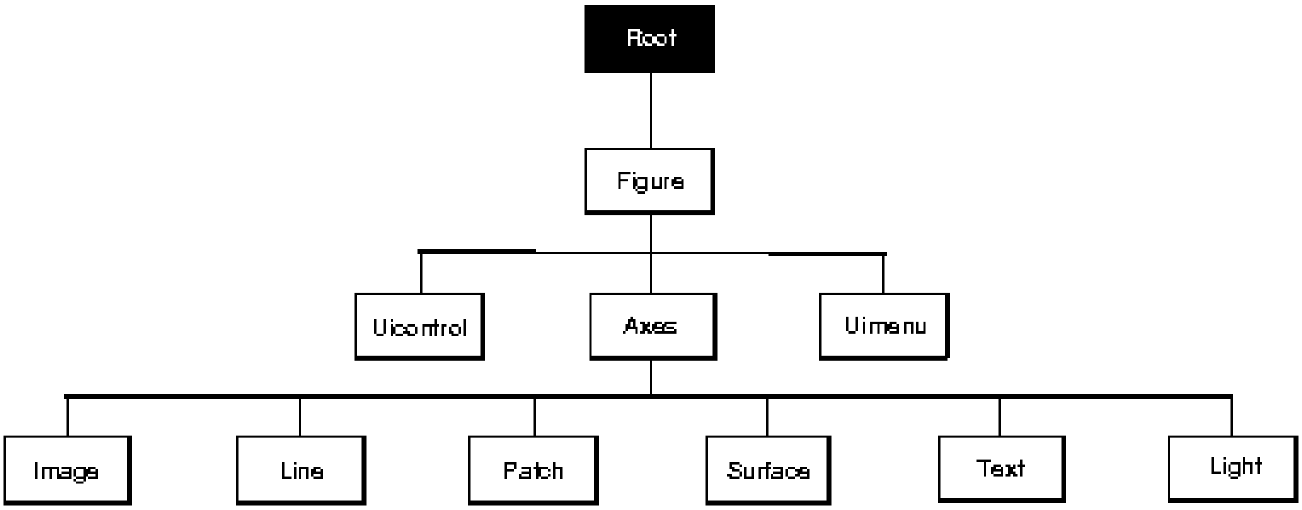
Luku 2

MATLABin grafiikasta

2.1 Graafiset objektit

MATLABin grafiikka perustuu hierarkkiselle rakenteelle graafisia objekteja. Objekteille asetetut ominaisuudet määrittävät niiden sisällön yksikäsitteisesti ja ominaisuudet myös periytyvät hierarkkiassa alaspäin vanhemmilta lapsille. Varsinainen näyttöpäätteellä näkyvä kuva muodostuu niiden objektien ilmentymistä, jotka kyseisessä kuvassa näytetään. Luotuihin objekteihin saa kiinnitettyä “kahvan” (*Handle Graphics*) eli symbolisen osoittimen (jonka MATLAB näyttää reaalilukuna), jonka avulla objektien ominaisuuksia voi selvittää ja tarvittaessa muuttaa. Tällä tavoin ruudulla näkyvien kuvien ulkoasun voi muokata täsmälleen sellaiseksi kuin sen itse haluaa.

Graafisten objektien perushierarkkia on esitetty kuvassa 2.1:



Kuva 2.1: Graafisten objektien hierarkkia.

Valitsemalla MATLABin `helpdeskistä` `Handle Graphics Objects` päästään kuvan 2.1 mukaisesti tarkastelemaan lähemmin eri graafisia objekteja sekä niihin liitettyjä ominaisuuksia (atribuutteja). Hierarkkiassa esiintyvät graafiset objektit tarkoittavat seuraavaa:

`Root`: Graafinen objekti, joka viittaa tietokoneen kuvaputkeen. Luonnollisesti tällaisia objekteja voi olla luotuna vain yksi, sillä ei ole vanhempia ja objektin lapsina esiintyvät

Figure-objektit. Rootin (aktiivisia) ominaisuuksia ovat mm.
CallbackObject, Children, Clipping, CurrentFigure, Diary, DiaryFile, Echo, Format, PointerLocation, PointerWindow, ScreenSize, UserData, ...

Figure: Syntaksi

```
figure
figure('PropertyName',PropertyValue,...)
figure(h)
h = figure(...)
```

Ruudulla näkyvä yksittäinen piirtoikkuna, johon MATLABin graafinen tulostus voidaan ohjata. Ikkunoita voi olla samanaikaisesti auki mv. lukumäärä (ainakin MATLABin mielessä, Window Manager saattaa päättää toisin). Objektin vanhempana on aina Root ja sen lapsina esiintyvät Axis, Uicontrol ja Uimenu. Figure on objektityyppi, joka luodaan automaattisesti annettaessa normaaleja piirtokomentoja kuten esimerkiksi plot, mutta objektin voi luoda sellaisenaankin komennolla figure. Figure-objektin ominaisuuksia ovat mm.

BusyAction, ButtonDownFcn, Children, Color, Colormap, CurrentAxes, CurrentObject, HandleVisibility, KeyPressFcn, MenuBar, Name, NextPlot, PaperPosition, PaperSize, Tag, Parent, Position, WindowButtonUpFcn, ...

Axes: Syntaksi

```
axes
axes('PropertyName',PropertyValue,...)
axes(h)
h = axes(...)
```

Tämä objekti määrittää Figuren sisään piirtoalueen, johon alemman tason objektit Image, Line, Patch, Surface, Text ja Light asettuvat. Axes-objektikin luodaan automaattisesti useimpien piirtokomentojen yhteydessä, mutta myös sen luonti onnistuu erilliselläkin käskyllä axes. Nimensä mukaisesti tähän objektiin liitetään piirrettävän kuvan koordinaattiakseleihin liittyvät tiedot (asteikot, nimet, tyylit, pistejoukot yms.) Siksi Axes-objektilla onkin mm. seuraavia ominaisuuksia:

Box, ButtonDownFcn, CameraPosition, Children, Color, DataAspectRatio, DrawMode, FontName, FontSize, GridLineStyle, XGrid, YGrid, ZGrid, XLabel, YLabel, ZLabel, XLim, YLim, ZLim, ...

Uicontrol: Käyttöliittymään liitettyjen toimintojen kontrolloinnin mahdollistava objekti. Tähän palataan tarkemmin, kun käsitellään graafisten käyttöliittymien (GUI) rakentamista MATLABin avulla.

Uimenu: Figure-ikkunoiden yläpalkkien sisältämien valikkorakenteiden (*menus*) sisältö ja käsittely. Myös tämä liittyy luonnollisesti graafisten käyttöliittymien rakentamiseen.

Line: Graafinen perusprimitiivi, jonka ominaisuudet määräävät useimmat 2D- ja 3D-kuvat. Line-objekti liitetään aina (aktiivisen tai aktivoitavan) Axes-objektin lapseksi ja niinpä käytetty koordinaatisto onkin eräs Axesissa määritelty ominaisuus. Line-objektin luovat automaattisesti mm. piirtokomennot plot, plot3, contour, countour3 jne.

Ominaisuudet tälle objektille ovat suuressa määrin samoja kuin edellisilläkin objekteilla. Lisäksi mm. ominaisuus `LineStyle` määrittelee käytetyn viivatyypin (-/–/:/-/`none`) ja `LineWidth` sekä `Marker` viittaavat piirrettävän viivan paksuuteen ja annettuihin pisteisiin piirrettävän merkin, "markkerin", (esim. +/$\circ$/$\star$/$\cdot$/$\times$/$\square$/$\diamond$/$\triangleright$/$\triangleleft$) tyyppiin. Huomaa, että jos samaan akselistoon (`axes`) piirretään useamman funktion kuvaajia, MATLAB kierrättää viivatyyppejä ja markkereita automaattisesti.

`Patch`: Graafinen objekti, joka määrittelee väritetyn 2- tai 3-ulotteisen monikulmion aktiiviselle akselistolle (`axes`) kulmien koordinaattien avulla. Näitä ei tällä kurssilla tarvita, mutta esim. komennot `fill` ja `fill3` aikaansaavat `Patch`-objektin luomisen.

`Surface`: Kolmiulotteinen pintaesitys matriisi(e)n sisältämästä datasta, joka määrittää pisteistön (x_i, y_j, z_k) avaruudessa $\mathbf{R}^3$. `Surface`-objekti voi esittää annettua pisteistöä joko värien tai vain pisteet yhdistävän viivaston avulla. Jos väritystä ei ole erikseen määrätty, käytetään pinnan korkeuden suhteen automaattisesti skaalattuja värejä. Esimerkiksi komennot `pcolor`, `surf` ja `mesh` luovat `Surface`-tyyppisen objektin, joka määrittää piirrettävän kuvan. Objektilla on mm. ominaisuudet

`CData, CDataMapping, CreateFcn, EdgeColor, FaceColor, MeshStyle, ...`

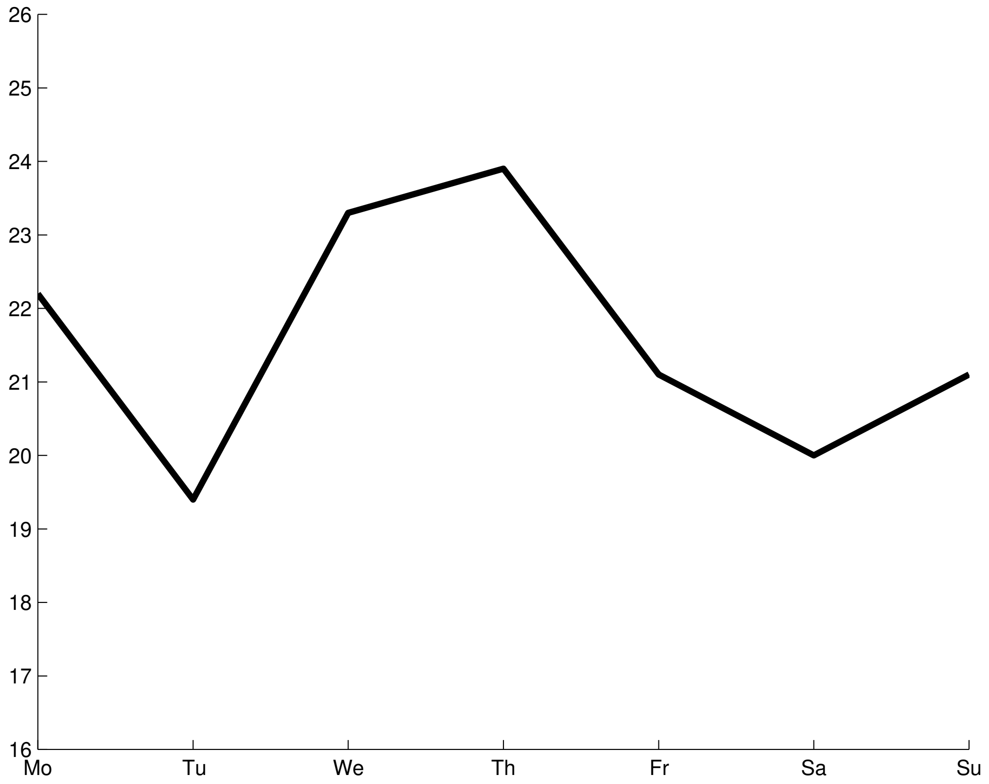
`Image`: Graafinen objekti, joka liittää (interpoloi) matriisin sisältämät luvut joko ikkunaan (`figure`) liittyvään *värikarttaan* (*indexed image*) tai suoraan RGB-arvoiksi (*truecolor*). Jokaisella `Image`-objektilla on yleensä oma värikarttansa eli *palettinsa*, joka määrittelee kuvan piirtämisessä käytettävät värisävyt. Tässä kohden on hyvä huomata, että muistin säästämiseksi MATLABissa kuville käytetty talletusmuoto on 8-bittinen kokonaislukuesitys (`uint8`). Toisin kuin `Surface`-objektit, `Image`-objektit ovat vain kaksiulotteisia, joten niiden katselupiste (jonka pystyi muuttamaan `Surface`ssa) on aina vakio. Objekti on aina `Axes`in, joka siis määrää käytetyn koordinaatiston sijainnin, lapsi. Näitä objekteja voi luoda itse komennolla `image`. Lisäksi hyodyllisiä komentoja ovat `imread` ja `imwrite`, joiden avulla voit lukea/kirjoittaa eri formaatissa (Windows Bitmap 'bmp', Hierarchical Data Format 'hdf', Joint Photographic Experts Group 'jpg'/'jpeg', Windows Paintbrush 'pcx', Tagged Image File Format 'tif'/'tiff', X Windows Dump 'xwd') olevia kuvia MATLABiin/ista. Ominaisuuksia `Image`lla ovat mm. `CData, CDataMapping, EraseMode, Selected, Tag, UserData, ...`

`Text`: Haluttuihin paikkoihin akselistolla (siis paikanmääritys kyseisen akseliston yksiköiden suhteen) sijoitettavia merkkijonoja, joiden avulla määritetään esimerkiksi koordinaattiakselien nimet, piirrettävän kuvan otsikko, siihen sisällytettävät lisätekstit jne. Ominaisuuksia mm.

`Color, Extent, FontAngle, Fontname, Fontsize, HorizontalAlignment,`
`Interpreter, Position, Rotation, String, VerticalAlignment, ...`

`Light`: Tällä objektilla määritellään `Patch`- ja `Surface`-objekteihin liittyvän valaistuksen ominaisuuksia (ei siis näy sellaisenaan missään), joita ovat mm. `Style, Position, ...`

Kaikille graafisille objekteille on luonnollisesti määritelty tietyt oletusominaisuudet, jotka ovat voimassa, jos objekti luodaan pelkällä peruskomenolla, esim. `figure`. Mikäli objektia luotaessa halutaan määritellä itse jokin sille asetettava ominaisuus, voi sen tehdä käyttämällä komentomuotoa `objekti('Ominaisuus',Arvo,...)`, esim. `figure('PaperType','a4')`. Yleensä objektit, joiden sisältämää informaatiota halutaan muuttaa, syntyvät kuitenkin jonkin annetun piirtokomennon seurauksena (*high-level/low-level functions*). Siksi graafiseen objektiin, johon muutettavaksi haluttu ominaisuus liittyy, täytyy päästä viittaamaan. Tätä varten tarvitaan ns. *kahva*. Määriteltävään objektiin voi liittää kahvan se luotaessa ja erityisesti jo luodun objektin ominaisuuksia voi kysellä ja muuttaa `get` ja `set` komentojen avulla. Tätä on havainnollistettu alla:




```
days = ['Mo'; 'Tu'; 'We'; 'Th'; 'Fr'; 'Sa'; 'Su'];
temp = [22.2 19.4 23.3 23.9 21.1 20 21.1];
f = figure;
a = axes ('YLim',[16 26],'Xtick',1:7,'XTickLabel',days);
h = line(1:7,temp);
set (h,'Color',[0 .8 .8],'LineWidth',3);
```

MATLABista löytyvät seuraavat graafisiin objekteihin liittyvät *kahvat* valmiina, joilla oletustilanteessa viitataan *viimeiseksi* luotuun kyseiseen objektiin:

`gcf` (*Graphical Current Figure*): Kahva aktiivisena olevaan `Figureen`, joka on talletettuna `Rootin` `CurrentFigure` kentässä. Aktiivinen ikkuna (ja yleisemmin aktiivinen objekti) viittaa siihen `Figureen`, joka on viimeksi luotu tai jonka käyttäjä on "aktivoinut" hiiren avulla.

`gca` (*Graphical Current Axes*): Aktiivisen `Figuren` aktiivinen `Axes`, johon viittaa `Figuren` kenttä `CurrentAxes`.

`gco` (*Graphical Current Object*): Aktiivisena oleva graafinen objekti.

Lisäksi graafisten objektien manipulointiin on käytettävissä mm. komennot `findobj`, `copyobj`, ja `delete`. Graafisiin objekteihin liittyvää atribuuttia `Tag` käytetään sopivan *nimilapun* liittämiseksi objektiin, jotta siihen voidaan myöhemmin tarttua objektisekamelskan keskellä mahdollisten muutosten tekemiseksi. Erityisen hyödyllinen kapistus näkymän muuntelussa on ns. *Ominaisuuseditori* (*Property Editor*, komento vaikkapa `propedit` tai `propedit(h)`, missä `h` on kahva haluttuun objektiin esim. `gcf`), jonka avulla objektien ominaisuuksiin pääsee käsiksi ja muuttamaan graafisen käyttöliittymän kautta.

Edellisten lisäksi kuvien kannalta oleellisia asioita ovat mm.

- asteikot (*Tick Marks*) x-, y- ja z-akseleilla: `xtick, ytick, ztick`
- eri akseleiden suhde toisiinsa (*Aspect ratio*): `axis normal/square/equal/tight`
- selventävät tekstit: `title, xlabel, ylabel, zlabel, legend, text, gtext`
- kuvissa käytetyt värit:

RGB-arvo	Lyhenne	Väri
[0 0 0]	`k`	musta (*black*)
[0 0 1]	`b`	sininen (*blue*)
[0 1 0]	`g`	vihreä (*green*)
[0 1 1]	`c`	turkoosi (*cyan*)
[1 0 0]	`r`	punainen (*red*)
[1 0 1]	`m`	violetti (*magenta*)
[1 1 0]	`y`	keltainen (*yellow*)
[1 1 1]	`w`	valkoinen (*white*)

2.2 2D grafiikka

Kaksiulotteinen (2D) grafiikka viittaa tilanteeseen, jossa MATLAB esittää kaksiulotteisen pisteistön (matriisin) (x_i, y_j) antamasi komennon mukaisesti. Tässä tulee siis huomata, että esim. yksiulotteisen funktion $f(x) = \exp(x)$ piirtämiseen MATLABilla tarvitaan x-koordinaatille määritelty hilapisteistö, esim. `xp = 0:0.1:2`, sekä tässä hilassa lasketut funktion arvot, esim. `yp = exp(xp)`, joiden avulla funktion kuvaaja (siis tarkemmin sen hilapisteiden kautta kulkeva, esim. lineaarinen aproksimaatio) voitaisiin piirtää vaikkapa komennolla

```
plot(xp,yp,'-mo','LineWidth',2,'MarkerEdgeColor','k',...
'MarkerFaceColor',[.49 1 .63],'MarkerSize',12)
```

Tyypillinen 2D grafiikkaa piirtävä makro voisi olla vaikkapa seuraavaa muotoa:

Mitä tehdään?	MATLAB komennot
0. Poistetaan vanha tauhka työtilasta	`clear` `close all`

1. Määritellään hilapisteet ja piirrettävät funktiot	`x = 0:.2:12;`
	`y1 = bessel(1,x);`
	`y2 = bessel(2,x);`
	`y3 = bessel(3,x);`

2. Määritellään ikkuna ja (ali)kuvan paikka	`figure(1)` `subplot(2,2,1)`
3. Piirretään kuvaajat	`h = plot(x,y1,x,y2,x,y3)`

4. Muokataan ulkoasua	`set(h,'LineWidth',2,{'LineStyle'},{'--';':';'-.'})`
	`set(h,{'Color'},{'r';'g';'b'})`
	`axis([0 12 -.5 1])`
	`set(gca,'XTick',[0 2 4 6 8 10 12])`
	`grid on`

5. Selitystekstejä	`xlabel('Aika')`
	`ylabel('Amplitudi')`
	`legend(h,'Eka','Toka','Kolmas')`
	`title('Bessel-funktioiden kuvaajia')`
	`[y,ix] = min(y1);`
	`text(x(ix),y,'Ekan minimi \rightarrow',...`
	`'HorizontalAlignment','right')`

6. Kuvaajat erikseen myös (ali)kuviin	`subplot(2,2,2)`
	`plot(x,y1,'r--','LineWidth',2)`
	`title('funktio Eka')`
	`subplot(2,2,3)`
	`plot(x,y1,'g:','LineWidth',2)`
	`title('funktio Toka')`
	`subplot(2,2,4)`
	`plot(x,y1,'b-.','LineWidth',2)`
	`title('funktio Kolmas')`

7. Kuvan talletus	`print -deps kuva1.eps`

2.3 3D grafiikka

Kolmiulotteinen (3D) grafiikka viittaa tilanteeseen, jossa MATLAB esittää kolmiulotteisen pisteistön (x_i, y_j, z_k) haluamallasi tavalla. Siten esimerkiksi kaksiulotteisen funktion $z = f(x,y) = \exp(x \cdot y)$ piirtämiseen tarvitaan (x,y)-koordinaatistolla ($\mathbf{R}^2$:lla) määritelty hilapisteistö, esim. `xp = 0:0.1:2; [x,y]=meshgrid(xp)`, sekä tässä hilassa lasketut funktion arvot `z = exp(x.*y)`, joiden avulla funktion kuvaajan aproksimaatio voitaisiin piirtää esim. komennolla `mesh(x,y,z)`. Muita mahdollisia komentoja 3D kuvien piirtämiseksi ovat

```
plot3, meshc, meshz, contour, surf, surfc, surfl, ...
```

Kuvien ulkoasuun liittyviä ominaisuuksia, joita voit muuttaa haluamallasi tavalla, ovat esimerkiksi *värikartta* (`colormap`), *varjostus* (`shading`), *katselupiste* (`view`), *piiloviivojen poisto* (`hidden`) ja *valaistus* (`light/lightning`). Tarkemmat ohjeet ja käyttöesimerkkejä läydät *help*eistä.