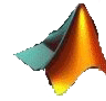




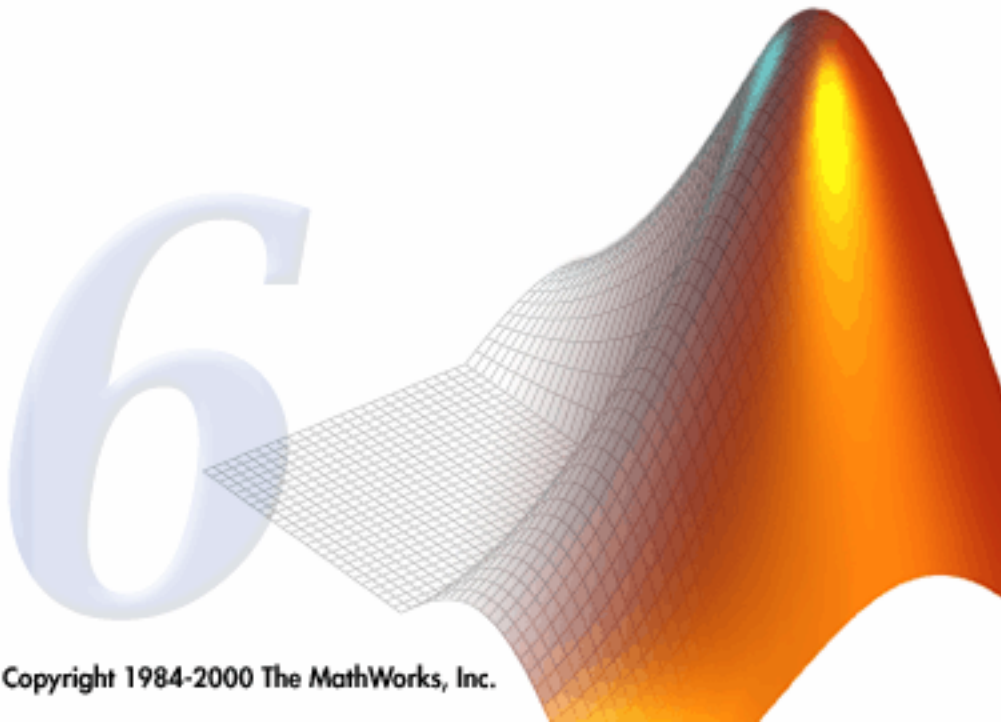
LUNDS TEKNISKA HÖGSKOLA
Lunds universitet



MATLAB

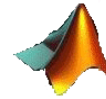
Introduktion till

MATLAB®

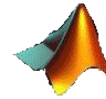


Copyright 1984-2000 The MathWorks, Inc.

av
Thomas Jacobsson



Inledning.....	3
Att arbeta med Matlab	3
Matlabsyntax och aritmetiska uttryck.....	4
Variabler	4
Matriser.....	5
Vektorer	7
Skript filer sk M-filer	8
Relations och logiska uttryck	9
In- och utdata.....	9
Utdata.....	10
Urval av vanliga MATLAB kommandon.....	13
Elementära matematiska funktioner	13
Komplexa funktioner.....	14
Matris och vektorfunktioner.....	14
Speciella matriser och vektorer	14
Polynomfunktioner	15
Statistiska funktioner	15
Grafik	15
Vektorberäkningar	16
Kommunikation mellan kommandofönstret och program (m-fil)	16
Logiska operatörer	16
Loopar och tester.....	17
Sifferformat	17
Tät eller lös utskrift på skämen	18



Inledning

Matlab är ett program primärt avsett för numerisk lösning av matematiska problem. Programmet bygger på en grundstomme bestående av en mängd s.k M-filer. Dessa filer är rena textfiler. Detta innebär att man har möjligheter att utveckla egna M-filer. Man har efter hand utvecklat en mängd en toolboxar, innehållande fler M-filer, avsedda för vitt skilda områden. Det finns t.ex. toolboxar för signalanalys, statistiska beräkningar och för reglersystem. Man har även tagit fram en toolbox (Simulink) som arbetar med ett grafiskt gränssnitt.

Alla data som hanteras i Matlab behandlas som matriser eller vektorer. Vid bruk av programmet så arbetar man i en interaktiv mod. Detta innebär att då du skriver ett kommando följer av <Enter> så utförs det aktuella kommandot omgående.

Matlab startas genom att klicka på ikonen för Matlab. Man kommer då in i ett kommandofönster.

Vill man undersöka lite vad Matlab kan göra så finns det en mängd färdiga program att köra.

Dessa färdiga program kommer man åt genom att i Matlabs kommandofönster efter prompten (>>) skriva *demo*.

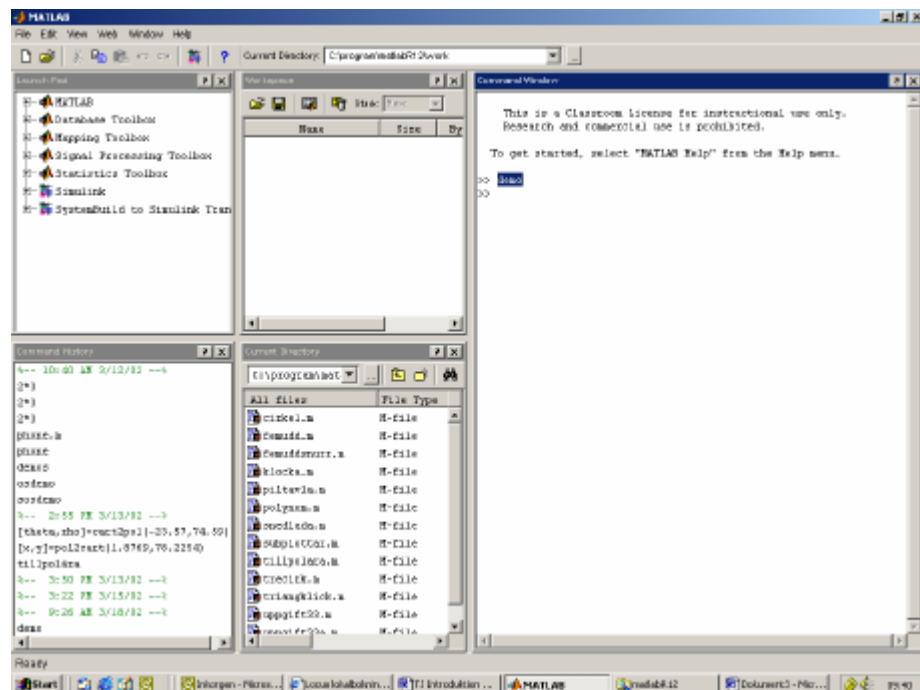
Att arbeta med Matlab

När man startat programmet så kan det vara lämpligt att man innan skapat ett eget bibliotek där man spar sina egna M-filer och data från beräkningar.

Vill man jobba mot ett eget bibliotek så måste man ange sökvägen dit. Antag att sökvägen är C:\program\matlabR12. För att temporärt skapa en sökväg så skriv i kommandofönstret: *path(path, 'C:\program\matlabR12')*.

Vill man ha sökvägen stationärt så måste man lägga till sin sökväg **pathbrowser**.

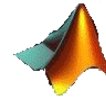
Från kommandofönstret meny kan några kommandon kommenteras.



File/New:	Öppnar en ny M-fil i matlabs egen editor
Open:	Öppnar en befintlig s.k M-fil.
Open select:	Visar ett i Command Window markerat kommando i Matlabs egen editor.
Save Workspace as:	Spar undan data och variabler i ett filformat med extentionen .MAT.
Load Workspace	Se.ovan
Show Workspace	Visar variablenamn och storlek på dessa.
Preferences	Sätter t.ex antalet decimaler, texttyp och copyoptions.

När man befinner sig i kommandofönstret så kan man söka hjälp för ett kommando genom att skriva *help* .Man får då upp en lista på samtliga toolboxar. Skriver man *help signal* så erhåller man en lista på alla M-filer som finns i toolboxen *signal*. Skriver man *help plot* så får man information om hur kommandot plot fungerar.

Vid tilldelning av namn på variabler och kommandon så skiljer man på namn skrivna med versaler och gemener. För de kommandon som redan finns i form av M-filer så används små bokstäver (gemener) t.ex plot ej PLOT.



Redan inskrivna kommandon kan återfås dels genom att bläddra $\uparrow\downarrow$ -tangenterna dels genom att i kommandofönstret skriva variabelnamnet eller kommandot och sedan använda $\uparrow\downarrow$ -tangenterna.

Kommandona `who` och `whos` ger oss möjligheter att undersöka vilka variabler vi har definierat. `Whos` ger även information om hur mycket plats respektive variabel utnyttjar av minnet.

Här kan du även använda **show workspace**.

Kommandot **clear** rensar alla variabler. Vill man plocka bort en variabel så skriver man **clear** följt av variabelnamn. En variabel kan vara global eller lokal. Variabler som förekommer i M-filer är lokala. Man gör dessa globala med kommandot **global** plus namnet på aktuell lokal variabel.

Vill man avbryta en beräkning så används `<CTRL C>`.

Bland mer övergripande funktioner kan nämnas att man från MATLAB kan kommunicera med funktioner skrivna i C. Denna finess ger möjligheter att hämta och skicka data till och från AD/DA-kort. Man har även möjligheter att slussa data mellan MATLAB och EXCEL.

Matlabsyntax och aritmetiska uttryck.

De aritmetiska operationerna i nedan prioritets ordning

Prioritet	Operation	Tecken	Matlab	Exempel
1	Upphöjt (potens)	^	2^3	2 ³
2	Multiplikation	*	2*3	2 • 3
	Division vänster	/	3/2	3/2=1,5
	Division höger	\	3\2	3/2=0,5
3	Addition	+	2+3	2+3
	Subtraktion	-	2-3	2-3

Denna ordning kan manipuleras genom användning av parenteser kring uttryck: $4+6/8 = 4,75$ eller $(4+6)/8=1,25$

Andra skrivsätt som kan vara bra att känna till är

- **10 potenser** kan skrivas som $6.626e-34$ $6,626*10^{-34}$
- **Decimaltecken** skrivs som **.** (**punkt**). Kommatecknet har andra funktioner i MATLAB, som att separera elementen i en vektor.
- **Kommentarer** skrivs med **%-tecken** före kommandona
Semikolon ; gör att kommandot inte skrivs ut i kommando - fönstret

Variabler

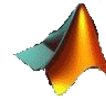
Variabler fungerar på samma sätt som i matematiken. De är en slags symbolisk ersättning för värden. I MATLAB kan variablerna heta i stort sätt vad som helst. Variabelnamn kan vara en bokstavskombination och bör då för Din egen skull vara "talande", distans i stället för bara d.

Antalet tecken kan vara upp till 19, *första tecknet måste vara* en bokstav **A-Z** eller **a-z**. Det får ingå siffror **0-9** och underscore-tecknet (**_**), men inga andra tecken som ingå å, ä eller ö.

Du bör inte använda **i** och **j** som variabel namn, eftersom de används för imaginära enheten och inte heller **pi** som är π -värdet.

Matlab skiljer på versaler och gemener, dvs **a** och **A** är ej samma variabel. Använd gemena bokstäver för kommandon!

Varje gång du definierar en variabel finns den kvar under hela tiden du använder MatLab om du inte själv tar bort den. Det går att ta bort variabler med hjälp av kommandot `Clear`.



Kommand	Förklaring
clear	Raderar alla variabler och återställer alla inbyggda variabler utom <i>eps</i>
clear namn	Raderar enbart variabeln <i>namn</i>
clear a*	Raderar alla variabler som börjar med <i>a</i>
exist('namn')	Ger svar på om det finns en variabel som heter namn
who	Listar alla variabler som finns i minnet i kommandofönstret
whos	Listar alla variabler och deras storlek, som finns i minnet i kommandofönstret

Matriser

Antag att vi har en 4x4 matris $A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$ som ska matas in.

Syntax: `A=[1 2 3 4;5 6 7 8;9 10 11 12;13 14 15 16];`

A(1,1) betecknar elementet i första raden första kolumnen alltså **1**.

A(3,4) betecknar elementet i tredje raden fjärde kolumnen alltså **12**

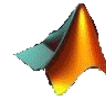
A(2:4,2:4) ger en del av matrisen *A* nämligen: $\begin{pmatrix} 6 & 7 & 8 \\ 10 & 11 & 12 \\ 14 & 15 & 16 \end{pmatrix}$.

A(:,4) ger fjärde kolumnen nämligen $\begin{pmatrix} 4 \\ 8 \\ 12 \\ 16 \end{pmatrix}$.

A(3,:) ger en radvektor innehållande elementen **[9 10 11 12]**.

Allmänt gäller att **A(:,j)** betecknar **j**:te kolumnen och **A(i,:)** anger **i**:te raden.

Kommando	Resulterande delmatris
a(i,j)	Delmatrisen till a bestående av element i rad i och kolum j .
a(:,j)	Den j :te kolumnen i a , dvs man kommer åt hela följderna av a :s rader och det j :te elementet i varje sådan rad.
a(i,:)	Den i :te raden i a .
a(:,:)	Detta är samma sak som a själv!
a(:)	Gör om formen på a till en enda lång kolumn genom att koppla ihop kolumnerna i a .
a(j:k)	Tar ut element j till och med k ur a(:) och lagrar dem i en rad.
a(:,j:k)	Delmatrisen till a bestående av kolumnerna j tom k .
a(i:k,:)	Delmatrisen till a bestående av raderna i tom k .
a(j:k, j:l)	Delmatrisen till a bestående av elementen i raderna i tom k och i kolumnerna j tom l .
a([j1 j2])	Tar ut elementen j1, j2, etc ur a(:) och lagrar dem i en rad.
a(:, [j1 j2])	Tar ut kolumn nr j1, j2, etc ur a .
a([i1 i2],:)	Tar ut rad nr i1, i2, etc ur a .
a([i1 i2], [j1 j2])	Ger delmatrisen till a bestående av elementen i raderna i1, i2, etc och kolumnerna j1, j2, etc



För att skapa speciella matriser finns särskilda kommandon, t.ex.

A=ones(M,N) ger en MxN matris av ettor.
B=zeros(M,N) ger en MxN matris av nollor.
C=eye(N) ger en NxN enhetsmatris.

För matrissräkningar gäller att:

C=A+B; ger summan av A och B (samma ordning (MxN) på A och B). $c_{ij} = a_{ij} + b_{ij}$.

C=A*B; ger produkten av A och B där A är av ordning (MxN) och B av ordning (NxP). C blir av

ordning (MxP). $c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$.

Observera att i allmänhet är **A*B ej lika med B*A** för matriser.

C=A'; ger transponatet av A. $c_{ij} = a_{ji}$.

C=A^(-1);
C=inv(A) ger inversen till matrisen A.

X=A\b löser ekvationssystemet $AX=b$.

C=det(A) ger determinanten av A.

C=eig(A) ger egenvärdena till en kvadratisk matris A.
size(A); ger storleken på (MxN) av matrisen A.

Man kan utföra matematiska operationer elementvis på matriser genom att sätta en punkt före operationen.

C=A.*B ;C=A./B; A och B måste ha samma ordning.

Visning av skillnaden med matrismultiplikation med, elementvis, och utan punkt, matrismultiplikation

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \text{ och } B = \begin{pmatrix} -1 & -2 \\ -3 & -4 \end{pmatrix}$$

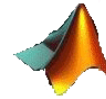
>>A*B	A.*B	2*A
-7 -10	-1 -4	2 4
-15 -22	-9 -16	6 8

Skillnaden ges av att

$$A * B = \begin{pmatrix} A11 * B11 + A12 * B21 & A11 * B12 + A12 * B22 \\ A21 * B11 + A22 * B21 & A21 * B12 + A22 * B22 \end{pmatrix}$$

men

$$A .* B = \begin{pmatrix} A11 * B11 & A12 * B12 \\ A21 * B21 & A22 * B22 \end{pmatrix}$$



Ett exempel på tillämpning av matriser är att lösa ekvationssystem:

- $X1+2*X2+3*X3=360$
- $4*X1+5*X2+6*X3=800$
- $7*X1+8*X2+0*X3=350$

$A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 0];b=[360\ 800\ 350]'$;
 $X=A\b$

$X =$
32.2222
15.5556
98.8889

Ekvationssystemet har lösningen $X1=32.2$, $X2=-15.6$ och $X3=98.9$.

Vektorer

Antag att vi vill plotta funktionen $y = f(x) = \sin(2\pi x)$ i intervallet $[0; 1]$. För att göra detta måste man:

1. definiera ett antal x -värde i intervallet $0 = x_1 < x_2 < \dots < x_n = 1$
2. beräkna funktionsvärdet för varje x -värde $y_k = f(x_k); k = 1; \dots; n$
3. rita polygonlinjen som förbinder punkterna $(x_1; y_1); \dots; (x_n; y_n)$

Ovanstående exempel visar att det är viktigt att kunna generera följder av x - och y -värden.

Den allmänna formen av kolon notationen är

$x=a:h:b$

där a är startvärdet, h steglängden och b slutvärdet. Notera att även negativa steglängder är tillåtna.

För att generera vektorer med ett bestämt antal element finns flera möjligheter. Kommandot $x=\text{linspace}(a,b,n)$

ger en vektor med n element jämnt fördelade i intervallet $[a; b]$. Till exempel är

$x=\text{linspace}(0,0.1,11)$

ekvivalent med

$x=[0.0\ 0.1\ 0.2\ 0.3\ 0.4\ 0.5\ 0.6\ 0.7\ 0.8\ 0.9\ 1.0]$

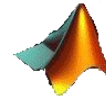
Funktionerna i MATLAB accepterar förutom tal även vektorer som argument. Om vi till exempel definiera en vektor

$x=1:1:10$

och skriver

$y=\text{sqrt}(x)$

får vi en vektor y vars element består av kvadratrötterna av heltalen från ett till tio.



Skript filer sk M-filer

Det är ofta bekvämt att skriva kommandon i en script-fil. Genom att skriva namnet på script-filen vid kommandopromtern kommer kommandona i filen att utföras i ordning. I MATLAB har script-filer alltid efternamnet (extension) .m och kallas därför M-filer.

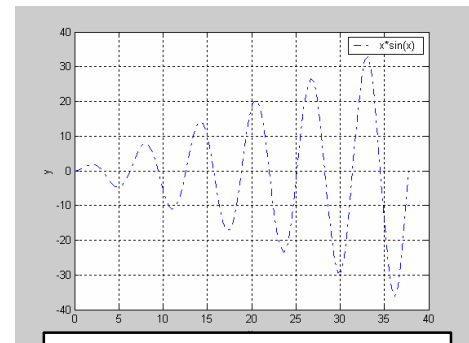
M-filer skapas i tre steg:

1. aktivera MATLABs Editor/Debugger genom att gå upp överst i MATLABs kommandofönster och klicka på figuren föreställande ett vitt pappersark
2. skriv in kommandona i editorn
3. spara filen genom att gå upp överst i editor fönstret och klicka på figuren föreställande en diskett Observera att namn på scriptfiler ej får börja med siffror. Namn får ej heller innehålla punkter annat än för att markera extension. Tex är 1a.m och uppgift1.1.m otillåtna namn på M-filer. Man ska också se till att man inte ger sin M-fil ett fördefinierat namn. Det är alltså olämpligt att döpa en M-fil till sin.m eftersom sin står för den fördefinierade standardfunktionen $\sin(x)$.

Funktionsfiler är en speciell typ av M-filer som definierar en funktion av en eller flera variabler. Som ett exempel kan vi ta funktionen $f(x) = x \sin(x)$. Om vi refererar till funktionen som fun1 så ges denna av *funktionsfilen*

```
function f=fun1(x);
f=x.*sin(x);
```

Filen skall alltid ges samma namn som funktionen. I detta fallet döper vi filen till fun1.m



Figur 3: Funktionen

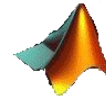
En funktion skiljer sig från ett skript genom att:

- Argument (värden) kan föras över till funktionen
- Variabler som är definierade och förändras inuti filen är lokala variabler i funktionen och påverkar inte de globala variablerna i Workspace.

```
function utdata_variabler=function_namn(indata_variabler)
% Förklarande text om vad funktionen gör och vilka indata den behöver och
% vilka utdata som den ge. Med help function_namn kan denna förklarande text ses (Testa
med Matlabs egna funktioner)
```

Deklarera en funktion med ett argument för indata (x) och ett argument för utdata (y).

```
function y = mittmean(x)
% MEAN Average or mean value.
% For vectors, MITTMEAN(X) is the mean value of the elements in X.
% For matrices, MITTMEAN(X) is a row vector containing the mean value of each column.
%
% See also MEDIAN, STD, MIN, MAX.
% Copyright © 1984-93 by The MathWorks, Inc.
```

```
[m,n] = size(x);    % get the size of x (m rows X n columns)
if m == 1          % if m equal to 1 then n is the number of values
m = n;
end
y = sum(x) / m;    % sum of x-values/number of values
```

Relations och logiska uttryck

Det finns några relation och logiska vilkor i MATLAB. Värdet på dessa uttryck blir antingen 1 (true) eller 0 (false) enligt avdelning *Urval av vanliga MATLAB kommandon* under rubrik *Logiska operatorer*

Det finns ett antal användbara relation och logiska funktioner i MATLAB, tex:

exist	Kontrollerar om variabler eller funktioner är definerade
any	Sant om något element i vektorn är sant
all	Sant om alla element av en vector är sant
find	Hittar index på icke noll element
isempty	Sant om matrisen är tom True for empty matrix
isstr	Sant det är en textsträng

In- och utdata

Fram till nu har svaret vi fått i kommandofönstret innehållit 15 tecken. Formatet på på skärmen utskriften i kommandofönstret går att ändra på till viss del med uttryck enligt avdelning *Urval av vanliga MATLAB kommandon* under rubrik *Sifferformat*

Avrundning

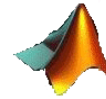
fix(x)	Avrundar x till närmaste heltal mot noll, dvs uppåt för negativa tal och nedåt för positiva
ceil(x)	Avrundar uppåt till närmaste heltal
round(x)	Avrundar till närmaste heltal
floor(x)	Avrundar nedåt till närmaste heltal

MATLAB följande elemntära indata funktioner:

input	Ger uppmaning till användaren att mata in värde i kommando fönstret
ginput	Grafisk inmatning med musen I det grafiska fönstret
pause	Väntar ett angivet antal sekunder eller på användaren svar
keyboard	Ger över kontrollen till tangentbordet
load	Tar emot värden från en fil

Input funktionen är till för att få ett numeriskt eller stäng värde från kommandofönstret:

```
langden=input('Skriv in värdet på Längden :')
strRubGraf=input('Skriv in rubriken till grafen ', 's')
```



I det första fallet accepteras endast numeriska värden med I det andra fallet går bra med vilket alfa numeriskt värde som helst.

Utdata

Förutom den grafiska utdata som grafer kan MatLab ge annan form som kan vara bra att känna till:

disp	Visar en matris eller text i kommandofönstret
format	Ger visning av olika format på siffror
save	Sparar workspace variabler Workspace- eller textfiler

Om du har en variabel $b=10.1$, och vill ha ett svar som: *Variablen $b = 10.1$*

Så måste du blanda strängar och siffror. Det går inte att skriva `disp('Variablen b=',b)` för att disp accepterar bara ett indata argument. Lösningen är att omvandla siffran b till sträng och lägga till det i slutet av en informations text:

```
b=10.1;
bstring=num2str(b);
outtxt=[' variabeln b = ' bstring];
disp(outtxt)
alternativt:
disp(['variablen b = ' num2str(b)])
```

Några användbara stäng funktioner

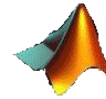
strcmp	Jämför strängar
findstr	Hitta en sträng i en annan
upper	Omvandlar till stora bokstäver
lower	Omvandlar till små bokstäver
Num2str	Omvandlar tal till sträng
str2num	Omvandlar sträng till tal
sprintf	Skriver formaterad text till en sträng

The permission string specifies the type of access you want to have

	'r' - for reading only
	'w' - for writing only
	'a' - for appending only
	'r+' - both for reading and writing

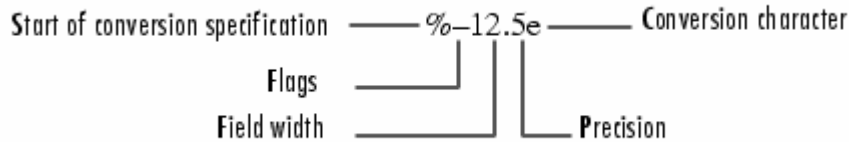
`fopen('fread.m','r');` tex öppnar file bara för läsning

n	- read at most n elements into a column vector
inf	- read at most to the end of the file
[m, n]	- read at most m n elements filling at least an m-by-n matrix, in column order; n can be inf



Ex

```
fid = fopen('exp.txt');
a = fscanf(fid,'%g %g',[2 inf]) % It has two rows now.
a = a';
fclose(fid)
```



Flags

You can control the alignment of the output using any of these optional flags.

Character	Description	Example
A minus sign (-)	Left-justifies the converted argument in its field.	%-5.2d
A plus sign (+)	Always prints a sign character (+ or -).	%+5.2d
Zero (0)	Pad with zeros rather than spaces.	%05.2d

Field Width and Precision Specifications

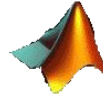
You can control the width and precision of the output by including these options in the format string.

Character	Description	Example
Fieldwidth	A digit string specifying the minimum number of digits to be printed.	%6f
Precision	A digit string including a period (.) specifying the number of digits to be printed to the right of the decimal point.	%6.2f

Conversion Characters

Conversion characters specify the notation of the output

Specifier	Description
%c	Single character
%d	Decimal notation (signed)
%e	Exponential notation (using a lowercase e as in 3.1415e+00)
%E	Exponential notation (using an uppercase E as in 3.1415E+00)
%f	Fixed-point notation
%g	The more compact of %e or %f, as defined in [2]. Insignificant zeros do not print.
%G	Same as %g, but using an uppercase E
%i	Decimal notation (signed)
%o	Octal notation (unsigned)
%s	String of characters
%u	Decimal notation (unsigned)
%x	Hexadecimal notation (using lowercase letters a-f)
%X	Hexadecimal notation (using uppercase letters A-F)



Escape Characters

This table lists the escape character sequences you use to specify non-printing characters in a format specification.

Character	Description
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\\	Backslash
\' or \"	(two single quotes)
%%	Percent character

Ex.

```
fprintf(fid,'%6.2f %12.8f\n',y);
```

%6.2f betyder 6 siffror där det finns 2 siffror till höger om decimaltecknet med fast decimal tecken

%12.8f\n' betyder 12 siffror där det finns 8 siffror till höger om decimaltecknet med fast decimal tecken och sedan skall det bli ny rad

Utskrift på skärm kan göras med *disp* som tidigare visat

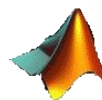
```
for i=1:1:n
    disp(['            '])
    disp(['Punkt nr ' num2str(i)])
    disp(['har höjden ' num2str(X(i,1))])
    disp(['Grundmedelfelet är ' num2str(Qxx(i,1))])
end
```

eller lite elegantare genom att först använda funktionen *sprintf* att generera en textsträng och sätta in denna i funktionen *disp*

```
for i=1:1:n
    Text=sprintf('För punkt %2.0f är höjden %5.3f meter med grundmedelfel på %5.15f ',i,X(i,1),Qxx(i,1));
    disp(['            '])
    disp([' ' Text])
end
```

Förhoppningsvis har Du insett att det är "de små små detaljerna som gör 'et'".

Det ligger lite i att det är "enkelt att lära men tar lång tid att behärska".



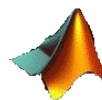
Urval av vanliga MATLAB kommandon

Tabellerna som följer utgör ett mycket litet urval av kommandon i MATLAB. För övrigt hänvisas till MATLABs **help**.

Anmärkning; i texten nedan förekommer begreppet vektor ofta. En vektor är en matris med endast en rad (radvektor) eller en matris med endast en kolonn (kolonnvektor). Sådana vektorer används väldigt ofta. Vektorn $vikt=[67\ 70\ 83\ 65\ 54\ 45\ 83]$ är exempelvis en vektor som innehåller resultatet av en mätning av några personers kroppsvikt. MATLAB kan med enkla kommandon räkna ut medelvikt, summavikt och en massa annat med denna typ av objekt.

Elementära matematiska funktioner

Matematisk beteckning	Skrivsätt i MATLAB	Kommentar
$\sin x$	<code>sin(x)</code>	Alla vinklar räknas i radianer
$\cos x$	<code>cos(x)</code>	
$\tan x$	<code>tan(x)</code>	
$\arcsin x$	<code>asin(x)</code>	
$\arccos x$	<code>acos(x)</code>	
Polära koordinater	<code>cart2pol</code>	<code>[theta,rho]=cart2pol(x,y)</code>
Vinkelräta koordinater	<code>pol2cart</code>	<code>[x,y]=pol2cart(theta,rho)</code>
$\arctan x$	<code>atan(x)</code>	
e^x	<code>exp(x)</code>	
$\ln x$	<code>log(x)</code>	
$\log x$	<code>log10(x)</code>	
\sqrt{x}	<code>sqrt(x)</code>	
$x * y$	<code>x.*y</code>	"Normal multiplikation" Multiplicerar vart och ett av elementen i x med motsvarande element i y. x och y måste vara av samma storlek försåvitt inte en av dem är en 1 gånger 1 matris (d.v.s. ett tal)
x/y	<code>x./y</code>	"Normal division". Se rutan ovan.
x^y	<code>x.^y</code>	upphöjer vart och ett av elementen i x till y
$ x $	<code>abs(x)</code>	
$n!$	<code>gamma(n-1)</code> eller <code>prod(1:n)</code>	



Komplexa funktioner

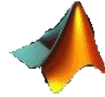
Matematisk beteckning	Skrivsätt i MATLAB	Kommentar
Re z	<code>real(z)</code>	
Im z	<code>imag(z)</code>	
$ z $	<code>abs(z)</code>	
arg z	<code>angle(z)</code>	
\bar{z}	<code>conj(z)</code>	

Matris och vektorfunktioner

Matematisk beteckning	Skrivsätt i MATLAB	Kommentar
A^t	<code>A'</code>	Gör rader till kolonner och tvärtom. Apostrofen på *-knappen
$u \cdot v$	<code>dot(u,v)</code>	Skalarprodukt. Kan även skrivas <code>u'*v</code> (eller <code>u*v'</code> , om u och v är radvektorer).
$ u $	<code>norm(u)</code>	Längden av en vektor. Använd inte <code>abs(u)</code> , det beräknar beloppet av vart och ett av talen i vektorn

Speciella matriser och vektorer

Skrivsätt i MATLAB	Förklaring
<code>eye(n)</code>	Bildar en enhetsmatris av typ $n \times n$. En enhetsmatris har ettor i diagonalen och i övriga positioner nollor.
<code>zeros(n)</code> respektive <code>zeros(n,m)</code>	Bildar en matris av typ $n \times n$ respektive av typ $n \times m$ som bara innehåller nollor.
<code>ones(n)</code> respektive <code>ones(n,m)</code>	Bildar en matris av typ $n \times n$ respektive av typ $n \times m$ som bara innehåller ettor.
<code>rand(n)</code> respektive <code>rand(n,m)</code>	Bildar en matris av typ $n \times n$ respektive av typ $n \times m$ med slumpstal rektangelfördelade i intervallet $[0,1]$.
<code>randn(n)</code> respektive <code>randn(n,m)</code>	Bildar en matris av typ $n \times n$ respektive av typ $n \times m$ med slumpstal normalfördelade med väntevärde 0 och standardavvikelse 1.
<code>start:stop</code>	Bildar en heltalsvektor med första elementet <i>start</i> och sista elementet <i>stop</i> , och alla heltal där emellan.
<code>start:steg:stopp</code>	Bildar en vektor med första elementet <i>start</i> följt av element med avstånden <i>steg</i> ifrån varandra, fram till sista elementet <i>stopp</i> .
<code>linspace(a,b,n)</code>	Bildar en vektor med n element som är jämt fördelade mellan a och b med början på talet a .



Polynomfunktioner

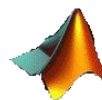
Beteckning i MATLAB	Förklaring
<code>poly(r)</code>	Skapar det enklaste polynomet med rötterna i vektorn r .
<code>roots(p)</code>	Beräknar rötterna till polynomet p (p är en vektor med polynomets koefficienter)
<code>polyder(p)</code>	Deriverar polynomet p . Svaret är ett polynom med gradtalet ett steg lägre än gradtalet för p .
<code>polyval(p,x)</code>	Beräknar värdet av polynomet p för värdet (värdena) x .
<code>polyfit(x,y,n)</code>	Anpassar ett polynom av grad n till samhörande värden på x och y .

Statistiska funktioner

Beteckning i MATLAB	Förklaring
<code>mean(x)</code>	Beräknar medelvärdet av talen i vektorn x .
<code>std(x)</code>	Beräknar standardavvikelsen av talen i vektorn x .

Grafik

Beteckning i MATLAB	Förklaring
<code>figure(k)</code>	skapar ett nytt figurfönster med löpnummer k
<code>plot(x,y)</code>	Ritar en graf av talen i vektorn y som funktion av talen i vektorn x och förbinder punkterna med räta linjer. Kommandot <code>plot</code> innehåller väldigt många möjligheter. Se vidare help <code>plot</code> .
<code>axis([x1 x2 y1 y2])</code>	Väljer område av en graf som ska visas. x1 x2 y1 y2 skall vara värden (tal)
<code>xlabel(txt)</code>	Skriver teckensträngen txt vid x-axeln. I det fall att man inte har någon variabel txt kan man skriva <code>xlabel('detta är texten')</code>
<code>ylabel(txt)</code>	se ovan fast på y-axeln
<code>title(txt)</code>	skriver rubrik till bilden
<code>text(x,y,txt)</code>	skriver teckensträngen txt vid punkten (x,y) i bilden.
<code>hist(x)</code>	Ritar ett histogram över talen i vektorn x .
<code>hold on</code>	Håller kvar en figur så att flera kan ritas tillsammans.
<code>hold off</code>	"Släpper" en figur så att nästa ritas i ett tomt fönster.
<code>grid</code>	Ritar rutnät
<code>clf</code>	Rensar figurfönstret och återställer allting.



Vektorberäkningar

Beteckning i MATLAB	Förklaring
<code>sum(x)</code>	Beräknar summan av alla talen i vektorn x .
<code>prod(x)</code>	Beräknar produkten av alla talen i vektorn x .
<code>length(x)</code>	Beräknar antalet tal i vektorn x .

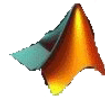
Kommunikation mellan kommandofönstret och program (m-fil)

Beteckning i MATLAB	Förklaring
<code>disp(txt)</code>	Skriver ut teckensträngen txt .
<code>a=input(txt)</code>	Skriver teckensträngen txt och förväntar sig ett värde på talet a
<code>a=input(txt,'s')</code>	Samma som ovan men a är i detta fall en teckensträng.
<code>eval(string)</code>	Tolkar teckensträngen string som ett MATLAB-kommando och utför detta kommando (om det går).
<code>num2str(x)</code>	Konverterar talet x till en teckensträng. Användbart när man vill få värden utskrivna i t.ex. en bildtext

Logiska operatörer

används vid tester. `A==B` ger värdet 1 (sant) om A är lika med B, annars 0 (falskt).

Beteckning i MATLAB	Förklaring
<code>==</code>	Lika med
<code>></code>	Större än
<code><</code>	Mindre än
<code>>=</code>	Större än eller lika med
<code><=</code>	Mindre än eller lika med
<code>~=</code>	Icke lika med
<code>&</code>	och
<code> </code>	eller
<code>~</code>	icke



Loopar och tester

Beteckning i MATLAB	Förklaring
<code>for i=start:step:stop command end</code>	Utför upprepade gånger de kommandon som står mellan <code>for</code> och <code>end</code> . Variabeln <code>i</code> räknas upp med början på ett startvärde så att <code>i=(förra värdet på i)+step</code> . Loopen avbryts då <code>i</code> är lika med eller större än stopvärdet.
<code>while statement command end</code>	Utför de kommandon som står mellan <code>while</code> och <code>end</code> så länge <code>statement</code> är sant. <code>Statement</code> är ett påstående av typ <code>a<7</code> . Loopen pågår så länge <code>a</code> är mindre än 7. Variabeln <code>a</code> bör ändras inuti loopen
<code>if statement1 command1 elseif statement2 command2 else command3 end</code>	Flervalsfråga. Om påstående1 är sant utförs efterföljande kommandon. Om påstående1 är falskt hoppas dessa kommandon över. På samma sätt testas påstående2. Om varken påstående1 eller påstående2 är sanna utförs kommandona under
<code>break</code>	Avbryter en loop. Bra att använda inuti en while-loop tillsammans med en räknare (<code>a=a+1</code>) och en IF-sats; <pre>for i=1:1:333333 a=a+1; if a>2000; break end end end</pre> På detta vis undviker man "evighets-loopar".

Litet kom ihåg

Du kan avbryta en m-fil genom att samtidigt trycka <Ctrl> c .

Använd piltangenterna i för att plocka fram gamla kommandon i kommandofönstret.

Sifferformat

Syntax

`format`

`format type`

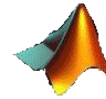
`format('type')`

Description

MATLAB performs all computations in double precision. Use the `format` function to control the output format of the numeric values displayed in the Command Window. The `format` function affects only how numbers are displayed, not how MATLAB computes or saves them. The specified format applies only to the current session.

To maintain a format across sessions, use MATLAB preferences.

`format` by itself, changes the output format to the default type, `short`, which is 5-digit scaled, fixed-point values.



format type changes the format to the specified type. The table below describes the allowable values for type. To see the current type file, use `get(0,'Format')`, or for compact versus loose, use `get(0,'FormatSpacing')`.

Value for type	Result	Example
+	+, -, blank	+
bank	Fixed dollars and cents	3.14
compact	Suppresses excess line feeds to show more output in a single screen. Contrast with loose.	theta = pi/2 theta= 1.5708
hex	Hexadecimal	400921fb54442d18
long	15-digit scaled fixed point	3.14159265358979
long e	15-digit floating point	3.141592653589793e+00
long g	Best of 15-digit fixed or floating point	3.14159265358979
loose	Adds linefeeds to make output more readable. Contrast with compact.	theta = pi/2 theta= 1.5708
rat	Ratio of small integers	355/113
short	5-digit scaled fixed point	3.1416
short e	5-digit floating point	3.1416e+00
short g	Best of 5-digit fixed or floating point	3.1416

Tät eller lös utskrift på skärmen

`FormatSpacing` compact | {loose}

Output format spacing (see also `format` command).

- compact -- Suppress extra line feeds for more compact display.
- loose -- Display extra line feeds for a more readable display.

Syntax **format compact**