

Kort MATLAB-guide

Diverse

help	Ger en lista över grupper av kommandon
help <i>kommando</i>	Ger hjälptext för <i>kommando</i>
%	Inleder kommentarer
type <i>filnamn</i>	Listar innehållet i <i>filnamn.m</i>
clear	Raderar alla variabler
clear <i>var1 var2 ... varn</i>	Raderar variablerna <i>var1, var2, ..., varn</i>
format short	Utskrift med 4 decimaler (standard)
format short e	Utskrift med flyttalsformat, 5 siffror
format long	Utskrift med 14 decimaler
format long e	Utskrift med flyttalsformat, 16 siffror
format hex	Hexadecimal utskrift
format	Återställer till format short

Matrisoperationer

I MATLAB betraktas alla variabler som matriser. En skalär betraktas alltså som en 1×1 -matris.

A, B är matriser; x, y är vektorer; s är en skalär.

Matriserna måste ha sådana dimensioner att operationerna blir definierade.

$A = [8 \ 3 \ 4 ; 1 \ 2 \ 3 ; 9 \ 9 \ 7]$	A blir en 3×3 -matris.
	Semikolon kan bytas ut mot radbyte
$x = [1 \ 2 \ 3]$	x blir en 1×3 -matris
$y = [1 ; 2 ; 3]$	y blir en 3×1 -matris
$A + B, A - B$	Matrisaddition resp. matrissubtraktion
$A + s$	s adderas till alla element i A
$A * B, A * y$	Matrismultiplikation
$A .* B, A ./ B$	Elementvis multiplikation resp. division
$A.^2$	Ekvivalent med $A .* A$
$A \setminus b$ eller $x = A \setminus b$	Löser ekvationssystemet $Ax = b$
$A(i, j)$	Matriselementet A_{ij}
$A(:, j)$	j -te kolumnen i A
$A(:, j : k)$	En matris som består av kolumnerna j till k i A
$A(1 : 2 : 5, :)$	En matris som består av raderna 1, 3 och 5 i A

Standardfunktioner

Följande elementära funktioner opererar elementvis på en matris A .

abs(A)	sqrt(A)	round(A)	sin(A)	cos(A)	tan(A)
asin(A)	acos(A)	atan(A)	sinh(A)	cosh(A)	tanh(A)
exp(A)	log(A)	log10(A)			

Notera att log betyder naturlig logaritm och att log10 betyder $^{10}\log$.

Matrisfunktioner

<code>zeros(m,n)</code>	Ger en $m \times n$ -matris med enbart nollor
<code>ones(m,n)</code>	Ger en $m \times n$ -matris med enbart ettor
<code>eye(n)</code>	Ger en $n \times n$ -enhetsmatris
<code>diag(y,k)</code>	Bildar en diagonalmatris med vektorn y i k -te diagonalen
<code>rand(m,n)</code>	Ger en slumpvalsgenererad $m \times n$ -matris
<code>hilb(n)</code>	Ger en Hilbertmatris av storlek $n \times n$
<code>triu(A)</code>	Ger A men med elementen under diagonalen nollställda
<code>tril(A)</code>	Ger A men med elementen över diagonalen nollställda
<code>A'</code>	Ger A^T , dvs transponatet till A
<code>inv(A)</code>	Ger A^{-1} , dvs inversen till A
<code>x = A\b</code>	Ger lösningen till det linjära ekvationssystemet $Ax = b$
<code>[L,R,P] = lu(A)</code>	LR -uppdelar A ($LR = PA$)
<code>eig(A)</code>	Ger en vektor med A :s egenvärden
<code>det(A)</code>	Ger determinanten till matrisen A
<code>rank(A)</code>	Ger rangen till matrisen A
<code>norm(A)</code> eller <code>norm(A,2)</code>	Ger 2-normen till en matris (eller vektor)
<code>norm(A,inf)</code>	Ger maxnormen för en matris (eller vektor)
<code>cond(A)</code>	Ger konditionstalet i 2-norm för matrisen A
<code>cond(A,inf)</code>	Ger konditionstalet i maxnorm för matrisen A

Grafik

punkttyper:	'.' '+' '*' 'o' 'x' (är några exempel)
linjetyper:	'_' '--' ':' '-.'
färger:	y m c r g b w k
<code>plot(x, 'b+')</code>	talparen (j, x_j) ritas med symbolen $+$ i blått
<code>plot(x, y, '+')</code>	ritar vektorn y mot x som punkter med symbolen $+$
<code>plot(x, y)</code>	ritar vektorn y mot x med heldragna räta linjer
<code>plot(x, y, t, z, '+')</code>	ritar vektorn y mot x med heldragen linje och vektorn z mot t med punkttypen $+$
<code>title('textsträng')</code>	skriver ut <i>textsträng</i> som rubrik till plotten
<code>xlabel('textsträng')</code>	skriver ut <i>textsträng</i> under x -axeln
<code>ylabel('textsträng')</code>	skriver ut <i>textsträng</i> till vänster om y -axeln
<code>hold on / off</code>	håller kvar / släpper bilden i grafikfönstret
<code>clf</code>	rensar grafikfönstret
<code>print</code>	skickar bild till printern

Funktion

<code>function [a,b]=funkt(c,d)</code>	inleder funktionen <code>funkt</code> på filen <code>funkt.m</code> utparametrar a och b , inparametrar c och d .
--	--

Programmering

Relationsoperatorer: $<$, \leq , $>$, \geq , $==$, $\sim =$ (betyder \neq)

Logiska operatorer: $\&$ (och), $|$ (eller), \sim (icke)

Operatorerna ovan opererar elementvis på matriser.

if *logiskt uttryck*

satsgrupp

else

satsgrupp

end

else-gruppen kan utelämnas

for $i = 1 : 2 : n$ (betyder **for** $i = 1$ **to** n **step** 2)

satsgrupp

end

while *logiskt uttryck*

satsgrupp

end

Avsluta satserna med ” ; ” så undertrycks utskrift (som kan bli mycket lång).

Filhantering

<code>diary filnamn</code>	Lagrar MATLAB-kommunikationen på filen <i>filnamn</i>
<code>diary off</code>	Stänger av lagringen
<code>diary on</code>	Sätter på lagringen (fortsätter på samma fil)
<code>save filnamn</code>	Sparar alla variabler på filen <i>filnamn.mat</i>
<code>save filnamn a1 a2 ... an</code>	Sparar variablerna a_1, a_2, \dots, a_n på filen <i>filnamn.mat</i>
<code>load filnamn</code>	Hämtar in variablerna som lagrats på <i>filnamn.mat</i>
<code>fil</code>	Exekverar kommandofilen <i>fil.m</i>
<code>funk (n,A)</code>	Anropar funktionen <i>funk</i> med aktuella parametrar n och A . <i>funk</i> skall finnas på ditt directory under namnet funk.m .

Funktionsuttryck som parameter till funktion

`f1 = inline ('.....')` Definierar funktionen *f1* m.a.p. den bokstav som använts.

`f2 = inline ('.....', 't', 'y')` Definierar funktionen *f2* (t, y).

Används sedan i anrop t.ex. `quad(f1,a,b)`

Alternativt definieras funktionen i filen *f1.m* och anropet blir `quad(@f1,a,b)`

Unix-kommando från Matlab

`!unixkommando` Utför *unixkommando* inifrån Matlab, t.ex. `!ls`