# Introduktion till MATLAB

Jönsson: 1.1-1.7, 2.1-2.4, 3.1-3.5

#### 1 Inledning

MATLAB är både en interaktiv matematikmiljö och ett programspråk, som används på de flesta tekniska högskolor runt om i världen, och har stor användning även inom industrin.

En av styrkorna med MATLAB är att systemet är utbyggbart med bibliotek eller verktygslådor, toolboxes, för olika tillämpningsområden.

Ni kommer använda MATLAB i många kurser i utbildningen. Bl.a. kommer ni göra laborationsgifter i matematikkurserna.

#### 2 Starta Matlab

Vid en WINDOWS-dator startas MATLAB genom att man går in under Start-symbolen och väljer All Programs och därunder MATLAB.

Vid en LINUX-dator går man in under Applications och väljer Chalmers Applications och Matlab.



Man avslutar MATLAB genom att gå in under File och välja Exit MATLAB (längst ned i menyn).

MATLAB-fönstret man får upp kallas **Desktop** och dess utseende eller uppdelning kallas **Desktop** Layout. Den standard **Desktop** Layout ni får då ni startar MATLAB första gången ser lite annorlunda ut än på bilden. Vi kommer som en övning göra en layout som ser ut ungefär som på bilden och som är lämplig för det fortsatta arbetet med MATLAB.

## 3 En enkel beräkning och några grafer

Här följer några exempel så att vi snabbt kommer igång och ser lite resultat. Följ gärna med vid datorn och knappa in efter hand i **Command Window** och se vad som händer.

**Exempel 1.** Beräkna volymen av ett klot med radien r = 3 cm. Volymen ges av  $V = \frac{4}{3}\pi r^3$ .

Först inför vi en variabel r, för radien, som vi ger värdet 3.

>> r=3

Ett variabelnamn skall börja med en bokstav (a-z, A-Z), därefter får vi ha bokstäver (a-z, A-Z), siffror (0-9) och understrykningstecken  $(\_)$ . MATLAB skiljer på stora och små bokstäver.

Den s.k. promptern >> skriver vi inte. Tecknet finns i Command Window på raden där vi skall skriva vårt kommando och visar att MATLAB är redo.

Därefter beräknar vi volymen enligt formeln och låter variabel<br/>n ${\tt V}$ få detta värde.

>> V=4/3\*pi\*r^3

Konstanten pi är en approximation av den matematiska konstanten  $\pi$ .



**Uppgift 1.** Beräkna arean av en cirkelskiva med radien r = 4 cm. Arean ges av  $A = \pi r^2$ .

**Exempel 2.** Rita grafen av  $f(x) = \sin(x) + 0.3 \sin(4x)$  för  $0 \le x \le 4\pi$ .

Först gör vi en lista eller radvektor x av x-värden mellan 0 och  $4\pi$ , med

>> x=0:0.1:4\*pi;

Närmare bestämt får vi värdena 0, 0.1, 0.2, 0.3,  $\cdots$ , 12.5, dvs. värden med start i 0, steget 0.1 och slut så nära upp mot  $4\pi$  som möjligt.

Därefter gör vi en lista eller radvektor **f** med f(x)-värden för varje x-värde i **x** och ritar upp grafen med **plot**.

```
>> f=sin(x)+0.3*sin(4*x);
>> plot(x,f)
```



Om vi hade inte skrivit ett semikolon (;) sist i uttrycket för x och f, hade alla värden skrivits ut på skärmen och det vill vi nog inte.

Det skulle fungera lika bra att räkna ut funktionsvärdena på plats i plot kommandot

```
>> x=0:0.1:4*pi;
>> plot(x,sin(x)+0.3*sin(4*x))
```

**Uppgift 2.** Rita grafen till  $f(x) = \sin(x) + 0.3 \sin(5x)$  över samma intervall.

Vi kan använda uppåtpil ( $\uparrow$ ) för att komma till ett kommando vi givit tidigare. Om vi vill kan vi gå längs raden med vänster- och högerpilarna ( $\leftarrow$ ), ( $\rightarrow$ ) och redigera kommandot. När kommandot ser ut som vi vill trycker vi på enter ( $\leftarrow$ ).

Vill vi rensa Command Window så ger vi kommandot clc och vill vi rensa Figure 1 ger vi kommandot clf.

**Exempel 3.** Rita graferna av  $f(x) = \sin(x)$  och  $g(x) = \sin(4x)$  för  $0 \le x \le 2\pi$ . Sätt rubrik och text på axlarna.

Vi använder funktionen linspace för att få 100 punkter jämnt fördelade mellan 0 och  $2\pi$ , då blir graferna jämna och snygga.

```
>> x=linspace(0,2*pi);
>> f=sin(x);
>> g=sin(4*x);
```

Vi ritar båda graferna samtidigt med plot, både paret x, f och paret x, g.

>> plot(x,f,'green',x,g,'red')



För att skilja graferna åt gjorde vi sin(x)-grafen grön 'green' och sin(4x)-grafen röd 'red'.

Vi sätter text på axlarna och rubrik samt lägger på ett rutnät med

```
>> xlabel('x')
>> ylabel('y')
>> title('sin(x) och sin(4x)')
>> grid on
```

Vill vi ta bort rutnätet, gör vi det med grid off.

Texterna inom apostrofer (' ') är s.k. textsträngar. Exempelvis är 'green', 'x' och 'sin(x) och sin(4x)' textsträngar.

#### 4 Något om matriser

En matris är som bekant ett rektangulärt talschema

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

Matrisen ovan har m rader och n kolonner, vi säger att den är av typ  $m \times n$ . Ett matriselement i rad nr i, kolonn nr j tecknas  $a_{ij}$ , där i är radindex och j är kolonnindex. I MATLAB skrivs detta A(i,j) och size(A) ger matrisens typ.

En matris av typ  $m \times 1$  kallas kolonnmatris (kolonnvektor) och en matris av typ  $1 \times n$  kallas radmatris (radvektor):

$$\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix}$$

Element nr i ges i MATLAB av b(i), c(i) och antalet element ges av length(b), length(c). Som exempel tar vi

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 0 & 2 & 4 & 6 & 8 \end{bmatrix}$$

Vi skriver in detta i MATLAB enligt

>> A=[1 4 7 10; 2 5 8 11; 3 6 9 12] >> b=[1; 3; 5] >> c=[0 2 4 6 8]

**Uppgift 3.** Skriv in matriserna i MATLAB och skriv sedan ut matriselementen  $a_{23}$ ,  $b_2$ ,  $c_3$ . Prova size och length. Ändra  $a_{23}$  genom att skriva A(2,3)=15.

### 5 Linjärt ekvationssystem

Linjära ekvationssystem kan vi lösa med MATLAB om vi först skriver dem på matrisform. Vi tar som exempel: Ekvationssystemet

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 14 \\ 3x_1 + 2x_2 + x_3 = 10 \\ 7x_1 + 8x_2 = 23 \end{cases}$$

skrivs på matrisform

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix}$$

dvs.

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \text{med } \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{och } \mathbf{b} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix}$$

Med backslash-kommandot (\) alternativt kommandot <br/>  ${\tt rref}$  (row-reduced-echelon form) löser vi systemet<br/>, ${\bf A}{\bf x}={\bf b}$ 

>> x=A\b >> rref([A b])

I det första fallet fungerar det bra om lösningen är entydig men sämre om det finns fria variabler eller inga lösningar alls. I det andra fallet reducerar MATLAB den utökade matrisen  $[\mathbf{A} \ \mathbf{b}]$  till reducerad trappstegsform.

**Uppgift 4.** Skriv följande ekvationssystem på matrisform och lös dem sedan med  $\$  respektive **rref**. Beskriv också hur högerledet beror av kolonnerna i koefficientmatrisen.

$\int x_1 + 5x_2 + 9x_3 = 29$	$\int x_1 + x_2 + 3x_3 + 4x_4 = 2$
$2x_1 + 5x_3 = 26$	$-2x_1 + 2x_2 + 2x_3 = -4$
$3x_1 + 7x_2 + 11x_3 = 39$	$ x_1 + x_2 + 2x_3 + 3x_4 = 1 $
	$x_1 - x_2 - 2x_3 - x_4 = 1$

#### 6 Script

För att slippa skriva om sina kommandon, eller bläddra med uppåt- och nedåtpilar  $(\uparrow)$ ,  $(\downarrow)$  eller i kommandofönstrets historik, så brukar man oftast skriva sin kod i en script.

Vi gör en script som ritar graferna från exempel 3.



En script är en textfil som innehåller det man skulle kunna skriva direkt vid promptern >> i Command Window, och som utförs i MATLAB genom att man ger textfilens namn som kommando. För att MATLAB skall hitta filen, förutsätter det att katalogen där filen ligger är aktuell katalog.

Man kan byta katalog med kommandot cd i Command Window, klicka sig fram i Current Folder eller använda Browse for folder i verktygsfältet i Desktop.

Utanför MATLAB får namnet på en script tillägget .m för att skilja den från andra filer.

MATLAB har en inbyggd editor som är det bästa verktyget att göra en script med. Om man inte redan har editorn uppe i Desktop så startas den genom att gå till File, sedan New och välja Script. Editorn markerar koden med olika färger för att visa vad som är kommentarer, nyckelord, textsträngar, etc. (Kommentarer inleds med procenttecken.) Spara kan vi göra under File och köra under Debug. Enklast är dock att trycka på 🔊 som finns i verktygsfältet. Då sparas vår script och utförs som om vi gav den som ett kommando. Alla utskrifter från programmet skrivs i Command Window, liksom alla felmeddelanden.

Om filen ligger i en annan katalog än den aktuella, så får man upp en fråga om att byta till den katalogen:



Välj Change Directory så byter MATLAB katalog. Den som är intresserad, kan titta i Helpdesk för att få reda på vad Add to Path betyder, mer om Helpdesk senare.

Vi får upp samma grafer som tidigare.



Vi kommer i fortsättningen ofta att kalla en script för en skriptfil.

Editor i MATLAB har något som kallas Cell Mode (cell-läge). Skriver man en kommentar som börjar med två procent-tecken, så avgränsar det en cell. Poängen är att man kan exekvera koden från en cell, istället för hela filen. På så sätt kan man dela upp en stor kommanofil (för ett helt övnings-tillfälle) i flera delar (varje deluppgift). När man skapar en sådan kommentar, så lägger MATLAB ut en informationsruta med ett par länkar. Den som vill kan klicka på dem för att se en liten film om hur man kan använda cell-läget.

I cell-läge kan man evaluera aktuell cell genom att klicka på 🛅 , evaluera aktuell cell och gå till nästa genom att klicka på 🖼 . Samtliga val finns vid Cell under Desktop i verktygsfältet.

### 7 Lite programmering

I MATLAB finns repetitions- och villkorssatser som påminner om motsvarande i programspråk som C och Java.

Vi nöjer oss för tillfället med att se på en repetitionssats, en **for**-sats, som vi använder för att beräkna en summa i följande exempel.

**Exempel 4.** Beräkna summan  $s = 3 + 4 + 5 + \dots + 52$ 

Vi gör en script med programkoden

```
s=0;
for i=3:52
      s=s+i;
end
                                              MATLAB 7.11.0 (R2010b)
                                                                                                         _ = ×
         File Edit Text Go Cell Tools Debug Parallel Desktop Window Help
         : 🎦 逽 👗 🐂 🛱 🤊 🍽 🍓 🗊 🖹 🥝 Current Folde<u>r</u>: /chalmers/users/jacques/kurser/Matlab 🛡 ... 🖻
          Shortcuts 🖪 How to Add 💽 What's New
         🛃 Figures - Figure 1
                                                  X 5 🗆 🕂
                                                            📝 Editor – / chalmers/users/jacques/kurser/Matla... 📲 🗖 🔻 🛪
          🗋 😂 🛃 🎍 🔍 🔍 🔏 🔹 👋 🖽 🖽 🖽 🗗 🗗 🔹 🗙
                                                             🗋 💼 👗 🐂 🛍 🤊 (° 🗐 • 💌 - - » 🗆 🔹 🗡
                                                              * - 1.0 +
                                                                                ÷ 1.1 × % 👯 🕕
                                                             1
                                                                    % Script Summera
                                                             2
                                                             3 4
                                                                    % Beräknar summan 3 + 4 + ... + 52
                                                             5 -
                                                                    s=0;
                                                                               % Sätter summan till 0
                                                                  □ for i=3:52 % För i=3, i=4, ..., i=52
                                                             6 -
                                                             7 -
                                                                        s=s+i; %
                                                                                     addera i till summan
                                                             8 -
                                                                   - end
                                                                    disp('Summan är') % Skriv ut texten och
                                                             9 -
                                                             10 -
                                                                    disp(s)
                                                                                      % värdet av summan
                                                              RitaGrafer.m × Summera.m ×
                                                            Command Window
                                                                                                       X 5 D It
         Current Folder
                                                  + D 7 X
                                                              Summan är
                                                                      1375
          🛅 « users 🕨 jacques 🕨 kurser 🕨 Matlab
                                                - 🔎 🖻 🌣-
                                                            fx >>
              Name 4
             RitaGrafer.m
           Summera.m
         Summera.m (MATLAB Script)
                                                         ~
         ▲ <u>Start</u>
                                                                   script
                                                                                            Ln 1
                                                                                                   Col 1
```

Vi skriver lämpliga kommentarer (grön text) i programkoden och gör lämplig utskrift, först textsträngen Summan är och sedan summans värde.

I matematik skriver man gärna summan  $3 + 4 + 5 + \dots + 52$  med beteckningen

$$\sum_{i=3}^{52} i$$

Uppgift 5. Skriv en script som beräknar summan

$$\sum_{i=1}^{5} i^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2$$

### 8 Function

Det finns flera olika sätt att göra egna funktioner i MATLAB. Om funktionen innehåller flera uttryck eller satser måste man göra en function, dvs. skapa en textfil med funktionsbeskrivningen. Består funktionen av ett enda uttryck så han vi göra ett s.k. funktionshandtag (function handle) eller en s.k. anonym funktion.

För större program kan man även vilja använda andra sätt att skriva funktioner, exempelvis underfunktioner (subfunction) eller nästlade funktioner (nested function), men vi lämnar det så länge.

En function är en textfil med samma namn som funktionen och som inleds med en funktionsdeklaration. I fortsättningen kommer vi ofta kalla en function för en *funktionsfil*.

**Exempel 5.** Vi vill hitta ett nollställe till funktionen  $f(x) = x^3 - \cos(x)$ .

Det finns en funktion fzero i MATLAB som hittar nollställen. För att använda fzero måste vi beskriva vår funktion och det gör vi som en function, som vi skriver in i editorn enligt

```
function y=min_fun(x)
y=x.^3-cos(x);
```

där y är funktionens värde (utdata), x är funktionens argument (indata) och min\_fun är funktionens namn.



Vi ritar grafen och använder fzero direkt i Command Window

```
>> x=linspace(-1.5,1.5);
```

```
>> y=min_fun(x);
```

```
>> plot(x,y)
```

```
>> grid on
```

Vi ser att vi har ett nollställe när<br/>ax=1och låter fzero söka nollstället genom

Med **@min\_fun** talar vi om för **fzero** vilken funktion den skall arbeta med.

Alternativt använder vi en skriptfil, vilket vi vanligtvis kommer göra



Lägg märke till att vi använde cell-läge i skriptfilen.

Råkar man försöka köra en funktionsfil, så får man ett felmeddelande av typen Input argument x is undefined, eftersom inget argument gavs.

Vi kan också använda ett funktionshandtag enligt

men det går bara på om funktionen kan beskrivas med en enda beräkningssats.

Exempel 6. Vi vill beräkna och rita lösningen till differentialekvationen

$$\begin{cases} u' = t\cos(t) + \sin(4t)u, \ 0 \le t \le 30\\ u(0) = 1 \end{cases}$$

Vi beskriver högerledet i differentialekvationen med en funktion och sedan beräknar vi lösningen med funktionen <code>ode45</code> enligt



Lägg märke till hur vi anger intervallet för t och hur vi ger det s.k. begynnelsevärdet u(0) = 1.

## 9 Desktop Layout

För att handledning och redovisning skall fungera effektivt kräver vi att all redovisning görs via en sammanhållande skriptfil tillsammans med nödvändiga funktionsfiler (från en skriptfil anropas t.ex. funktionsfiler som behövs för att lösa uppgiften). Skriptfilen som används för redovisning bör vara lämpligt uppdelad med hjälp av t.ex. kommandot **pause** eller alternativt vara i cell-läge (Cell Mode).

Vi käver också att ni har en MATLAB desktop layout av ett slag som visas i figuren i exempel 2. Man åstadkommer detta genom att "docka" in MATLAB-editorn respektive figurfönstret och sedan "dra" dem till rätt plats (om det behövs!). Att "docka" in eller ut ett MATLAB-fönster görs med de små pilar som finns uppe till höger i fönstren (strax intill "krysset").

Fördelarna med detta är att man får en bra (översiktlig och effektiv) interaktiv miljö för att utveckla program och för tolkning av resultat. En bieffekt blir dessutom att både handledning och redovisningar blir effektivare.

**Uppgift 6.** Gör en desktop layout som ser ut ungefär som den i exempel 2. Spara denna layout med ett lämpligt namn, vilket görs genom att välja **Save Layout** ... under **Desktop** i verktygsfältet. Denna layout skall sedan användas vid redovisningar.

## $10 \quad \text{Helpdesk} \ i \ \mathrm{MATLAB}$

Tryck på 🥙 i verktygsfältet, eller på MATLAB Help under Window, och Help Navigator öppnas.



Vi ser den stora uppsättningen av verktygslådor, för olika tillämpningsområden, som följer med. Här ser vi hjäptexten för funktionen **fzero**. Vi har skrivit **fzero** i sökrutan och tryckt på enter.



Läs gärna lite i texten och titta tillbaka på exempel 5 där vi använde **fzero**, och leta gärna upp hjälptexten för **ode45**.

Man kan söka sig fram för att hitta referenssidor (hjälptexter) för olika kommandon och funktioner. Nästan alla inbyggda kommandon och funktioner har en referenssida.

Man kan också titta på dessa hjälptexter med kommandon som ges i Command Window: help som ger texten i Command Window och doc som plockar fram aktuell referenssida i webläsaren. Texten är ungefär samma, men för vissa kommandon (speciellt de för grafik) så innehåller docsidan bilder, vilket kan vara till hjälp, medan help enbart visar ren text.

Det är viktigt att lära sig att läsa dokumentationen. Den är inte skriven för att lära ut till nybörjare hur man löser ett problem med MATLAB, utan för att visa den något vane användaren exakt hur en funktion eller ett kommando används. Det är inte lättläst, och man måste lära sig att plocka fram den informationen som är av intresse för tillfället, dvs. man måste lära sig att "skumma" texterna.