

Mer om funktioner och grafik i MATLAB

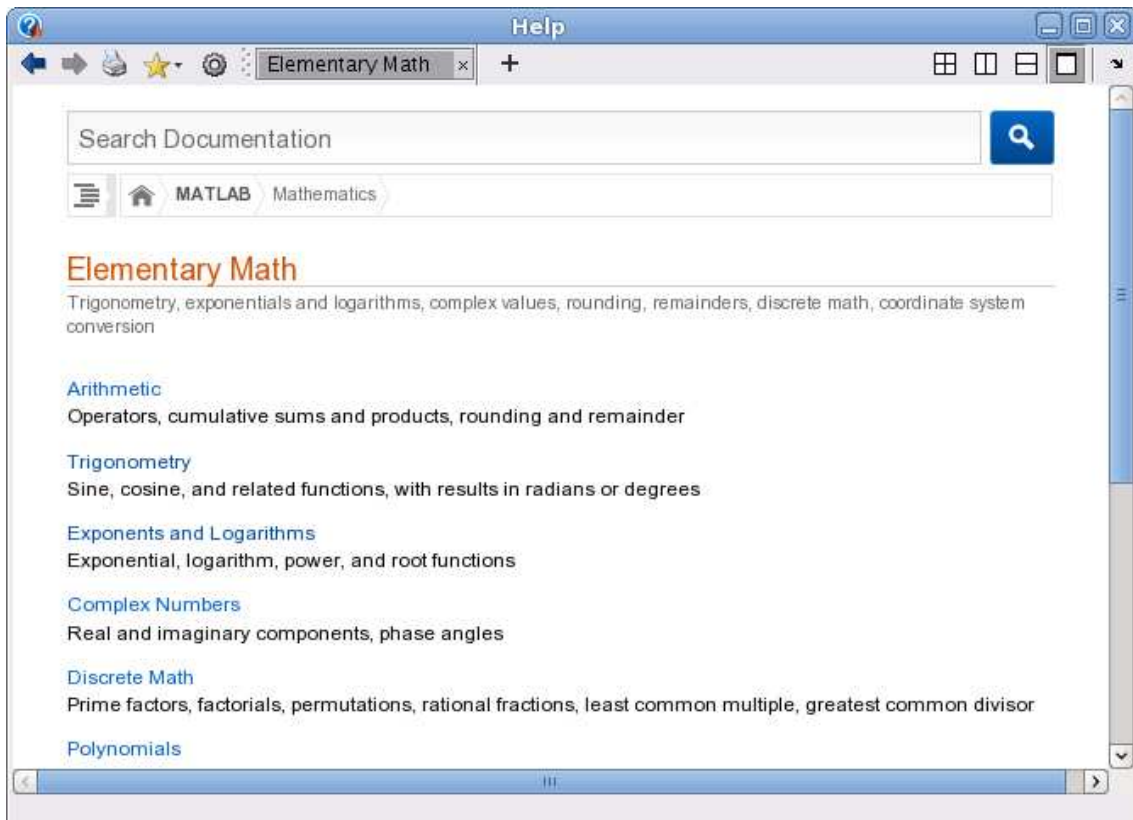
1 Inledning

Först skall vi se lite på funktioner som redan finns i MATLAB, (elementära) matematiska funktioner som sinus och cosinus samt funktioner för att bilda och operera på vektorer och matriser. Sedan ser vi lite på grafitrning och annan grafik för att avslutningvis se hur man definerar egna funktioner.

2 Inbyggda funktioner

2.1 Elementära funktioner

Vi letar upp hjälptexterna för elementära eller matematiska funktioner i **Help** genom att successivt öppna MATLAB, Mathematics och sedan Elementary Math.



Vi ser att funktionerna är grupperade, t.ex. en grupp med trigonometriska funktioner och en grupp med exponent- och logaritmfunktioner.

Funktioner som exempelvis sinus och cosinus, kan operera både på enskilda tal och på matriser. Man får som resultat en matris av samma storlek, vars element är funktionsvärdet av respektive element i argumentet.

Som exempel tar vi

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

som vi skriver in i MATLAB enligt

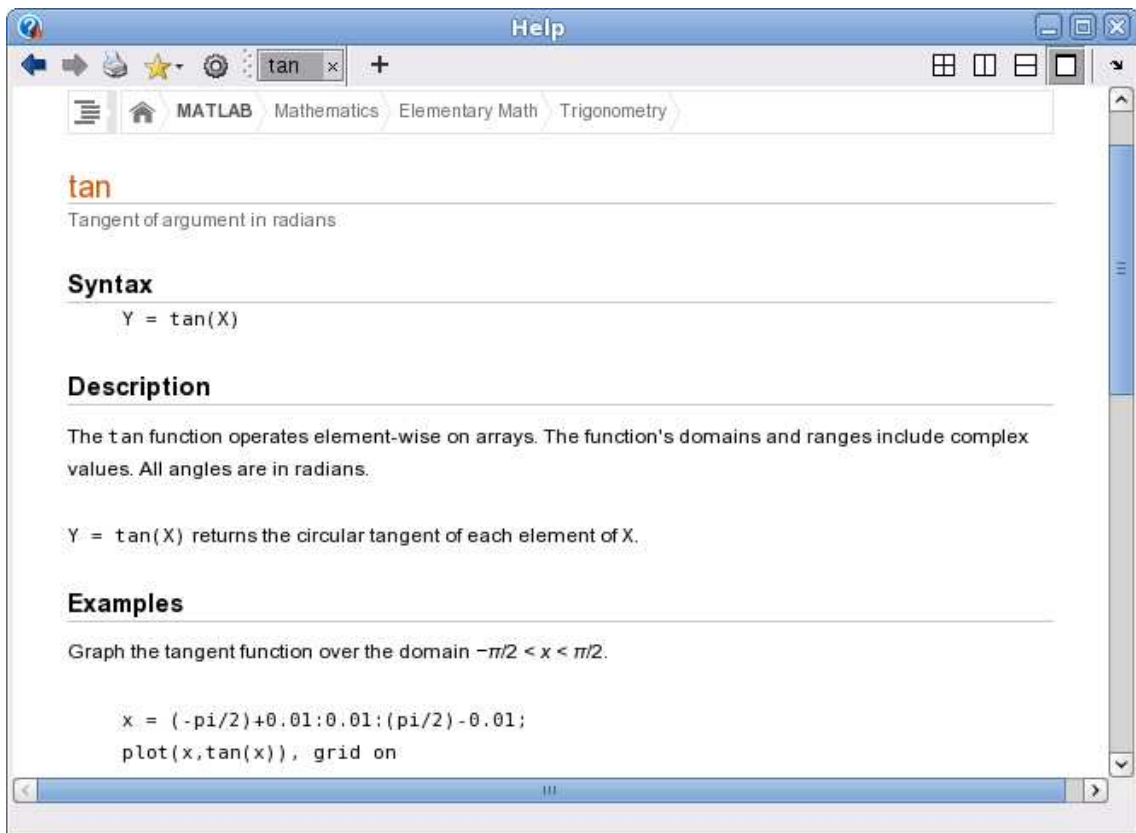
```
>> A=[1 4 7 10; 2 5 8 11; 3 6 9 12]
>> c=[0 2 4 6 8]
```

Nu beräknar vi sinus av vektorn \mathbf{c} och matrisen \mathbf{A} med

```
>> v=sin(c)
v =
    0    0.9093   -0.7568   -0.2794    0.9894
```

```
>> V=sin(A)
V =
    0.8415   -0.7568    0.6570   -0.5440
    0.9093   -0.9589    0.9894   -1.0000
    0.1411   -0.2794    0.4121   -0.5366
```

Vi söker på `tan`, dvs. tangensfunktionen, och ser på hjälptexten.



Uppgift 1. Leta upp hjälptexten du ser i figuren och rita upp tangensfunktionen enligt exemplet. Gör en skriptfil och använd cell-läge så att ni kan bygga på med kommande övningsuppgifter.

2.2 Matris- och vektorfunktioner

Vi har redan sett de inbyggda funktionerna `length` och `size`. Som exempel använder vi vektorn \mathbf{c} och matrisen \mathbf{A} från förra avsnittet. Antal element i vektorn \mathbf{c} ges av

```
>> l=length(c)
l =
    5
```

och antalet rader och kolonner \mathbf{A} fås med

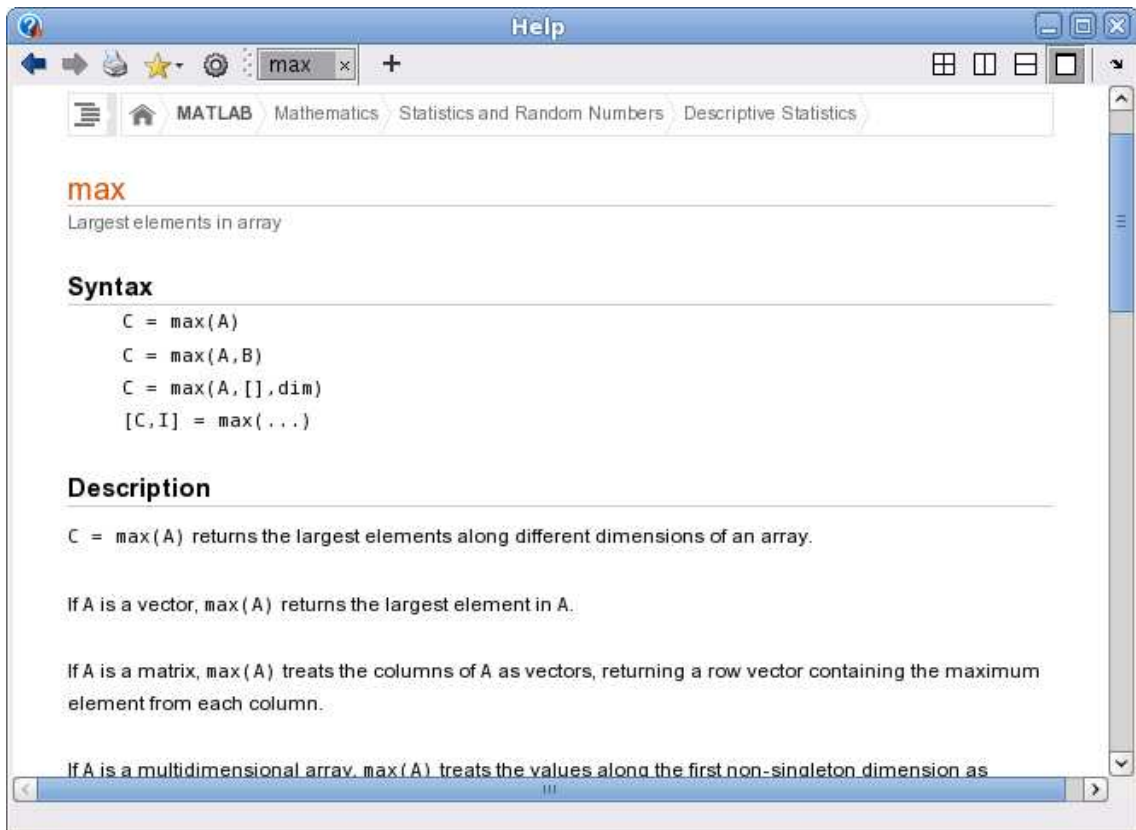
```
>> [m,n]=size(A)
m =
    3
n =
    4
```

Största och minsta elementet i en vektor fås med funktionerna `max` och `min`. För en matris blir det de största elementen i varje kolonn.

```
>> v=max(c)
v =
    8
```

```
>> v=max(A)
v =
    3    6    9   12
```

Vi ser på hjälptexten för `max`



Vi ser att vi med `[v,index]=max(c)` även kan få reda på var det maximala värdet finns någonstans.

Summan och produkten av elementen i vektorn fås med `sum` och `prod`. För en matris blir det summan eller produkten av varje kolonn.

```
>> s=sum(c)
s =
    20
```

```
>> s=sum(A)
s =
     6    15    24    33
```

Kumulativ summa (som blir en vektor) fås med `cumsum` och differens (som också blir en vektor) bildas med `diff`.

```
>> cs=cumsum(c)
cs =
     0     2     6    12    20
```

```
>> ds=diff(c)
ds =
     2     2     2     2
```

För en matris sker kumulationen ned längs kolonnerna och differenserna tas kolonnvis. Resultatet blir matriser i båda fallen.

Användbart om vi skall approximera integral eller derivata av mätresultat.

Vill vi sortera en vektor i stigande ordning gör vi det med `sort`. För en matris blir det varje kolonn som sorteras om i stigande ordning.

3 Matrisoperationer

Vektorn $\mathbf{c} = (0, 2, 4, 6, 8)$ från förra avsnittet kan bildas på två sätt

```
>> c=[0 2 4 6 8]
>> c=0:2:8
```

det senare lite enklare sättet kallas kolon-notation.

Vi låter s få tredje värdet c_3 med $\mathbf{s}=\mathbf{c}(3)$ och bildar vektorn \mathbf{v} av andra och femte värdet, dvs. $\mathbf{v} = (c_2, c_5)$, med

```
>> v=c([2,5])
v =
     2     8
```

Vi kan ändra ett element i \mathbf{v} , t.ex. låta $v_2 = 0$, med

```
>> v(2)=0
v =
     2     0
```

Vi låter s få värdet av elementet på rad 2, kolonn 3 i matrisen \mathbf{A} från inledningen med $s=A(2,3)$ och vi bildar en radvektor \mathbf{v} av rad 3, alla kolonner med

```
>> v=A(3,:)
v =
     3     6     9    12
```

samt en kolonnvektor \mathbf{u} av rad 2-3, kolonn 2 med

```
>> u=A(2:3,2)
u =
     5
     6
```

Vi bildar en matris \mathbf{V} av blocket rad 1-2, kolonn 2-3

```
>> V=A(1:2,2:3)
V =
     4     7
     5     8
```

Om vi har två vektorer $\mathbf{u} = (2, 3, 5)$ och $\mathbf{v} = (1, 2, 3)$ av samma typ och vill bilda summan $\mathbf{a} = \mathbf{u} + \mathbf{v}$ och skillnaden $\mathbf{b} = \mathbf{u} - \mathbf{v}$, så gör vi det med $\mathbf{a}=\mathbf{u}+\mathbf{v}$ respektive $\mathbf{b}=\mathbf{u}-\mathbf{v}$. Operationerna sker elementvis

$$\begin{aligned}\mathbf{a} &= \mathbf{u} + \mathbf{v} = (2, 3, 5) + (1, 2, 3) = (2 + 1, 3 + 2, 5 + 3) = (3, 5, 8) \\ \mathbf{b} &= \mathbf{u} - \mathbf{v} = (2, 3, 5) - (1, 2, 3) = (2 - 1, 3 - 2, 5 - 3) = (1, 1, 2)\end{aligned}$$

eller med andra ord $a_i = u_i + v_i$ och $b_i = u_i - v_i$.

T.ex. vid grafritning behövs de elementvisa motsvarigheterna till multiplikation och division

$$\begin{aligned}\mathbf{u} \text{ .* } \mathbf{v} &= (2, 3, 5) \text{ .* } (1, 2, 3) = (2 \cdot 1, 3 \cdot 2, 5 \cdot 3) = (2, 6, 15) \\ \mathbf{u} \text{ ./ } \mathbf{v} &= (2, 3, 5) \text{ ./ } (1, 2, 3) = (2/1, 3/2, 5/3) = (2, 1.5, 1.666\dots)\end{aligned}$$

Här har vi lånat beteckningar från MATLAB där vi skriver $\mathbf{u} \text{ .* } \mathbf{v}$ respektive $\mathbf{u} \text{ ./ } \mathbf{v}$ för att utföra beräkningarna.

Vi behöver även elementvis upphöjt till, t.ex. kvadrering

$$\mathbf{u} \text{ .}^{\wedge} 2 = (2, 3, 5) \text{ .}^{\wedge} 2 = (2^2, 3^2, 5^2) = (4, 9, 25)$$

Även här har vi lånat beteckningen från MATLAB.

Det finns en funktion `linspace` som är bra för att bygga upp vektorer och funktionerna `zeros` och `ones` för att bygga upp matriser fylla med nollor respektive ettor samt funktionen `eye` för att göra enhetsmatriser.

Uppgift 2. Använd `linspace`, läs hjälptexten, för att bilda vektorn $\mathbf{d} = (2, 5, 8, 11, 14)$.

4 Grafik

4.1 Grafitning

Vi har redan ritat några grafer av funktioner $f(x)$ över ett intervall $a \leq x \leq b$. Nu ser vi på ett exempel där vi behöver den elementvisa multiplikationen från förra avsnittet.

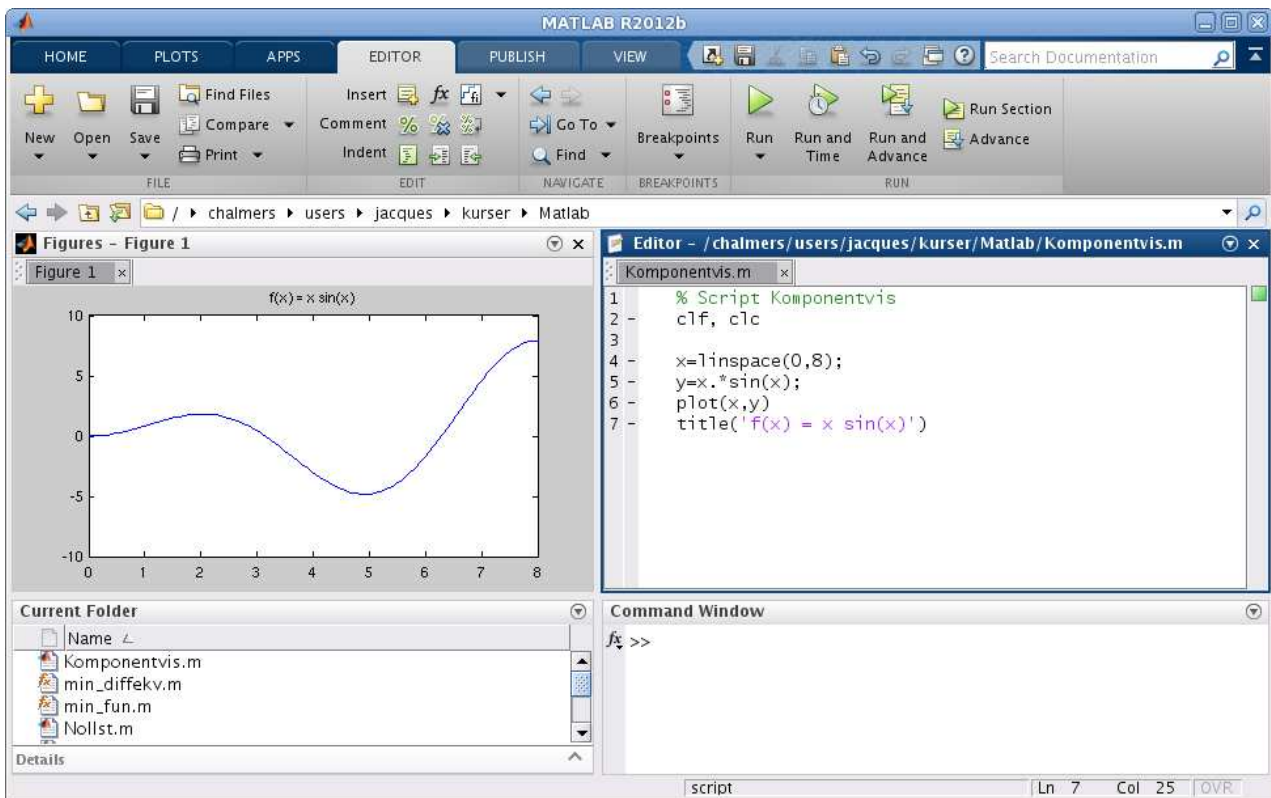
Exempel 1. Rita grafen till $f(x) = x \sin(x)$ över intervallet $0 \leq x \leq 8$.

Vi bildar en vektor $\mathbf{x} = (x_1, x_2, \dots, x_n)$ med värden jämnt fördelade över intervallet $0 \leq x \leq 8$. Sedan bildar vi vektorn

$$\mathbf{y} = (f(x_1), f(x_2), \dots, f(x_n)) = (x_1 \sin(x_1), x_2 \sin(x_2), \dots, x_n \sin(x_n)) = \mathbf{x} .* \sin(\mathbf{x})$$

och ritat upp grafen.

```
>> x=linspace(0,8);  
>> y=x.*sin(x);  
>> plot(x,y)  
>> title('f(x) = x sin(x)')
```



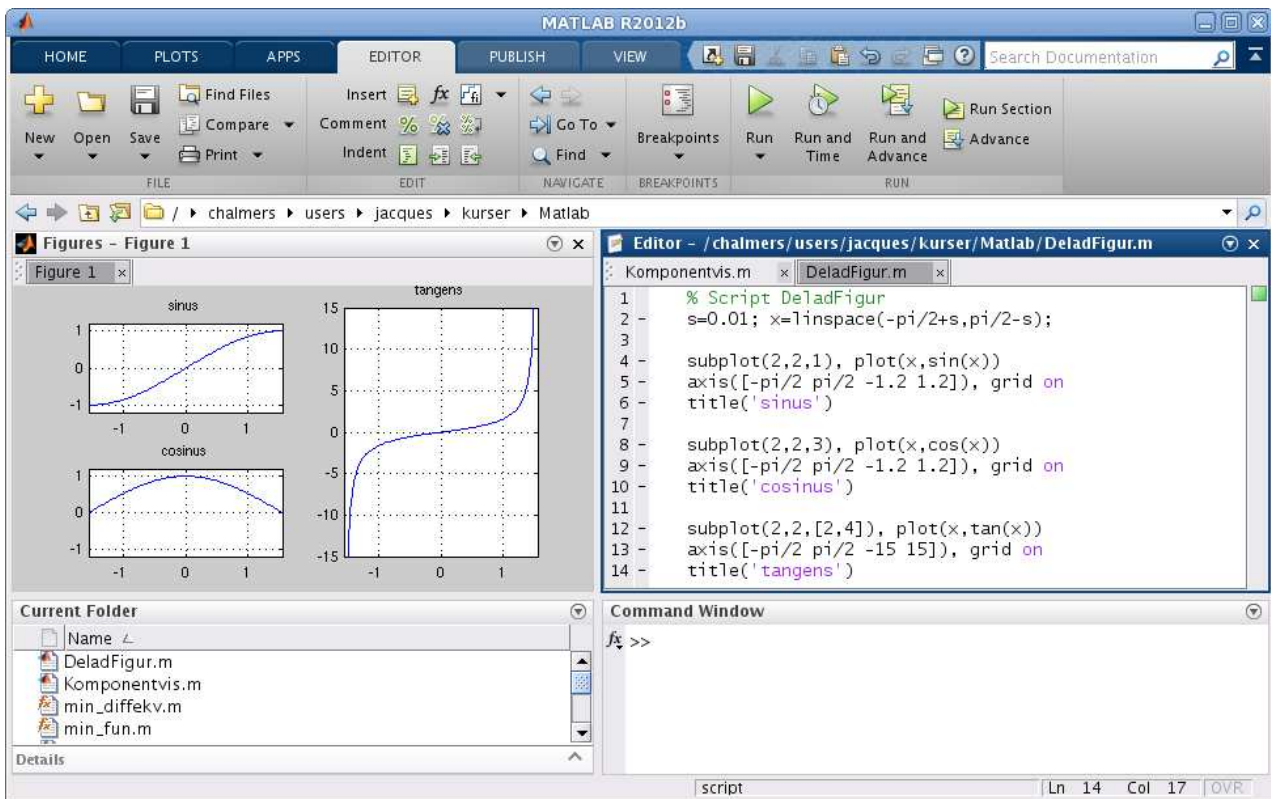
Uppgift 3. Rita grafen till $f(x) = x - x \cos(7x)$ över intervallet $0 \leq x \leq 8$.

4.2 Dela figuren

Ibland vill man ha flera koordinatsystem i samma figur-fönster (Figure). Då använder man kommandot `subplot`. Vi ser på ett exempel.

Exempel 2. Vi skall i samma figur göra tre olika koordinatsystem. I dessa skall vi rita grafen av $\sin(x)$, $\cos(x)$ respektive $\tan(x)$ över intervallet $-\frac{\pi}{2} \leq x \leq \frac{\pi}{2}$.

Så här kommer det se ut



```
>> s=0.01; x=linspace(-pi/2+s,pi/2-s);
>> subplot(2,2,1) % delar in Figure i 2x2 delar och gör 1:a aktiv
>> plot(x,sin(x))
>> axis([-pi/2 pi/2 -1.2 1.2]), grid on, title('sinus')
>> subplot(2,2,3) % delar in Figure i 2x2 delar och gör 3:e aktiv
>> plot(x,cos(x))
>> axis([-pi/2 pi/2 -1.2 1.2]), grid on, title('cosinus')
>> subplot(2,2,[2,4]) % samma indelning men gör 2:a och 4:e aktiva
>> plot(x,tan(x))
>> axis([-pi/2 pi/2 -15 15]), grid on, title('tangens')
```

4.3 Kurvritning

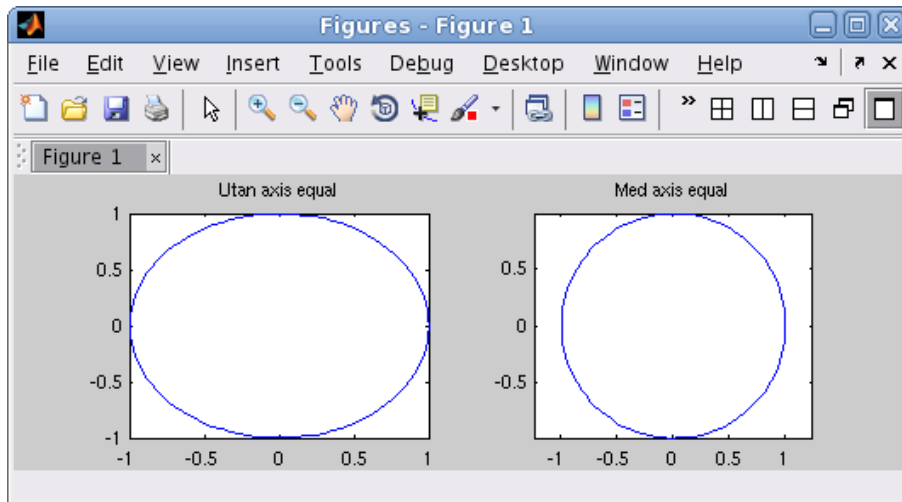
Nu skall vi rita s.k. parameterframställda kurvor. Som exempel tar vi enhetscirkeln

$$t \mapsto (x(t), y(t)) = (\cos(t), \sin(t)), \quad 0 \leq t \leq 2\pi$$

När man ritar sådana kurvor ritar man inte ut parametern t utan enbart x - och y -värdena.

```
>> t=linspace(0,2*pi);
>> x=cos(t); y=sin(t);
>> subplot(1,2,1)
>> plot(x,y)
>> title('Utan axis equal')
```

```
>> subplot(1,2,2)
>> plot(x,y)
>> axis equal % annars blir cirkeln tillplattad
>> title('Med axis equal')
```



Uppgift 4. Rita kurvorna $t \mapsto (x(t), y(t)) = (\cos(t) + \cos(3t), \sin(2t))$ och $t \mapsto (x(t), y(t)) = (\cos(t) + \cos(4t), \sin(2t))$, för $0 \leq t \leq 2\pi$. Använd `subplot`.

4.4 Polygontåg

Ett polygontåg som ges av $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, ritas upp i MATLAB genom att man bildar vektorerna $\mathbf{x} = (x_1, x_2, \dots, x_n)$ och $\mathbf{y} = (y_1, y_2, \dots, y_n)$ och sedan gör `plot(x,y)`.

Grafitrning är ett polygontåg vi ritat upp. Tag t.ex. grafen till $f(x) = \sin(x)$ för $0 \leq x \leq 2\pi$. Vi har då $\mathbf{x} = (x_1, x_2, \dots, x_n)$ med $0 = x_1 < x_2 < \dots < x_n = 2\pi$ och $\mathbf{y} = (y_1, y_2, \dots, y_n)$ med $y_i = \sin(x_i)$. Sedan ritas vi upp med `plot(x,y)`.

Om polygontåget är slutet, dvs. $x_n = x_1$ och $y_n = y_1$, och om det inte korsar sig självt så omsluter det ett område i planet, ett s.k. polygonområde. Vi kan använda `fill` för att färglägga ett sådant område.

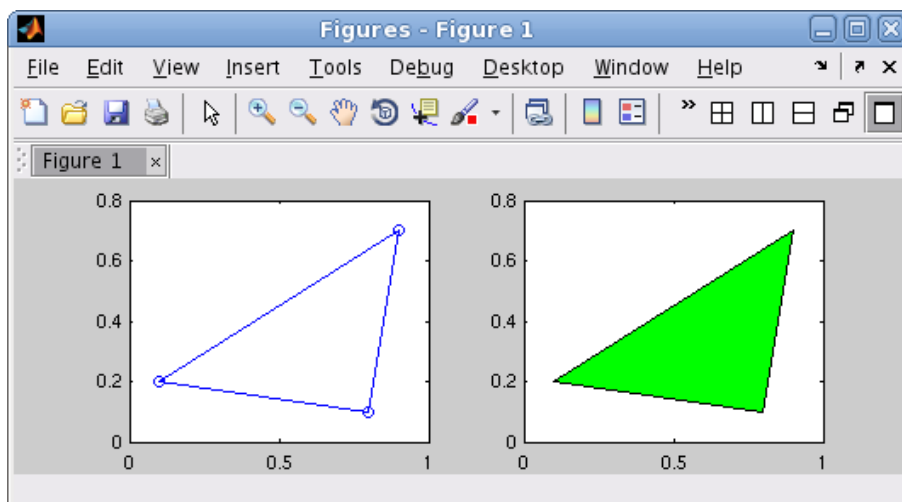
Vi ritas upp polygontåg som ges av $(0.1, 0.2), (0.8, 0.1), (0.9, 0.7), (0.1, 0.2)$, dvs. en triangel.

```
>> x=[0.1 0.8 0.9 0.1];
>> y=[0.2 0.1 0.7 0.2];
>> subplot(1,2,1)
>> plot(x,y, '-o')
>> axis([0 1 0 0.8])
```

Med `'-o'` anger vi att punkterna både skall förbindas med räta linjer och markeras med små ringar.

Vi fyller området med grön färg och vi använder `axis` för att få lite "luft" runt triangeln.

```
>> subplot(1,2,2)
>> fill(x,y, 'g')
>> axis([0 1 0 0.8])
```

Uppgift 5. Rita cirkel som fylls med grön färg och rita en inskriven kvadrat i en cirkel, som fylls med gul färg (använd `hold on`).

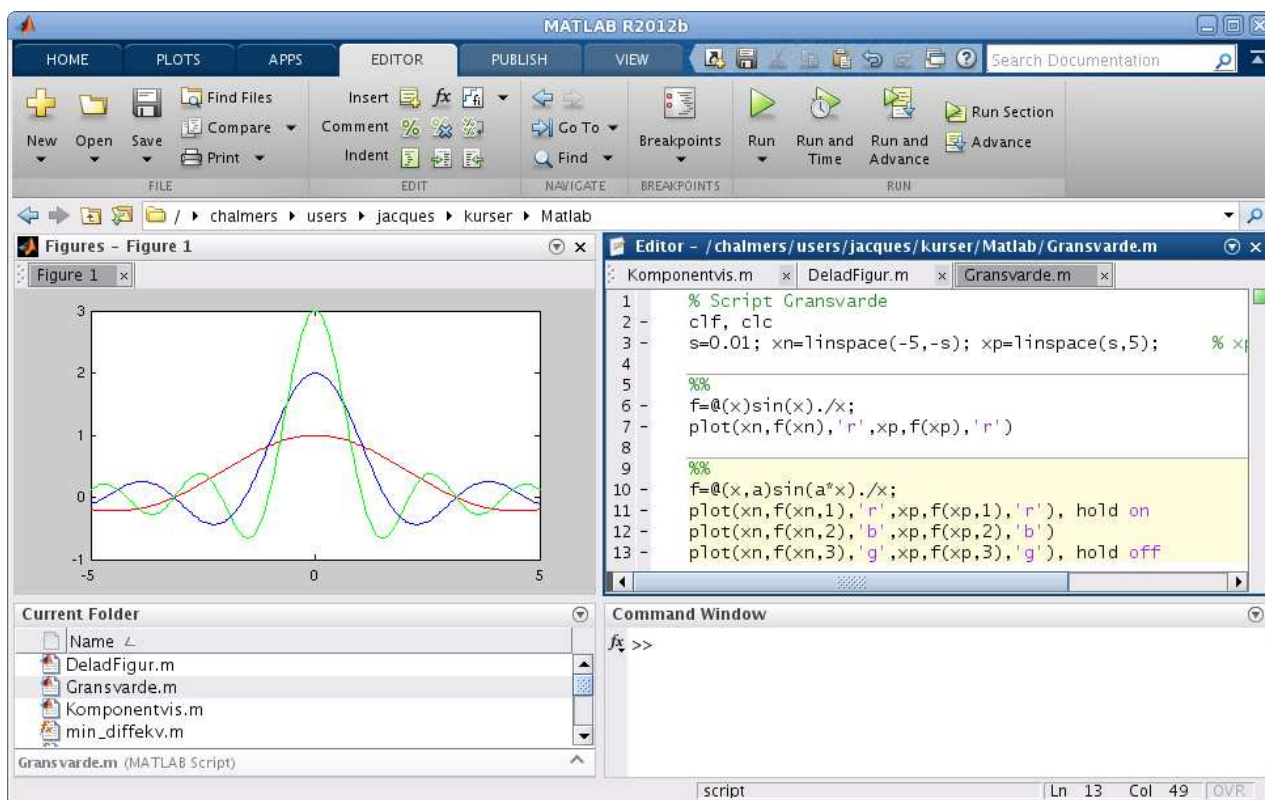
5 Egna funktioner

Exempel 3. Vi skall rita grafen av

$$f(x) = \frac{\sin(ax)}{x}$$

över intervallet $-5 \leq x \leq 5$, för $a = 1, 2$, och 3 .

Vi måste tänka på att $f(x)$ inte definierad i $x = 0$. Finns gränsvärdet och vad blir det i så fall?



```
>> s=0.01;           % s för att separera nollan
>> xn=linspace(-5,-s); % xn negativa x-värden
>> xp=linspace(s,5);  % xp positiva x-värden
```

Nu behöver vi elementvis division och vi inför ett funktionshandtag (function handle).

Vi ser först på fallet $a = 1$

```
>> f=@(x)sin(x)./x;
>> plot(xn,f(xn),'r',xp,f(xp),'r')
```

Nu skall vi ta fallen $a = 2$ och 3 också. Enklast är om vi gör om vårt funktionshandtag så att även a blir ett argument.

```
>> f=@(x,a)sin(a*x)./x;
>> plot(xn,f(xn,1),'r',xp,f(xp,1),'r') % a=1
>> hold on
>> plot(xn,f(xn,2),'b',xp,f(xp,2),'b') % a=2
>> plot(xn,f(xn,3),'g',xp,f(xp,3),'g') % a=3
>> hold off
```

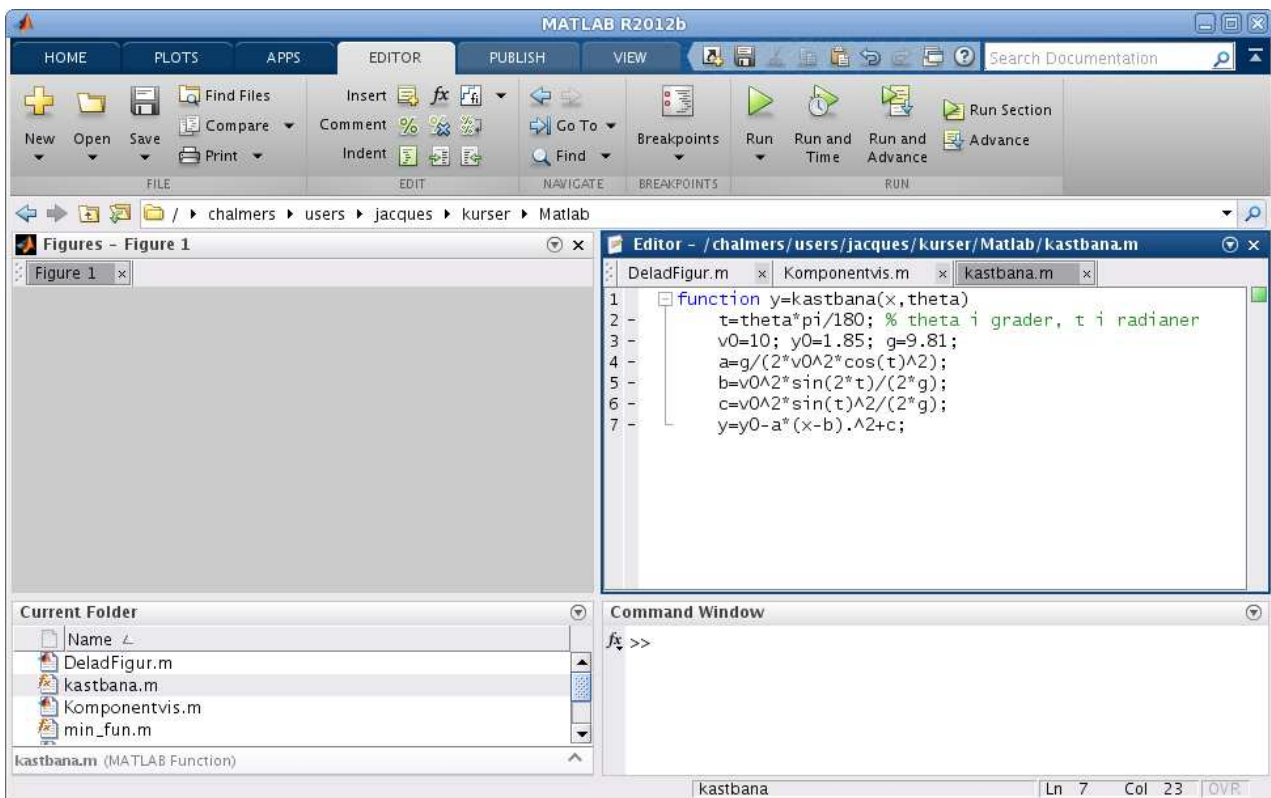
Exempel 4. Kastbana utan luftmotstånd beskrivs av

$$y(x) = y_0 - \frac{g}{2v_0^2 \cos^2(\theta)} \left(x - \frac{v_0^2 \sin(2\theta)}{2g} \right)^2 + \frac{v_0^2 \sin^2(\theta)}{2g}$$

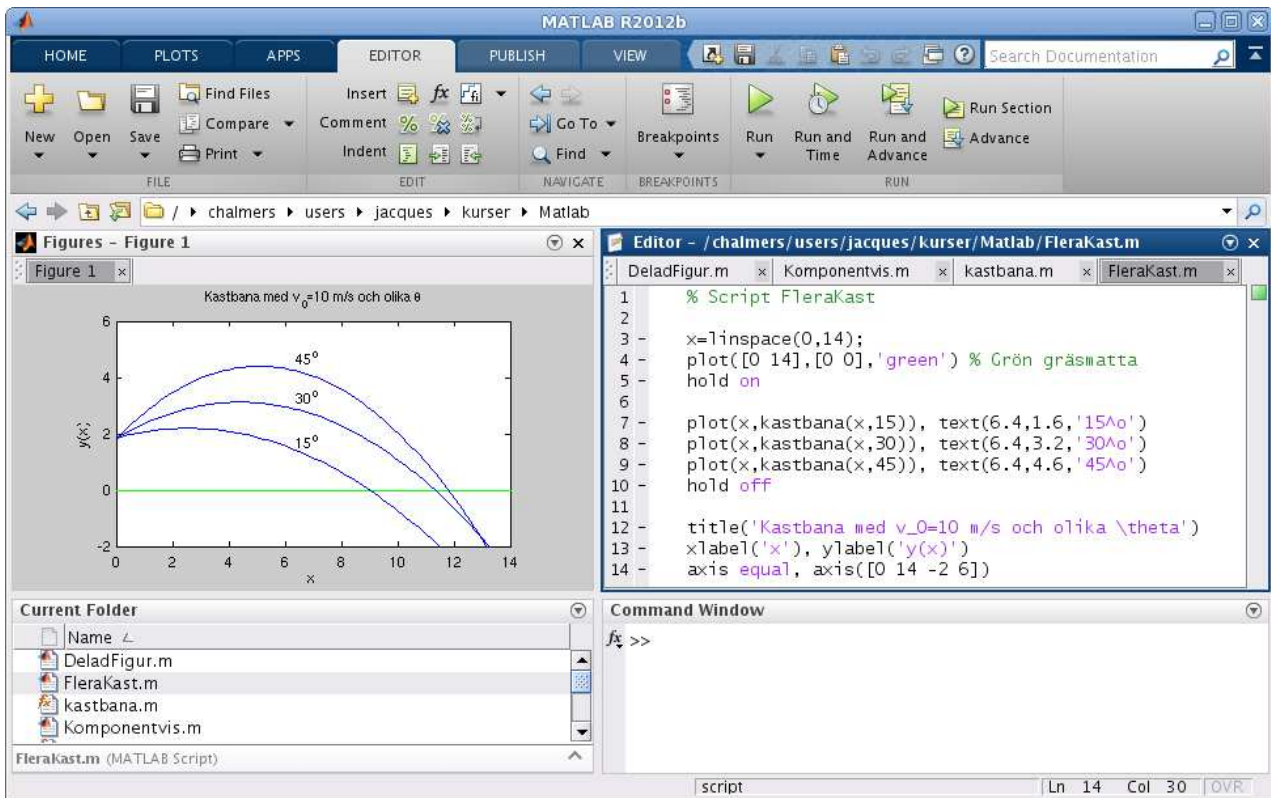
där v_0 är utkastfarten, y_0 är utkasthöjden och θ är utkastvinkeln.

Tag $v_0 = 10$ m/s, $y_0 = 1.85$ m och rita kastbanorna för några olika utkastvinklar.

Vi gör en function med namnet `kastbana` som beskriver kastbanan för olika utkastvinklar.



Vi gör ett script för att rita graferna. Med kommandot `text` placerar vi ut lite förklarande text vid graferna.



Uppgift 6. Skriv den function och det script för kastbanan som vi pratar om i exemplet. Rita graferna. Varför delar vi upp funktionsuttrycket för $y(x)$ i flera delar?