-verkko I

Luento 5: MLP-verkko II

Luento 6: API/MEXit ja GUIt

Luento 7: *Simulink*

Luento 8: Seminaarityöt

Luku 1

Johdanto

1.1 Mikä on MATLAB?

MATLAB (Matrix Laboratory) on lähtökohdistaan, sekalaisesta seurakunnasta lineaarialgebrallisia aliohjelmia, laajentunut täysiveriseksi ohjelmankehitysympäristöksi. Nykyisin yhteen ja samaan pakettiin integroituu erittäin kattava joukko valmiita laskentarutiineja (aliohjelmia), monipuoliset graafiset ominaisuudet, sekä tarvittaessa myös normaalit (alemman tason) ohjelmointikielen perusrakenteet. Valmiita aliohjelmia löytyy mm. dataanalyysiin ja tiedon visualisointiin, numeeriseen ja symboliseen laskentaan sekä erityisesti sovellusohjelmointiin, prototyypitykseen, simulointiin ja graafisten käyttöliittymien rakentamiseen. Lisäksi MATLABin käytettävyyttä voi lisätä linkittämällä siihen omia, C- tai Fortran-kielisiä (ali)ohjelmia. MATLABin valmiiden aliohjelmien toteutukset ovat myös avoimesti saatavissa ja niitä voidaan siten kopioida ja muokata tilanteen mukaan.

Ympäristön monipuolisuus mahdollistaa MATLABin käytön hyvin erilaisista lähtökohdista, erityyppisten (laskenta)algoritmien prototyypityksestä aina valmiiden, esimerkiksi graafisilla käyttöliittymillä varustettujen sovellusten tuottamiseen ja kaupallistamiseen. Tässä suhteessa (korvausta vastaan) saatavilla oleva ominaisuus konvertoida MATLABin avulla kehitetyt sovellukset suoraan C/C++ koodiksi mahdollistaa alustariippumattomien toteutusten tekemisen. Lisäksi myös mahdollinen asiakas välttyy hankkimasta koko kyseisen ohjelmiston käyttölisenssiä voidakseen hyödyntää tehtyä työtä.

MATLABin perusympäristöä täydentävät erilaiset "työkalupaketit" (toolboxes), joita voidaan hankkia tarpeen mukaan. Olemassaolevia laajennuksia löytyy mm. viestintä- (Communations) ja kontrollijärjestelmien (Control Systems, e.g. LMI Control, QFT Control Design, Robust Control), symbolisen matematiikan (Extended Symbolic Math), taloustieteiden (Financial), sumean logiikan (Fuzzy Logic), spektraalianalyysin (Higher-Order Spectral Analysis), kuvankäsittelyn (Image Processing), neuroverkkojen (Neural Networks), optimoinnin (Optimization), osittaisdifferentiaaliyhtälöiden (Partial Differential Equations), signaalinkäsittelyn (Signal Processing) ja tilastotieteen (Statistics) tarpeita varten. Koska luentojen pitäjälle ei makseta korvausta MATLABin mainosmiehenä olemisesta (jos maksettaisiin, niin en kai minä sitä ääneen huutelisi!), löytyvät tarkemmat selostukset oheismateriaaleineen MathWorksin kotisivulta `http://www.mathworks.com/(products/matlab/)`.

1.2 Mihin MATLABia käytetään tällä kursilla?

Kurssin lähtökohtana on liittää MATLABin käyttöön perehtyminen konkreettiseen prototyypitysongelmaan: *Kuinka opetan ja konfiguroin hermoverkon?* Hermoverkoista (neural networks) on tullut viime vuosina eräs keskeinen elementti ns. Laskennallisesti älykkäiden järjestelmien -nimellä (Computationally Intelligent Methods, Soft Computing) kulkevien tekniikoiden joukossa. Peruslähtökohtana on opettaa reaalimaailman dataa hyväksikäyttäen verkko-tietorakenteelle tuntematon kuvaus, joka liittää mitattujen input-output (sisään-ulos) -muuttujien arvot toisiinsa.

Sovellusohjelmointi MATLAB-ympäristössä -kurssin tavoitteena (mikään ei takaa, että tavoite saavutetaan, mutta konstruktivistisen oppimiskäsityksen mukaan prosessi on tärkeämpi kuin lopputulos :-) on tutustua siihen osaan MATLABia, jota tarvitsemme sovellusongelmamme prototyypittämiseen. Keskeinen osa kurssin sisältöä on toimia siten, että eri osaratkaisuja kehitettäessä meillä on aina mahdollisuus rakentamamme osakokonaisuuden oikeellisuuden verifiointiin. Yksinkertaisimmillaan (mikäli projektin aikataulu antaa myöten ;-) tämä onnistuu, jos meillä on mahdollisuus testata rakentamamme palaset vertaamalla niitä vastaavan ongelman toiseen toteutukseen. Voidaksemme lähteä kulkemaan meille viitoitettua, osin tuntematonta polkua verkko selässämme, meidän tulee hallita MATLABista ainakin seuraavia piirteitä:

- matriisien määritys ja lineaarialgebran perusoperaatiot
- optimointimenetelmien perusteet ja niiden toteuttaminen
- datan käsittelyn perusteet
- kaikissa vaiheissa, operaatioiden ja algoritmien havainnollistaminen ja visualisointi = GRAFIIKKA

1.3 Käytön perusteet

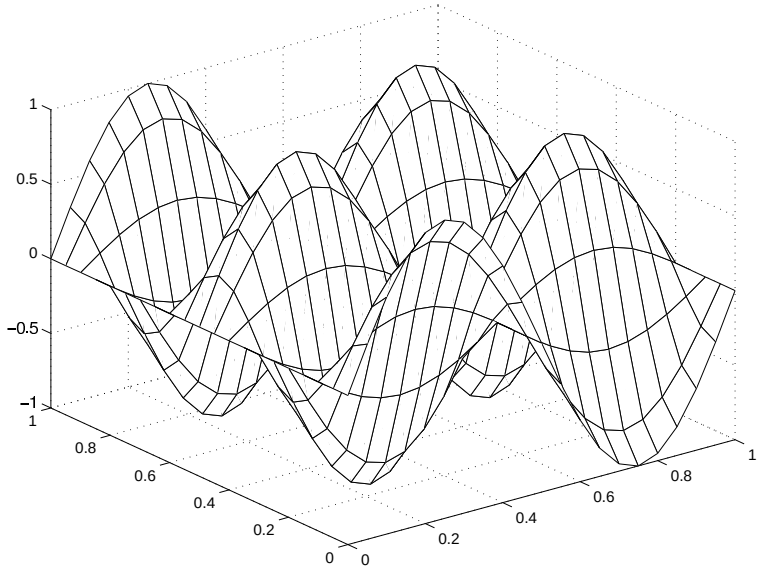
MATLAB toimii perustilassaan komentotulkkina, joka suorittaa komentorivillä annetun käskyn mukaisen toiminnan. Peruskäyttö rakentuu yleensä omien makrotiedostojen (.m-file) ajamisten ympärille. Parhaiten käyttöön pääsee kiinni tutustumalla valmiisiin esimerkkeihin `intro/demo` sekä erityisesti kattavaan valikoimaan käyttäjälle suunnattuja *helppejä*: `help, helpwin, doc, helpdesk, lookfor`. Peruskäyttöön tutustuminen ja sen harjoittelu on aiheena kurssin ensimmäisissä harjoituksissa.

1.3.1 Matriiseista ja vektoreista

Kuten nimestä saattaa nokkela lukija jo epäillä, MATLABin perustietorakenne on *matriisi*:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \in \mathbf{R}^{n \times m}. \tag{1.1}$$

Matriisin sisältämiä yksittäisiä reaalilukuja a_{ij} kutsutaan sen *komponenteiksi* ja MATLABin tallennusmuoto näille kuten kaikille muillekin luvuille on tuplatarkkuus *double precision*



Kuva 1.1: Funktion kuvaaja: h = 0:0.05:1; [x y] = meshgrid(h); z = sin(2*pi*x).*cos(4*pi*y); mesh(x,y,z);

(saa näkyviin muuttamalla komentilassa tulostusformaatiksi `format long`). Matriisi $\mathbf{A}$ voi esittää vaikkapa lineaarisen yhtälöryhmän

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n &= y_1 \\ \vdots \quad + \ldots + \quad \vdots &= \vdots \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= y_n \end{cases} \Leftrightarrow \mathbf{Ax} = \mathbf{y}$$

kerroinmatriisia $\mathbf{A} \in \mathbf{R}^{n \times n}$. Jos kyseistä matriisia $\mathbf{A}$ ajatellaan pelkästään tietorakenteena, se ei ole mitään muuta kuin kaksiulotteinen taulukko, jossa on n riviä ja m saraketta. Siten matriisin jokaiseen komponenttiin a_{ij} voidaankin liittää esim. jonkin funktion $a(x, y)$ arvo laskettuna annetuissa ns. *hilapisteissä* $(x_i, y_j) \in \mathbf{R}^2$:

$$\mathbf{A} = \begin{bmatrix} a(x_1, y_1) & \ldots & a(x_n, y_1) \\ \vdots & \ddots & \vdots \\ a(x_1, y_m) & \ldots & a(x_n, y_m) \end{bmatrix}.$$

Vastaavasti myös käytettyjen hilapisteiden koordinaatit voidaan esittää täsmälleen samanlaisissa rakenteissa

$$\mathbf{X} = \begin{bmatrix} x_1 & \ldots & x_n \\ \vdots & \ddots & \vdots \\ x_1 & \ldots & x_n \end{bmatrix} \quad \text{ja} \quad \mathbf{Y} = \begin{bmatrix} y_1 & \ldots & y_1 \\ \vdots & \ddots & \vdots \\ y_m & \ldots & y_m \end{bmatrix}.$$

Kun nämä kolme taulukkoa "pinotaan päällekkäin", saadaan määriteltyä $n \times m$-kappalatta $\mathbf{R}^3$:n pisteitä, joiden avulla funktion $a(x, y)$ kuvaaja voidaan piirtää näkyviin kuvan 1.1 esimerkin mukaisesti. Täsmälleen samalla tavalla voidaan määrätä myös esim. `rgb`-arvot, jotka liittävät kuhunkin pisteeseen jonkin värin ja siten mahdollistavat värikuvien piirtämisen ja käsittelyn.

Matemaattiselta kannalta matriisit ovat ns. *lineaarikuvauksia*. Kuten kaikille kuvauksille, täytyy meillä tällöin olla olemassa jokin sopivanmuotoinen alkio johon kuvauksia voidaan so-

veltaa. Matriiseille tämmöistä alkiota kutsutaan *vektoriksi*. Matriisille $\mathbf{A} \in \mathbf{R}^{n\times m}$ sopivanmuotoinen kaveri on vektori $\mathbf{x} \in \mathbf{R}^m \equiv \mathbf{R}^{m\times 1}$, koska kuvaus *matriisi*(*vektori*) määritellään muodossa

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} \;\Leftrightarrow\; y_i = \sum_{j=1}^{m} a_{ij}\,x_j.$$

Tästä nähdään, että matemaattisesti matriisi $\mathbf{A}$ on siis kuvaus $\mathbf{A} : \mathbf{R}^m \to \mathbf{R}^n$ avaruudelta $\mathbf{R}^m$ avaruudelle $\mathbf{R}^n$. Luennollisesti vektori itsessään on vain erikoistapaus matriisista.

Merkinnöistä: enemmän tai vähemmän johdonmukaisesti pyrimme merkitsemään reaalilukuja symboleilla $a, b, c, \ldots, w, x, y, z, \ldots$. Vektoreille käytetään merkintöjä $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$ ja matriiseihin viitataan symboleilla $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots, \mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \ldots$.

Tietorakenteena (*pysty*)vektorit $\mathbf{a}, \mathbf{x} \in \mathbf{R}^n \equiv \mathbf{R}^{n\times 1}$ eivät ole taaskaan mitään muuta kuin yksiulotteisia taulukoita, joihin voidaan esimerkiksi tallettaa jonkin funktion $a(x)$ arvot sekä pisteet, joissa arvot on laskettu:

$$\mathbf{a} = \begin{bmatrix} a(x_1) \\ \vdots \\ a(x_n) \end{bmatrix} \quad \text{ja} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

Kun nämä kaksi listaa liitetään yhteen, saadaan määriteltyä n-kappalatta $\mathbf{R}^2$:n pisteitä, joiden avulla funktion $a(x)$ kuvaaja voidaan piirtää näkyviin.

Matemaattisina otuksina matriisit täydentyvät, kun niille määritellään tarvittavat laskutoimitukset sopivalla tavalla (vrt. luokka ja metodit):

Transpoosi: $\mathbf{B} = \mathbf{A}^T \;\Leftrightarrow\; b_{ij} = a_{ji}$ (MATLAB `b=a'`).

Skalaarilla kertominen: $\mathbf{B} = t\mathbf{A} \;\Leftrightarrow\; b_{ij} = ta_{ij}$ (MATLAB `b=t*a`).

Yhteenlasku: $\mathbf{C} = \mathbf{A} + \mathbf{B} \;\Leftrightarrow\; c_{ij} = a_{ij} + b_{ij}$ (MATLAB `c=a+b`).

Kertolasku: $\mathbf{C} = \mathbf{AB} \;\Leftrightarrow\; c_{ij} = \sum_k a_{ik}b_{kj}$ (MATLAB `c=a*b`) .

Komponenttien kertolasku: $c_{ij} = a_{ij}b_{ij}$ (MATLAB `c=a.*b`).

Komponenttien osamäärä: $c_{ij} = a_{ij}/b_{ij}$ ($\mathbf{b}_{ij} \neq 0$) (MATLAB `c=a./b`).

Käänteismatriisi: $\mathbf{B} = \mathbf{A}^{-1} \;\Leftrightarrow\; \mathbf{BA} = \mathbf{AB} = \mathbf{I}$ (MATLAB `b=inv(a)`). HUOM: tietenkin sillä edellytyksellä, että käänteismatriisi $\mathbf{A}^{-1}$ on olemassa eli että matriisi $\mathbf{A}$ on *kääntyvä* ($\det(\mathbf{A}) \neq 0$).

Yhtälöryhmien ratkaisut: (vasen käänteinen) $\mathbf{Ax} = \mathbf{y} \;\Leftrightarrow\; \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ (MATLAB `x=a\y`), (oikea käänteinen) $\mathbf{xA} = \mathbf{y} \;\Leftrightarrow\; \mathbf{x} = \mathbf{yA}^{-1}$ (MATLAB `x=y/a`). HUOM: MATLAB toimii nopeammin ja tarkemmin kun em. yhtälöryhmät ratkaistaan annetuilla käskyillä eikä pelkästään kertomalla vektoria $\mathbf{y}$ käänteismatriisilla `inv(a)` oikealta tai vasemmalta. Käänteismatriisin muodostamiseen `inv`-komennolla käytetään ennalta määrättyä algoritmia, kun taas ratkaisukäskyihin \ ja / liittyen MATLAB etsii matriisin $\mathbf{A}$ ominaisuuksista riippuen "omasta mielestään" parhaan tavan ratkaista annettu tehtävä.

Ylläolevista määritelmistä on syytä huomata, että koska matriisilla lähtökohtaisesti tarkoitetaan matemaattista kuvausta, tarvitaan vastaavan rakenteen käsittelemiseksi taulukkona *erilliset* määritelmät ja niitä vastaavat symbolit taulukkotyylisille komponenttien kertolaskulle ja osamäärälle. Esimerkiksi Kuvan 1.1 tapauksessa piirtäminen aloitetaan luomalla vektori $h \in \mathbf{R}^{1\times 21}$ (sillä, onko kyseessä vaaka- vai pystyvektori, ei ole tässä vaiheessa merkitystä, koska todellisuudessa ollaan käsittelemässä vain taulukoituja lukuja), jonka avulla sitten generoidaan 2d-hilapisteet matriiseihin/taulukoihin, joiden komponenttien avulla lopuksi lasketaan halutun funktion arvot kolmanteen samankokoiseen matriisiin/taulukkoon. Ja eikun piirtämään...

1.3.2 Matriisien ominaisarvoista ja -vektoreista

Matriisien ja vektoreiden kanssa puljaamisen veryttelemiseksi tarkastellaan toivottavasti mahdollisimman monelle ennestään tuttuja juttuja.

Matriisin $\mathbf{A} \in \mathbf{R}^{n\times n}$ sanotaan olevan *symmetrinen*, jos $\mathbf{A}^T = \mathbf{A}$. Symmetrisen matriisin käänteismatriisi $\mathbf{A}^{-1}$ on myös symmetrinen, sillä

$$\mathbf{A}(\mathbf{A}^{-1})^T = \mathbf{A}^T(\mathbf{A}^{-1})^T \overset{(\mathbf{AB})^T=\mathbf{B}^T\mathbf{A}^T}{=} (\mathbf{A}^{-1}\mathbf{A})^T = \mathbf{I}^T = \mathbf{I},$$

josta seuraa (kertomalla vasemmalta matriisilla $\mathbf{A}^{-1}$) $(\mathbf{A}^{-1})^T = \mathbf{A}^{-1}$.

Neliömatriisin $\mathbf{A} \in \mathbf{R}^{n\times n}$ *ominaisarvot* $\{\lambda_k\}$ ovat reaalilukuja ja *ominaisvektorit* $\{\mathbf{u}_k\}$ ($\mathbf{u}_k \neq 0$) $\mathbf{R}^n$:n vektoreita, jotka yhdessä toteuttavat seuraavan yhtälön

$$\mathbf{A}\mathbf{u}_k = \lambda_k\mathbf{u}_k \quad k = 1, \ldots, n. \tag{1.2}$$

Tästä seuraa välittömästi (kertomalla identiteetti molemmilta puolilta $\mathbf{A}^{-1}$:llä) hyödyllinen tulos (jota ei siksi varmaankaan tarvita tällä kurssilla :-) $\mathbf{A}^{-1}\mathbf{u}_k = 1/\lambda_k\mathbf{u}_k$, joka liittää matriisin ja sen käänteismatriisin ominaisarvot ja -vektorit toisiinsa. Yhtälö (1.2) voidaan kirjoittaa myös matriisimuodossa

$$\mathbf{AU} = [\mathbf{A}\mathbf{u}_1\ \mathbf{A}\mathbf{u}_2\ \ldots\ \mathbf{A}\mathbf{u}_n] = [\lambda_1\mathbf{u}_1\ \lambda_2\mathbf{u}_2\ \ldots\ \lambda_n\mathbf{u}_n] = \mathbf{UD}, \tag{1.3}$$

missä

$$\mathbf{D} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \lambda_n \end{bmatrix} \quad \text{ja} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n \end{bmatrix} \tag{1.4}$$

Yhtälöstä (1.2) saadaan

$$\mathbf{u}_j^T\mathbf{A}\mathbf{u}_k = \mathbf{u}_j^T\lambda_k\mathbf{u}_k = \lambda_k\mathbf{u}_j^T\mathbf{u}_k \quad \text{ja} \quad \mathbf{u}_k^T\mathbf{A}\mathbf{u}_j = \lambda_j\mathbf{u}_k^T\mathbf{u}_j, \tag{1.5}$$

josta, kun $\mathbf{A}$ on symmetrinen, seuraa ($\mathbf{u}_j^T\mathbf{A}\mathbf{u}_k = \mathbf{u}_k^T\mathbf{A}^T\mathbf{u}_j$ ja $\mathbf{u}_j^T\mathbf{u}_k = \mathbf{u}_k^T\mathbf{u}_j$)

$$(\lambda_k - \lambda_j)\mathbf{u}_j^T\mathbf{u}_k = 0. \tag{1.6}$$

Siispä, jos $\lambda_k \neq \lambda_j$, täytyy olla $\mathbf{u}_j^T\mathbf{u}_k = 0$, joka tarkoittaa sitä, että ominaisvektorit $\mathbf{u}_j$ ja $\mathbf{u}_k$, $j \neq k$, ovat toisiaan vastaan kohtisuorassa eli *ortogonaalisia*. Tämän lisäksi ominaisvektoreiden pituus skaalataan yleensä ykköseksi asettamalla $\mathbf{u}_k/||\mathbf{u}_k||$, jolloin pätee

$$\mathbf{u}_j^T\mathbf{u}_k = \delta_{ij} = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases}$$

Tällaisia (ominais)vektoreita kutsutaan keskenään *ortonormaaleiksi*. Ominaisvektorit sisältävän matriisin $\mathbf{U}$ suhteen tämä merkitsee sitä, että

$$\mathbf{U}^T\mathbf{U} = \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix} = \mathbf{I} = \mathbf{U}\mathbf{U}^T, \tag{1.7}$$

missä $\mathbf{I}$ on *identtinen matriisi*

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} = \mathrm{Diag}\{1, \dots, 1\}.$$

Tällöin sanotaan, että matriisi $\mathbf{U}$ on *unitaarinen*, jota merkitään $\mathbf{U} \in \mathrm{Unit}(n)$.
Matriisiesityksestä (1.3) ja identiteetistä (1.7) seuraa, että

$$\mathbf{U}^T\mathbf{A}\mathbf{U} = \mathbf{U}^T\mathbf{U}\mathbf{D} = \mathbf{D} \;\Leftrightarrow\; \mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^T. \tag{1.8}$$

Tätä operaatiota kutsutaan matriisin $\mathbf{A}$ *diagonalisoimiseksi*. Yleisemmin pätee tulos, jonka mukaan mv. neliömatriisi $\mathbf{A}$ voidaan esittää muodossa $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^T$, missä $\mathbf{T}$ on ns. yläkolmiomatriisi ($\mathbf{t}_{ij} = 0$ kun $i > j$). Diagonalisointikaavan (1.8) mukaisesti voidaan matriisille $\mathbf{A}$ määritellä *neliöjuuri* kaavalla $\mathbf{A}^{\frac{1}{2}} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}\mathbf{U}^T$, sillä määritelmän perusteella saadaan $(\mathbf{A}^{\frac{1}{2}})^2 = (\mathbf{A}^{\frac{1}{2}})^T\mathbf{A}^{\frac{1}{2}} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}\mathbf{U}^T\mathbf{U}\mathbf{D}^{\frac{1}{2}}\mathbf{U}^T = \mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{A}$.
Ominaisvektorimatriisin $\mathbf{U}$ avulla voidaan annettu vektori $\mathbf{x}$ muuntaa uudeksi vektoriksi $\mathbf{y}$ asettamalla $\mathbf{y} = \mathbf{U}^T\mathbf{x}$. Käyttämällä uudelleen kaavaa (1.7) nähdään, että näin määritellyssä muunnoksessa alkuperäisen vektorin *Euklidinen* pituus säilyy samana:

$$\|\mathbf{y}\|^2 = \mathbf{y}^T\mathbf{y} = (\mathbf{U}^T\mathbf{x})^T\mathbf{U}^T\mathbf{x} = \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2.$$

Vastaavasti myös kahden vektorin $\mathbf{x}_1, \mathbf{x}_2$ välinen kulma $\cos(\theta(\mathbf{x}_1, \mathbf{x}_2)) = \frac{\mathbf{x}_1^T\mathbf{x}_2}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|}$ säilyy muunnoksessa samana, sillä

$$\mathbf{y}_1^T\mathbf{y}_2 = (\mathbf{U}^T\mathbf{x}_1)^T\mathbf{U}^T\mathbf{x}_2 = \mathbf{x}_1^T\mathbf{U}\mathbf{U}^T\mathbf{x}_2 = \mathbf{x}_1^T\mathbf{x}_2.$$

Lopullinen tulkinta on siis se, että vektorin kertominen matriisilla $\mathbf{U}^T$ suorittaa käytetyn koordinaatiston rotaation, esim. $\left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right\} \;\Leftrightarrow\; \left\{\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}-1\\0\end{bmatrix}\right\}$.
Matriisin $\mathbf{A}$ avulla voidaan määritellä reaaliarvoinen, ns. *kvadraattinen muoto*

$$\mathcal{J}(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

Matriisin $\mathbf{A} = \mathbf{A}^T$ sanotaan olevan *positiividefiniitti* (positiivisesti semidefiniitti), jos kvadraattinen muoto $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ (≥ 0) kaikille $\mathbf{x} \neq 0$. Geometrinen tulkinta tilanteelle saadaan käyttämällä jälleen muunnosta $\mathbf{y} = \mathbf{U}^T\mathbf{x}$ ja jo tutuksi tullutta temppuilua matriisin $\mathbf{U}$ ominaisuuksilla:

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{A}\mathbf{U}\mathbf{U}^T\mathbf{x} = \mathbf{y}^T\mathbf{U}^T\mathbf{A}\mathbf{U}\mathbf{y} = \mathbf{y}^T\mathbf{D}\mathbf{y} = \sum_{k=1}^{n} \lambda_k \mathbf{y}_k^2 = \sum_{k=1}^{n} \frac{\mathbf{y}_k^2}{\left(\lambda_k^{-\frac{1}{2}}\right)^2}. \tag{1.9}$$

Geometrisesti tämä tarkoittaa sitä, että muunnetussa koordinaatistossa $\mathbf{U}^T\mathbf{e}_k$ ($\mathbf{e}_k$:t ovat alkuperäisen koordinaatiston kantavektoreita) kvadraattinen muoto määrittää n-ulotteisen hyperellipsin, jonka pääakseleiden pituudet ovat verrannollisia $\lambda_k^{-\frac{1}{2}}$:een. Lisäksi kaavasta (1.9) nähdään, että symmetrisen ja positiividefiniittisen (lyh. SPD) matriisin $\mathbf{A}$ kaikki ominaisarvot λ_k ovat positiivisia.

Jos matriisi $\mathbf{A} \in \mathbf{R}^{m\times n}$ ei ole neliömatriisi ($n \neq m$), kaavaa (1.8) vastaa ns. *singulaariarvohajotelma*:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \;\Leftrightarrow\; \mathbf{U}^T\mathbf{A}\mathbf{V} = \mathbf{S}, \tag{1.10}$$

missä $\mathbf{U} \in \mathrm{Unit}(m)$ ja $\mathbf{V} \in \mathrm{Unit}(n)$. Matriisi $\mathbf{S}$ on muotoa

$$\mathbf{S} = \begin{bmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{bmatrix},$$

missä $\mathbf{D} \in \mathbf{R}^{r\times r}$ on diagonaalimatriisi, joka sisältää r kappaletta matriisin $\mathbf{A}$ positiivisia singulaariarvoja $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_r$. Tässä r on matriisin $\mathbf{A}$ *ranki*, joka ilmaisee matriisin sisältämien lineaarisesti riippumattomien vektoreiden dimension. Singulaariarvot $\mathbf{s}_1, \ldots, \mathbf{s}_r$ ja matriisit $\mathbf{V}$ ja $\mathbf{U}$ liittyvät ominaisarvotehtävän $\mathbf{A}^T\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \;\Leftrightarrow\; \mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{D}$ ratkaisuun seuraavan ketjun mukaisesti: $\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{D} \;\Leftrightarrow\; (\mathbf{A}\mathbf{V})^T\mathbf{A}\mathbf{V} = \mathbf{D} \;\Leftrightarrow\; \mathbf{U}^T\mathbf{A}\mathbf{V} = \mathbf{D} \;\Leftrightarrow\; \mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, missä $\mathbf{U} = \mathbf{A}\mathbf{V}$. Siis, jos $\mathbf{U} = \mathbf{A}\mathbf{V}$:sta tulee unitaarinen (niinkuin tapahtuu) ja sekä $\mathbf{U}$:n että $\mathbf{D}$:n dimensiot "laajennetaan" sopiviksi, antaa yo. ketju perustan singulaariarvohajotelman konstruktiolle.