

Mer om funktioner och grafik i MATLAB

Jönsson: 4.1-4.4, 4.6-4.12, 5.1, 5.3-5.5

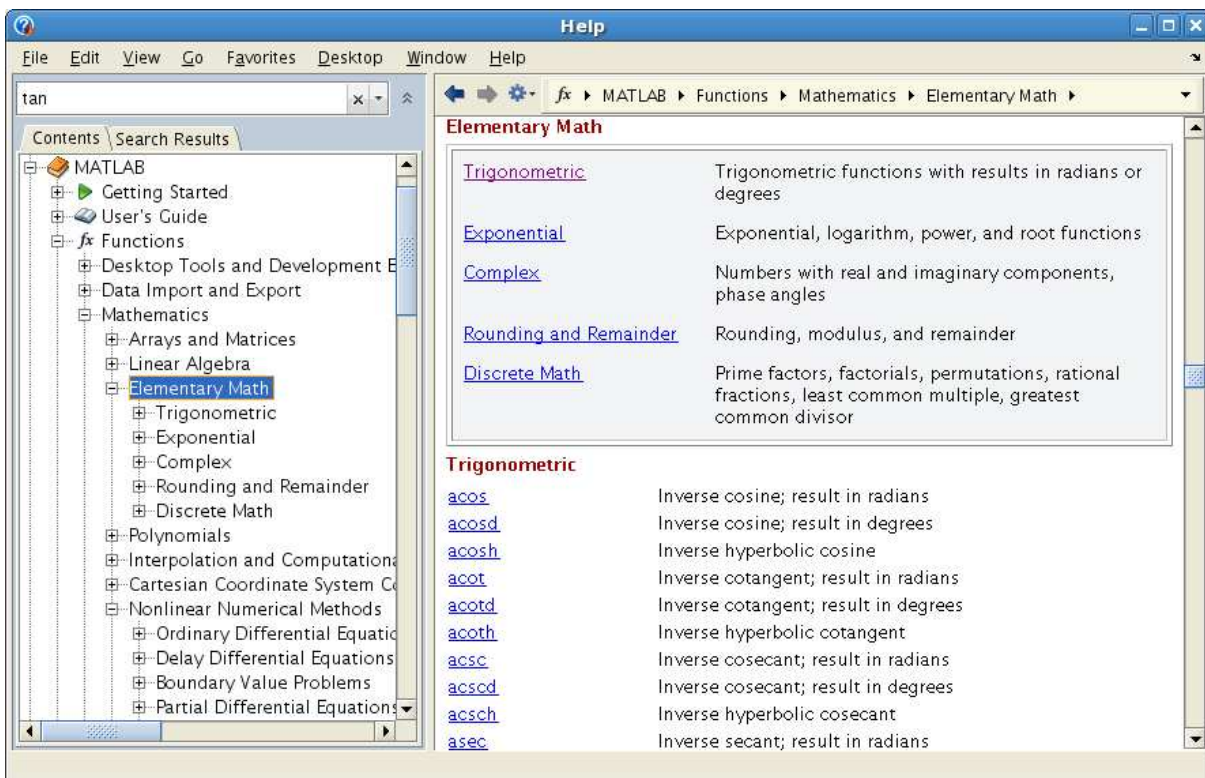
1 Inledning

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 avslutningsvis se hur man definerar egna funktioner.

2 Inbyggda funktioner

2.1 Elementära funktioner

Vi letar upp hjälptexterna elementära eller matematiska funktioner i Help Navigator genom att succesivt öppna MATLAB, User Guide, Functions, Mathematics och slutligen Elementary Math.



Vi har en gruppering i trigonometriska, exponential, komplexa, avrundning och rester samt diskreta funktioner.

Funktioner som sinus och cosinus, kan operera på matriser. Man får som resultat en matris av samma typ 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

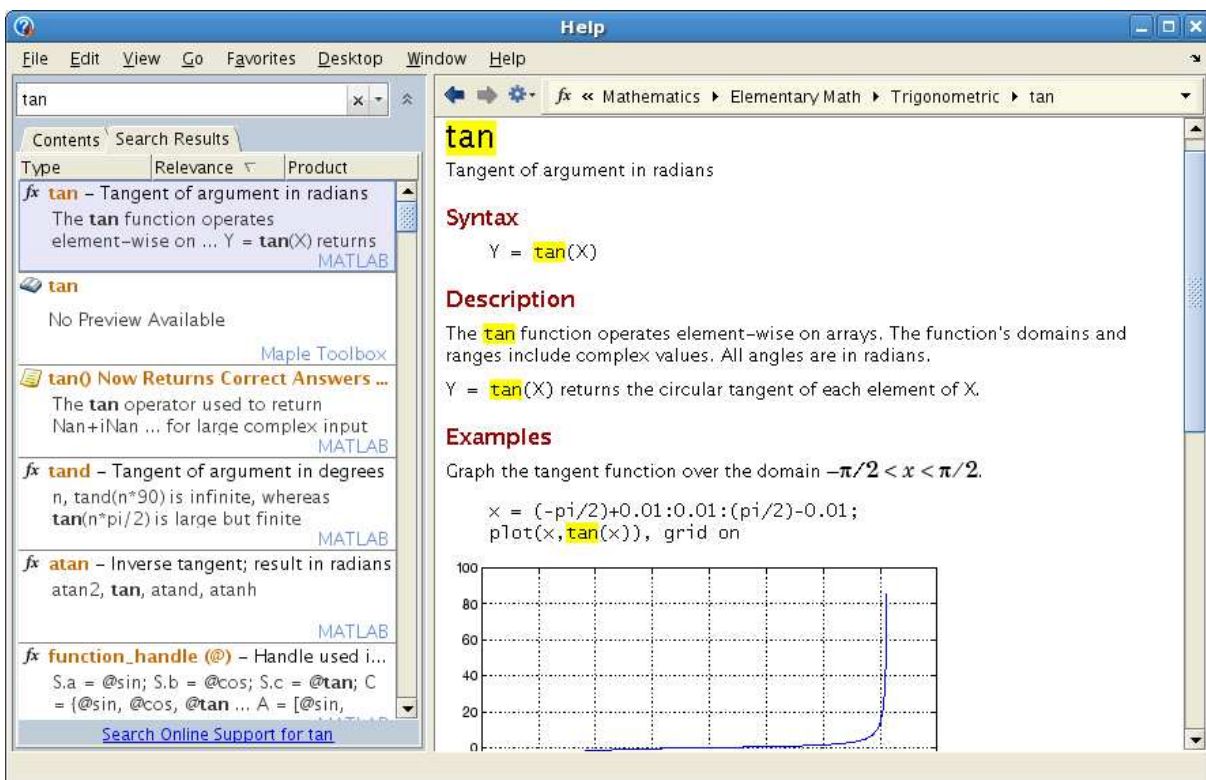
```
>> A=[1 4 7 10; 2 5 8 11; 3 6 9 12]
>> c=[0 2 4 6 8]
```

Nu beräknar vi sinus av vektorn **c** och matrisen **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 `c` och matrisen `A` från förra avsnittet. Antal element i vektorn `c` ges av

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

och antalet rader och kolonner `A` fås med

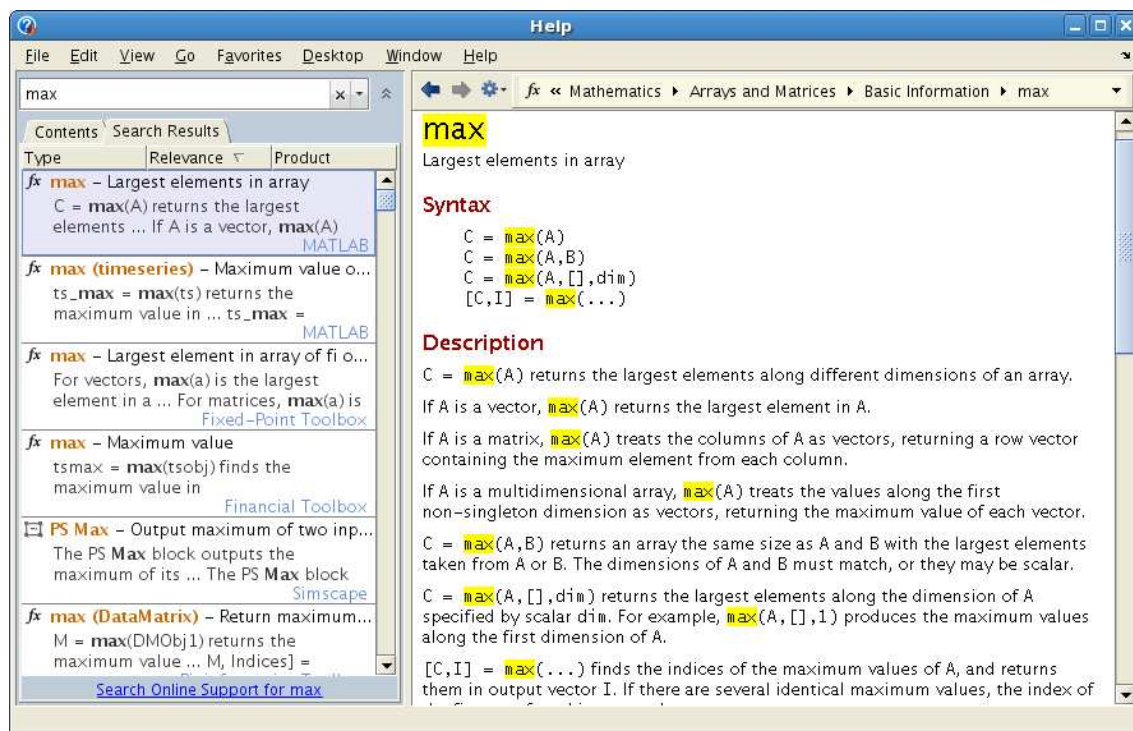
```
>> [m,n]=size(A)
m =
    3
n =
    4
```

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 diskreta funktioner.

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 `s=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 $\mathbf{s}=\mathbf{A}(2,3)$ 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 komponentvis

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

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

T.ex. vid grafitning behövs de komponentvisa 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 komponentvis 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. Se i Jönsson avsnitt 4.2 och 4.6.

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 komponentvisa operationer.

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

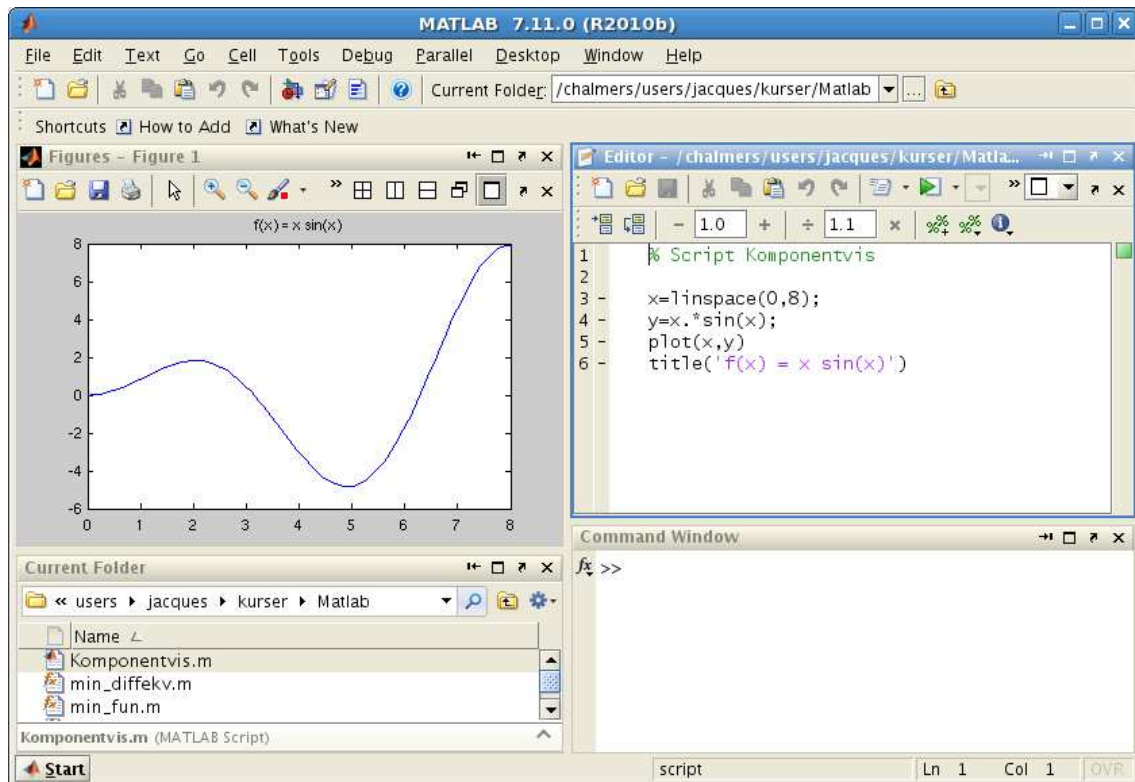
$$\mathbf{y} = (f(x_1), f(x_2), \dots, f(x_n)) = (x_1 \sin(x_1), x_2 \sin(x_2), \dots, x_n \sin(x_n))$$

och ritar upp grafen. För att bilda vektorn $\mathbf{y}$ behövs den komponentvisa multiplikationen.

Vi ritar grafen med

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

och så här ser resultatet ut



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

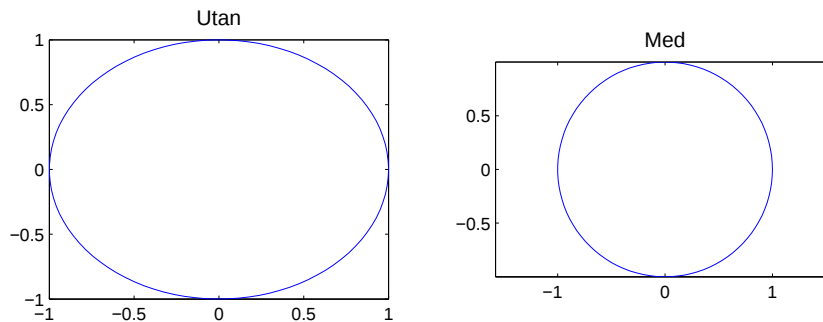
4.2 Kurvritning

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);  
>> plot(cos(t),sin(t))  
>> axis equal           % annars blir cirkeln tillplattad
```



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 $0 \leq t \leq 2\pi$. Rita en kurva i taget!

4.3 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)`.

Grafritning är ett polygontåg vi ritar 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 ritar vi upp med `plot(x,y)`.

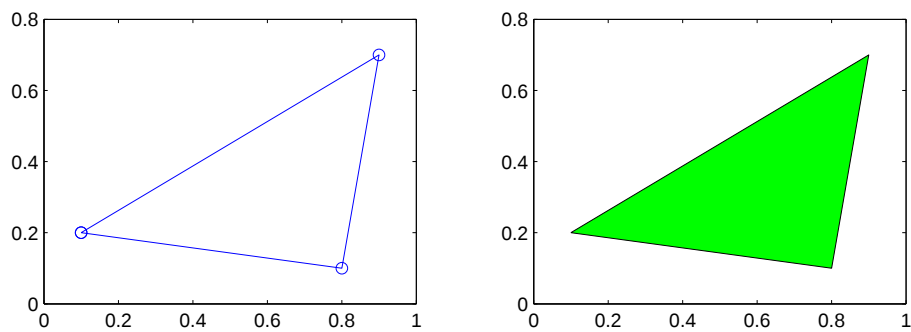
Om polygontåget är slutet, dvs. $x_n = x_1$ och $y_n = y_1$, och om det inte korsar sig självt så omsluter det ett område i planet, ett s.k. polygonområde. Vi kan använda `fill` för att färglägga ett sådant område.

Vi ritar 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];  
>> plot(x,y,'o-')  
>> axis([0 1 0 0.8])
```

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

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



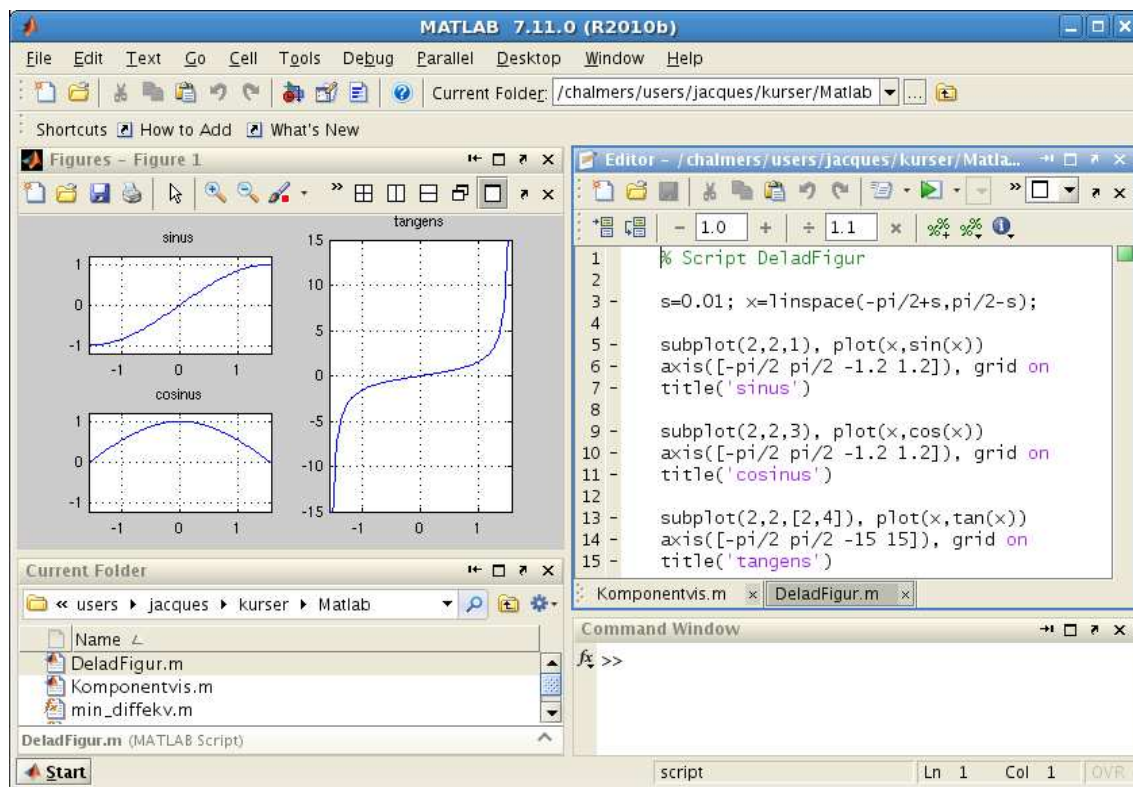
Uppgift 5. Rita cirkel som fylls med grön färg och rita en inskriven kvadrat i en cirkel, som fylls med gul färg (använd `hold on`).

4.4 Dela figuren

Ibland vill man ha flera koordinatsystem i samma figur-fönster (Figure).

Exempel 2. 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')
```

Uppgift 6. Rita kurvan

$$t \mapsto (x(t), y(t)) = (4 \cos(-\frac{at}{4}) + 7 \cos(t), 4 \sin(-\frac{at}{4}) + 7 \sin(t))$$

med $0 \leq t \leq 80\pi$, för t.ex. $a = 1$, $a = 1.01$, $a = 2$ och $a = 7$. Använd subplot.

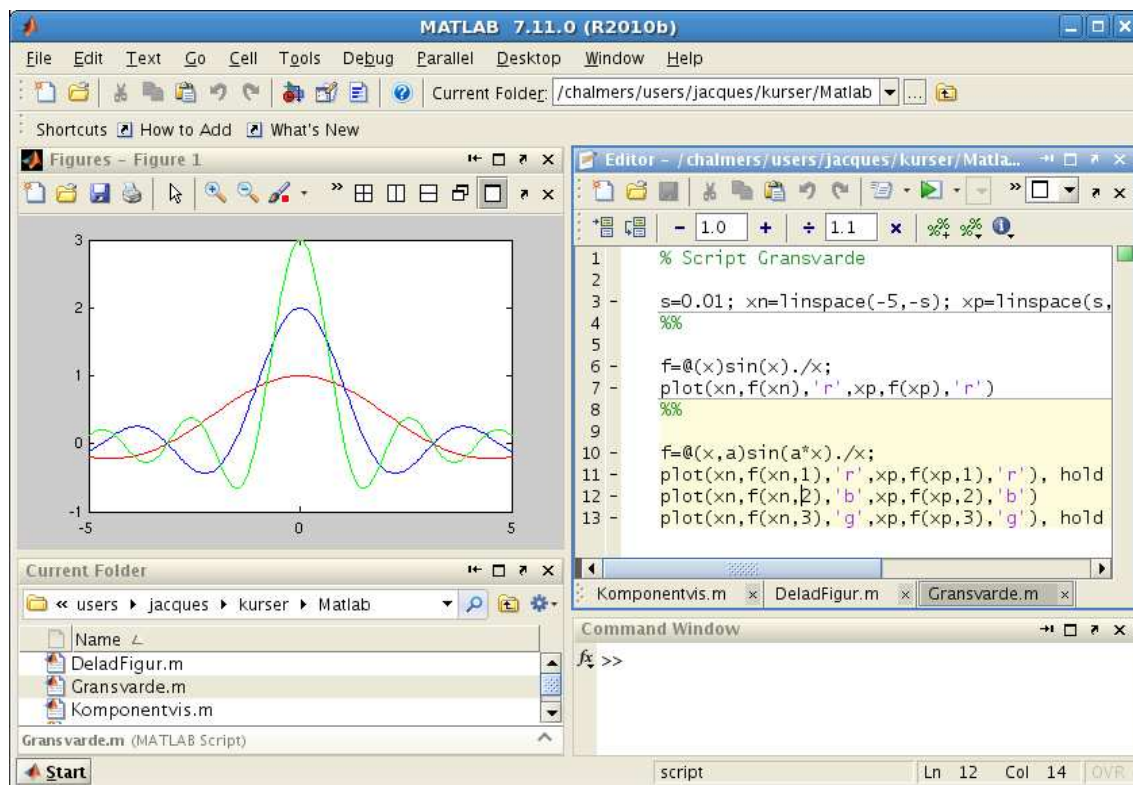
5 Egna funktioner

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 komponentvis 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
```

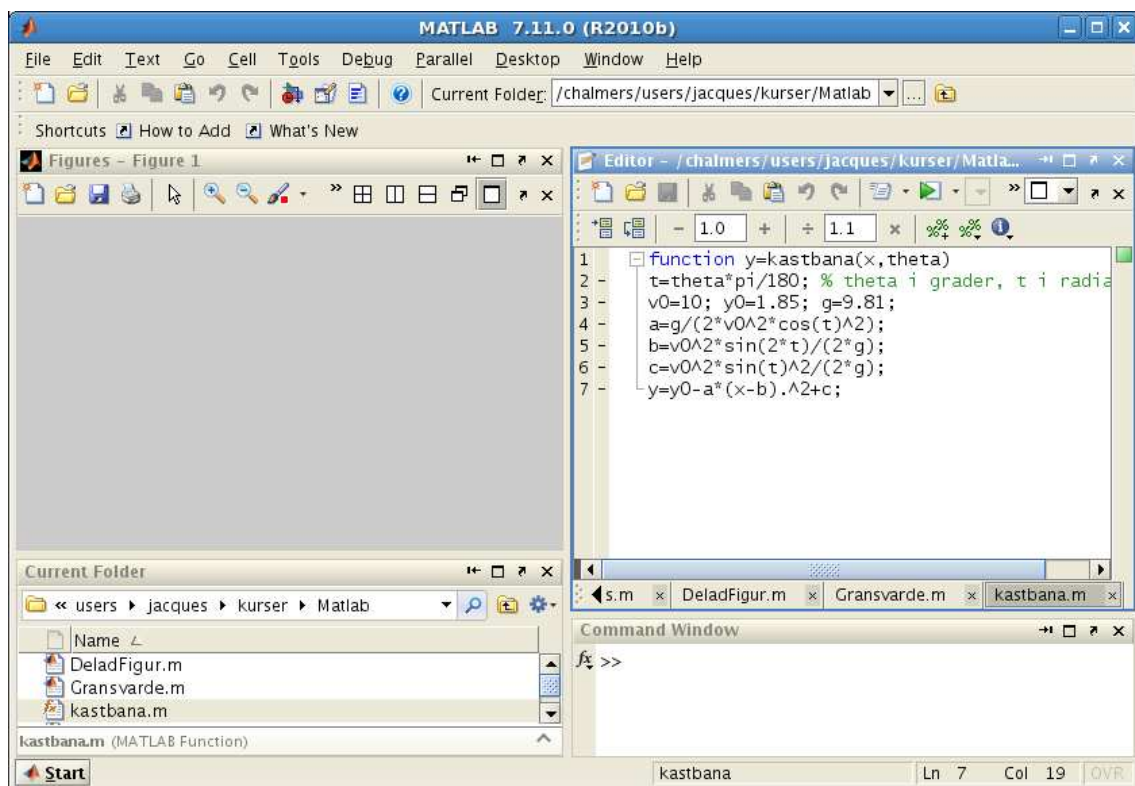
Exempel 4. Kastbana utan luftmotstånd beskrivs av

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

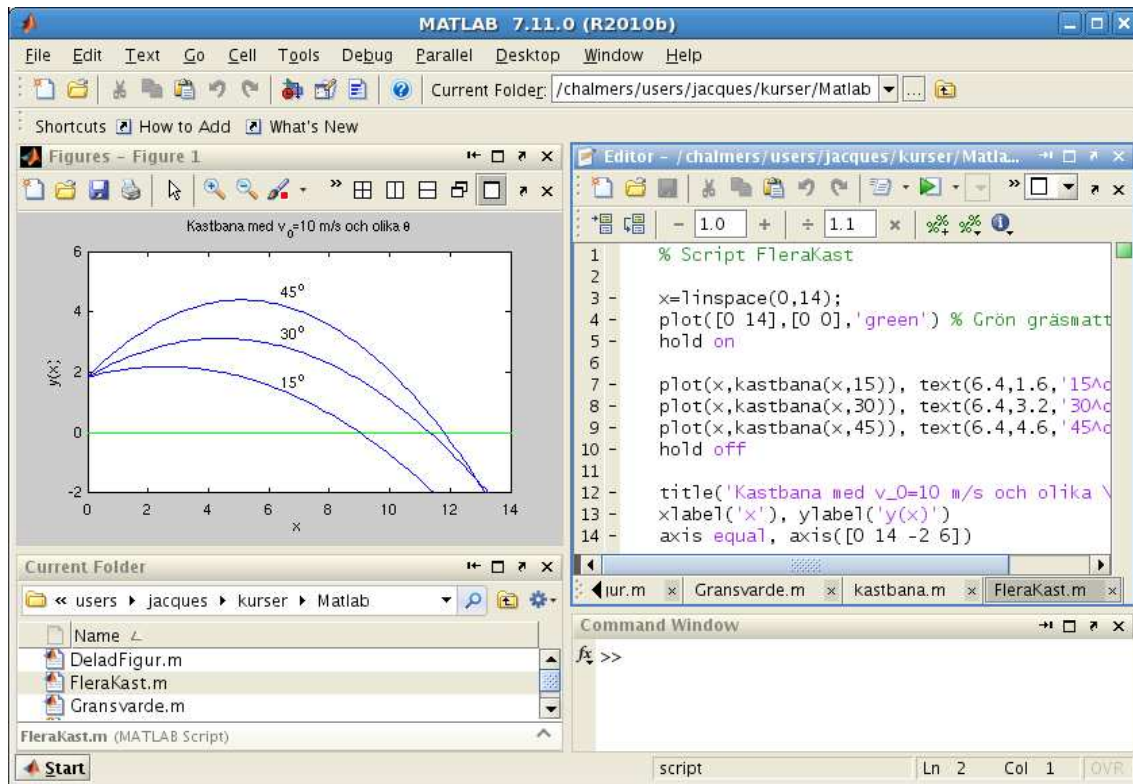
där v_0 är utkastfarten, y_0 är utkasthöjden och θ är utkastvinkeln.

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 en script för att rita graferna.



Vi har också placerat ut lite förklarande text vid graferna med kommandot `text`.

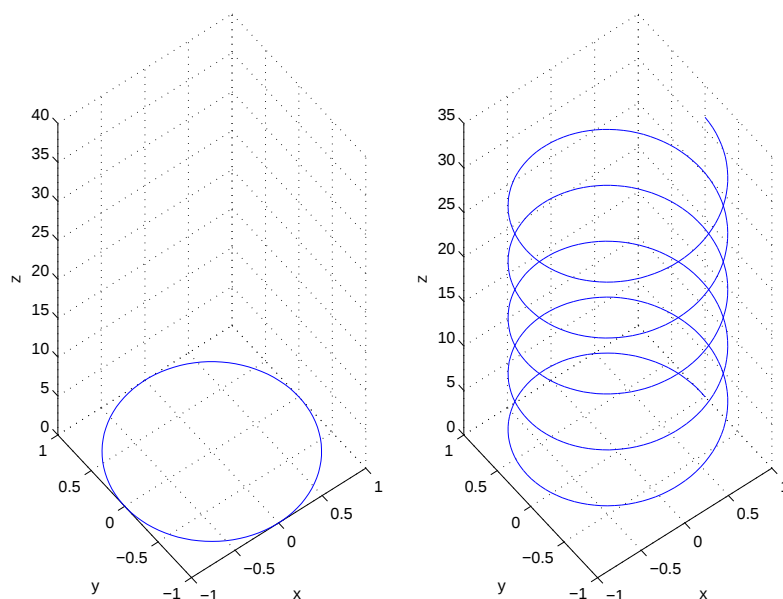
6 Lite om kurvor och ytor i $\mathbb{R}^3$

Vi har ritat kurvor i $\mathbb{R}^2$ med kommandot `plot`, t.ex. en cirkel $t \mapsto (x(t), y(t)) = (\cos(t), \sin(t))$, $0 \leq t \leq 2\pi$. Om vi vill rita en kurva i $\mathbb{R}^3$ kan vi använda kommandot `plot3`.

Som exempel ser vi på spiralen

$$t \mapsto (x(t), y(t), z(t)) = (\cos(t), \sin(t), t)$$

för $0 \leq t \leq 10\pi$.



```
>> t=linspace(0,10*pi,500);
>> plot3(cos(t),sin(t),t)
>> grid on
>> xlabel('x'), ylabel('y')
```

Till vänster ritade vi cirkeln för jämförelse. Vi genomlöper den fem gånger samtidigt som vi rör oss uppåt för att få spiralen.

I kursen matematisk analys i flera variabler, studerar man funktioner som inte beror bara av en variabel, utan av flera. Man vill då bl.a. rita funktionsytor, motsvarigheten till en graf. Det som följer får ni nog se som överkurs, om man vill kan gärna vänta tills läsperiod 3.

Exempel 5. Vi ser på följande funktionsytan, eller grafen, till funktionen

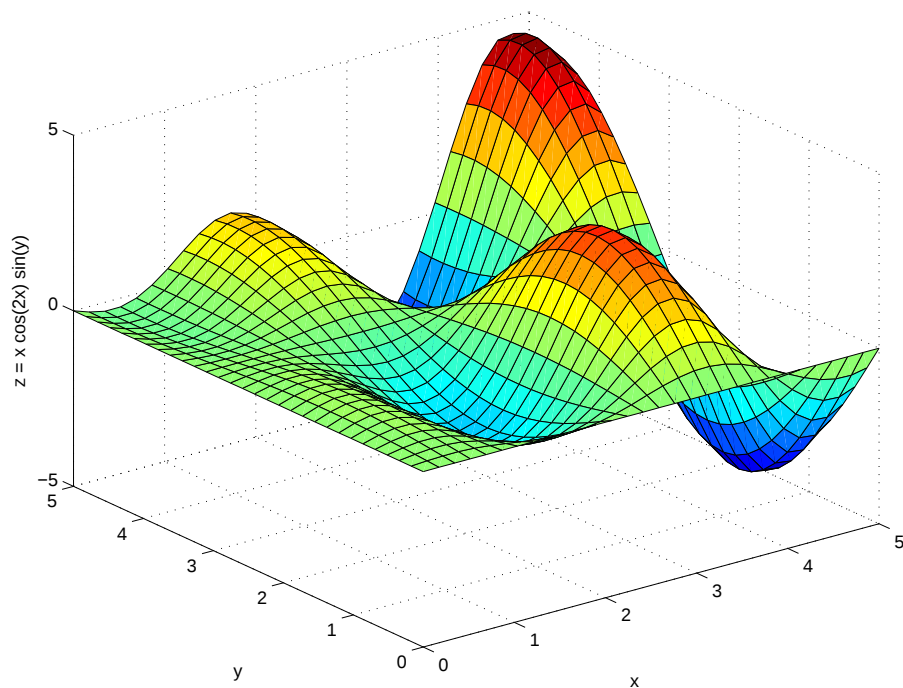
$$f(x, y) = x \cos(2x) \sin(y)$$

över området $0 \leq x \leq 5$ och $0 \leq y \leq 5$.

Den yta vi skall rita upp består av alla punkter $(x, y, f(x, y))$ i $\mathbb{R}^3$ där $0 \leq x \leq 5$ och $0 \leq y \leq 5$.

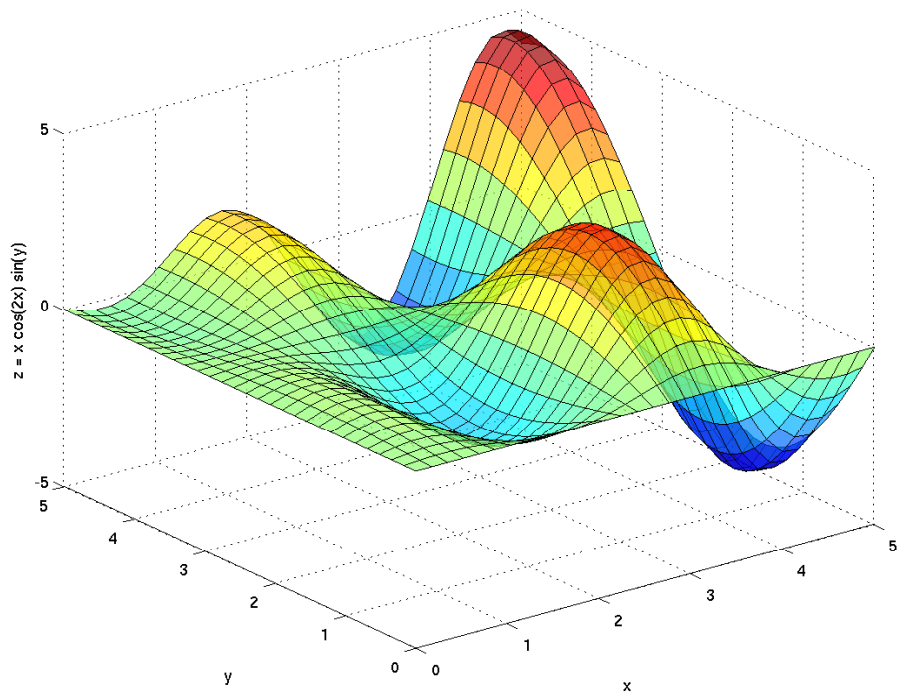
Resultatet får vi med kommandot **surf**, vilket är motsvarigheten till **plot** då vi skall rita ytor.

```
>> x=linspace(0,5,30); y=linspace(0,5,30);
>> [X,Y]=meshgrid(x,y);
>> Z=X.*cos(2*X).*sin(Y);
>> surf(X,Y,Z)
>> grid on
>> xlabel('x'), ylabel('y'), zlabel('z = x cos(2x) sin(y)')
```



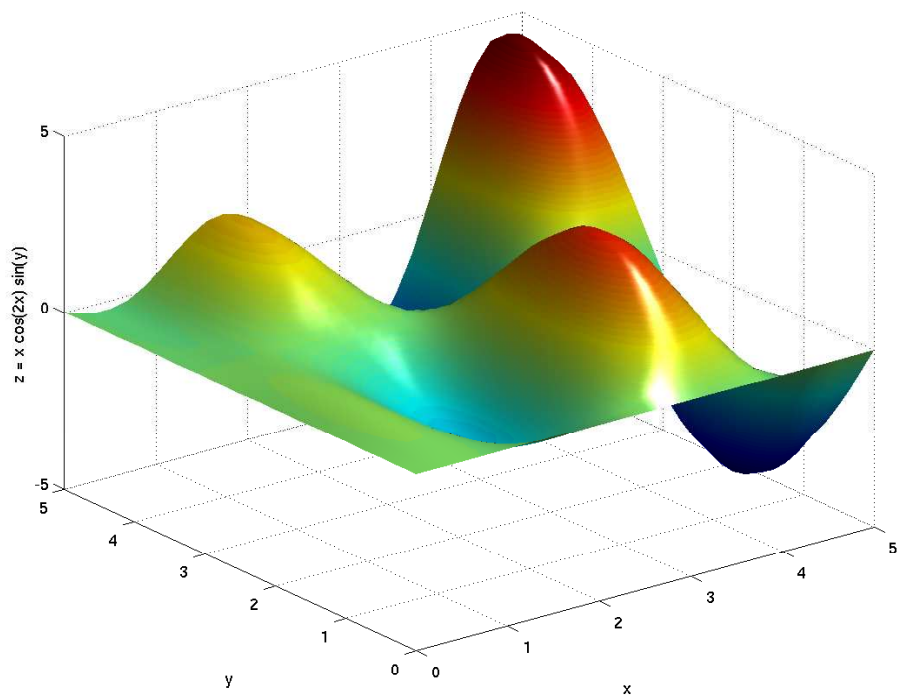
Ibland vill man göra en yta lite genomskinlig

```
>> surf(X,Y,Z,'facealpha',0.7)
```



Vi kan också lägga på belysning, välja mellan olika belysningsmodeller och välja mellan olika reflexionsmodeller (material).

```
>> surf(X,Y,Z)
>> shading interp          % flat, interp, faceted
>> camlight right          % left, right, headlight
>> lighting phong          % none, flat, phong, gouraud
>> material shiny          % shiny, dull, metal
```



Uppgift 7. Rita funktionsytan till

$$f(x, y) = -xye^{-2(x^2+y^2)}$$

över området $-2 \leq x \leq 2, -2 \leq y \leq 2$.

Vi ser på två parametriserade ytor som till utseendet är välbekanta.

Exempel 6. En cirkulär **cylinder** med radien r och höjden h som kan beskrivas av

$$x^2 + y^2 = r^2, \quad 0 \leq z \leq h$$

Vi kan också göra en s.k. parametrisering

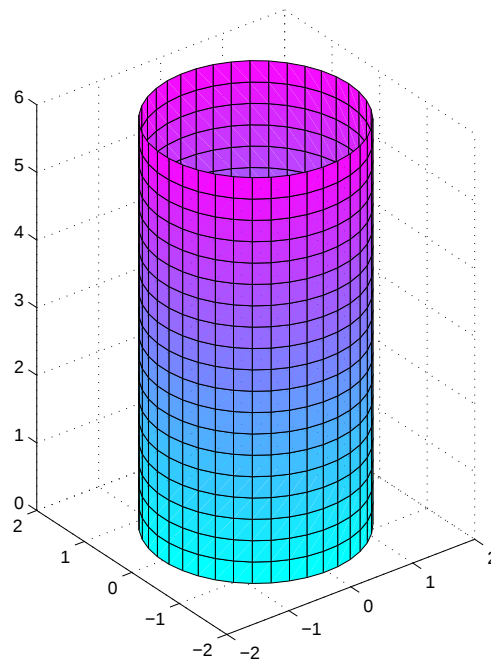
$$\begin{cases} x(s, t) = r \cos(t) \\ y(s, t) = r \sin(t) \\ z(s, t) = s \end{cases}$$

där $0 \leq t \leq 2\pi$ och $0 \leq s \leq h$.

Parametriseringen gör att vi lätt kan rita upp ytan i MATLAB.

Vi ritar en bild av en cylinder med radien $r = 1.5$ och höjden $h = 6$ enligt

```
r=1.5; h=6; n=40; m=20;  
s=linspace(0,h,m); t=linspace(0,2*pi,n);  
[S,T]=meshgrid(s,t);  
R=r*ones(size(S));  
X=r*cos(T); Y=r*sin(T); Z=S;  
surf(X,Y,Z)  
axis equal, axis([-2 2 -2 2 0 6])  
colormap(cool)
```



En **sfär** med radien r och centrum i origo ges av

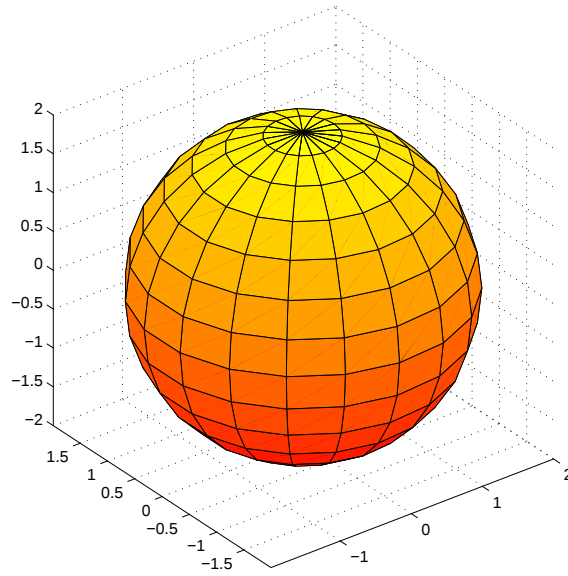
$$x^2 + y^2 + z^2 = r^2$$

och kan parametriseras med

$$\begin{cases} x(s, t) = r \cos(s) \cos(t) \\ y(s, t) = r \cos(s) \sin(t) \\ z(s, t) = r \sin(s) \end{cases}$$

där $-\frac{\pi}{2} \leq s \leq \frac{\pi}{2}$ och $0 \leq t \leq 2\pi$.

Vi ritar upp en sfär med radien $r = 2$.



```
r=2; n=15; m=20;
s=linspace(-pi/2,pi/2,n); t=linspace(0,2*pi,m);
[S,T]=meshgrid(s,t);
% Genererar n x m matriser X, Y, Z, så att surf(X,Y,Z) ger sfären
X=r*cos(S).*cos(T);
Y=r*cos(S).*sin(T);
Z=r*sin(S);
surf(X,Y,Z)
axis equal
colormap(autumn)
```

En **torus** med lateralradien r och centralradien a samt centrum i origo ges av

$$(x^2 + y^2 + z^2 + a^2 - r^2)^2 = 4a^2(x^2 + y^2)$$

och kan parametriseras med

$$\begin{cases} x(s, t) = (a + r \cos(s)) \cos(t) \\ y(s, t) = (a + r \cos(s)) \sin(t) \\ z(s, t) = r \sin(s) \end{cases}$$

där $-\pi \leq s \leq \pi$ och $0 \leq t \leq 2\pi$.

Vi ritar en torus lateralradien $r = 0.8$ och centralradien $a = 2$.

```

r=0.8; a=2; n=15; m=40;
s=linspace(-pi,pi,n); t=linspace(0,2*pi,m);
[S,T]=meshgrid(s,t);
% Genererar n x m matriser X, Y, Z, så att surf(X,Y,Z) ger torusen
X=(a+r*cos(S)).*cos(T);
Y=(a+r*cos(S)).*sin(T);
Z=r*sin(S);
surf(X,Y,Z)
axis equal
colormap(winter)

```

