

MATLABin merkkijonot

- Merkkijono määritellään antamalla se yksinkertaisten lainausmerkkien sisällä
- esimerkiksi `A='Tässä on tekstiä'`, jonka jälkeen `A(1)=T`, `A(2)=ä` ...
- Merkkijonon sisältämä komento voidaan suorittaa komennolla, esimerkiksi `jono='A=[1 2;3 4]'` jonka jälkeen komento `eval(jono)` luo matriisin `A`
- Komennolla `feval` voidaan suorittaa merkkijonossa annettu funktio, esimerkiksi `feval(neliot,1,2)` on sama kuin kutsu `neliot(1,2)`

Funktiot, jotka käsittelevät funktioita

- Löytyvät komennolla `help funfun`
- Funktion nollakohdan laskeminen komennolla `fzero`:
 - `fzero('sin(x)+cos(x)',1)`
 - Kirjoitetaan tarkasteltava funktio M-tiedostoksi ja käytetään sitä, `fzero('nimi',1)`
- Lokaalin minimin etsiminen komennolla `fminbnd`
 - `fminbnd('x^2+10*sin(x)+1',-2,2)`
 - M-tiedoston avulla: `fminbnd('nimi',-2,2)`
- Numeerinen integrointi: `quad,quad8`
- 1. kertaluvun differentiaaliyhtälöiden numeerinen ratkaiseminen
- **Esimerkki** Harmoninen värähtelijä voidaan esittää differentiaaliyhtälöryhmänä

$$\begin{aligned}y_1' &= -y_2 \\ y_2' &= y_1\end{aligned}$$

Kirjoitetaan funktio M-tiedostona, joka palauttaa yhtälöryhmän oikean puolen

```
function dy = harmonic(t,y)
```

```
dy=zeros(2,1);
```

```
dy(1)=-y(2);
```

```
dy(2)=y(1);
```

- Käytetään ratkaisijaa `ode45`, `[T,Y]=ode45('harmonic',[0 10],[1,0])`
- Vektorissa `T` on ajanhetket $t_i = T[i]$ ja matriisissa `Y` vastaavat ratkaisun arvot $y_1(t_i) = Y(i,1)$ ja $y_2(t_i) = Y(i,2)$
- Funktioiden y_1 ja y_2 kuvaajat saadaan komennolla `plot(T,Y(:,1),T,Y(:,2))`
- Kuvaaja pisteistä (y_1, y_2) saadaan komennolla `plot(Y(:,1),Y(:,2))`

MATLABin graafiset ominaisuudet

- Piirretään numeerisen tiedon perusteella
- Lista piirtokäskyistä löytyy komennolla `help graph2d` ja `help graph3d`
- Kuvaajien ja pintojen piirtäminen ei ole yhtä automatisoitua kuin esimerkiksi Maplessa
- Uusia kuvaikkunoita saa luotua komennolla `figure`
- Kuvaaja tulee siihen ikkunaan, jota on viimeksi hiirellä klikattu tai joka on valittu `figure(numero)`-komennolla
- Piirtoikkunan saa tyhjennettyä komennolla `clf`

Kaksiulotteinen grafiikka

- Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ kuvaaja: Lasketaan vektoriin `x` ne arvot, joissa funktion arvot lasketaan, muodostetaan vektori `y` kyseisistä arvoista ja piirretään komennolla `plot(x,y)`
- Oletusarvoisesti MATLAB yhdistää peräkkäiset pisteet suoralla, tämän voi muuttaa antamalla `plot`-komennon perässä seuraavista merkeistä koostuvan merkkijonon
- **Esimerkki** `plot(x,y,'ro--')` piirtää kuvaajan punaisella, jokaisen pisteen kohdalle piirretään ympyrä ja pisteet yhdistetään katkoviivalla

y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

- `polar`-komennolla voidaan piirtää napakoordinaateissa annettujen käyrien kuvaajia, esimerkiksi `t=0:0.1:20*pi; r=exp(-t/10); polar(t,r);`
- Parametrikäyrien kuvaaja saadaan normaaliin tapaan `plot`-komennolla, esimerkiksi `t=0:0.1:2*pi; x=cos(t); \verb y=sin(t); plot(x,y);|`
- Kuvaajan akseleita säädetään `axis`-komennolla
 - `axis on`, `axis off` näyttää/ei näytä akseleita
 - `axis equal` skaalaa akselit siten, että molemmilla on sama yksikön pituus
 - `axis([xmin xmax ymin ymax])` asettaa x - ja y -akseleiden ylä- ja alarajat
- Samaan kuvaan saadaan useampia kuvaajia antamalla `plot`-komennolle enemmän x, y -pareja, esimerkiksi `plot(x1,y1,'b',x2,y2)`
- Toinen mahdollisuus on laittaa kuva "pysäytystilaan" komennolla `hold on`, jolloin uusi kuvaaja piirretään samaan koordinaatistoon kuin edellinenkin
- **Esimerkki** `t=0:0.1:2*pi; x=cos(t); y=sin(t); plot(x,y); hold on; polar(x,y);`
- Pysäytystila saadaan pois päältä komennolla `hold off`, pelkkä komento `hold` vaihtaa tilasta toiseen

Kolmiulotteinen grafiikka

- Funktion $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ kuvaaja: Lasketaan pisteet x ja y matriiseihin X, Y ja funktion arvot matriisiin Z

```
x=-2:0.1:2;  
y=-2:0.1:2;  
[X,Y]=meshgrid(x,y);  
Z=sin(X.*Y);  
surf(X,Y,Z);
```

- `plot3`-komennolla kuva voidaan piirtää käyrinä, yksi käyrä jokaista matriisien X, Y, Z saraketta kohden
- Parametripinnan piirtäminen tapahtuu normaaliin tapaan `surf`- tai `plot3`-komennolla
- **Esimerkki** Helikoidi

```
r = 0:0.1:1;  
phi = 0:0.1:6*pi;  
[R,P]=meshgrid(r,phi);  
H1=R.*cos(P);  
H2=R.*sin(P);  
H3=P;  
surf(H1,H2,H3);
```

Kuvien esittäminen

- Kuva, esimerkiksi digitaalikameralla otettu valokuva, esitetään MATLABissa matriisina, jossa on reaalilukuja.
- Kuva saadaan näkyville `image`-komennolla
- Kuvan värikartta saadaan vaihdettua `colormap`-komennolla
- Jos I on harmaasävykuva, niin matriisin alkion arvo kertoo, kuinka kirkas pikseli on, `colormap(gray(256))`
- Indeksoidulle kuvalle alkion arvo kertoo, mitä väriä kuvan väripaletista käytetään
- $m \times n \times 3$ matriisi on *RGB*-kuva, jossa jokaiselle pikselille määrätään erikseen sinisen, punaisen ja vihreän värikomponentin arvo
- Kuvien lukeminen MATLABiin onnistuu `imread`-komennon avulla