



MATLAB

Introduktion

MATLAB är en integrerad miljö, med matriser som grundkomponent. Här finns avancerade möjligheter att göra beräkningar och att visualisera storheter och resultat. I MATLAB har man möjlighet både till direkt interpreterande beräkningar samt programmering med hjälp av funktioner och kommandofiler (.m-filer)

Syfte

Övningarna här syftar till att göra dig bekant med MATLAB'S kommandosyntax och lite av grafikmöjligheterna.

Tips

- Använd de inbyggda hjälpfunktionerna, kommandot `help` i kommandofönstret, eller via meny-val eller med kommandot `helpbrowser` som ger dig mer detaljerad information i web-liknande format (förteckning över hjälpfunktioner i tabell på s. 23).
- `who` visar aktuella variabler
- `format compact` ger mindre blankrader i utskriften
- Tidigare kommandon lagras i stack och kan nås via piltangenterna \uparrow och \downarrow .
- Kom ihåg att operatorerna normalt är definierade som matrisoperationer, vill man ha elementvis applikation av en operator så sker det genom att operatören föregås av en punkt (`.* ./ .^`)
- Dimensioner är allt! Varje enskild operation (även tilldelning) måste vara rätt vad gäller dimensioner, även i MATLAB är dock skalärer undantag (trots att de kan betraktas som 1*1-matriser)

Baskommandon**(Sidhänvisningar till Användarhandledning för MATLAB 6.5)**

1. Mata in matriserna (s.26-27)

$$A = \begin{bmatrix} 1 & 4 & 0 \\ 6 & 3 & 1 \\ -1 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 1 & -4 \\ -1 & 5 & 19 \\ 2 & -6 & -5 \end{bmatrix} \text{ och } v = \begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix}$$

2. Beräkna produkterna $2A$, AB och $A \cdot *B$ och observera skillnaden (s. 48,52)
3. Beräkna B^T (s.51)
4. Beräkna lösningen till $Ax = v$ utan att beräkna A^{-1} (s.121)
5. Lägg in v som andra kolumn i A (s.74)
6. Lägg in v som tredje rad i B (se till att matrisdimensionerna är rätt) (s.74)
7. Bilda en radvektor x med värdena $0, \pi/6, 2\pi/6, \dots, 2\pi$. Kontrollera storleken på x med kommandot `size` (s. 30,72)
8. Bilda en kolumnvektor t med 200 element mellan 0 och 4π (s.72,73)
9. På vilket sätt skiljer sig `length` från `size`?

10. Bilda matrisen $D = \begin{bmatrix} 11 & 9 & 9 \\ 9 & 11 & 9 \\ 9 & 9 & 11 \end{bmatrix}$ utan att mata in elementen ett i taget (avsnitt 4.1)

Övningar (Med tillstånd av Gerd Eriksson NADA, KTH)

11. Funktionskurvor

Rita kurvor för funktionerna

$$f(x) = x(1 + \sin \pi x) \text{ och}$$

$$g(x) = \frac{5e^{-x/2}}{3 - 2\cos 2\pi x} \text{ i intervallet}$$

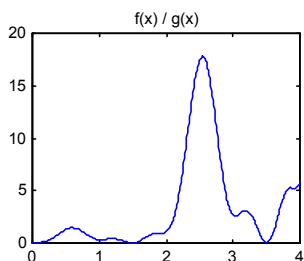
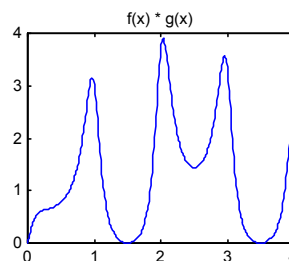
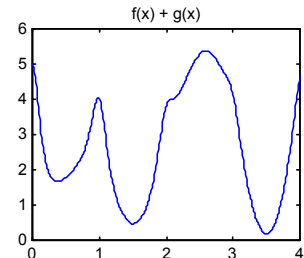
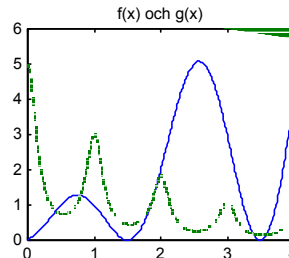
$$0 \leq x \leq 4.$$

Prova vad som kan vara lagom avstånd mellan punkterna.

Rita med hjälp av `subplot` även kurvor för

$$f(x) + g(x), f(x) \cdot g(x) \text{ och}$$

$$f(x) / g(x).$$



Användbara Matlab -kommandon

`plot` : (op) `plot` `subplot` `title`

`./` (op)

12. Ytterligare en funktionskurva

Plotta kurvan $f(x) = \frac{10}{\sqrt{1+x^2}} + \frac{e^{x/2}}{\sqrt{2+\sin \pi x}} + \frac{4}{x-5}$ i intervallet $-2 \leq x \leq 4$, med ett lagom litet steg så att kurvan inte ser kantig ut.

Användbara Matlab -kommandon

`.^` (op) `exp` `grid`

13. Cirkel

Cirklar ritas bäst på parameterform: $x = x_c + R \cos \varphi$, $y = y_c + R \sin \varphi$ där vinkeln går från 0 till 2π . Rita en cirkel som har radien 3 och origo i $(0, 1.42)$. Använd steget $2\pi/60$. Markera mittpunkten. För att undvika en oval cirkel i plotten använder man `axis equal`.

Användbara Matlab -kommandon

`pi` `axis equal` `plot(x,y,'o')`

14. Tre tangerande cirklar

Låt tre tangerande cirklar ha centrum i i punkterna

$x = [0 \ 4 \ 6.5]'$; $y = [1.42 \ 6.18 \ 4.75]'$; Bestäm deras radier så att de tangerar varandra och rita upp triangeln samt de tre cirklarna.

Ledning: Sambanden $r_1 + r_2 = s_1$, $r_2 + r_3 = s_2$ och $r_3 + r_1 = s_3$ mellan radier och triangelsidor skrivs i MATLAB enklast på matris-vektor-form (bara ettor och nollor i matrisen). Lös med $r = A \setminus s$.

Användbara Matlab –kommandon

\ (op) clear clf hold on diff
 ^ . (op) sqrt for fill

15. Pilkastning

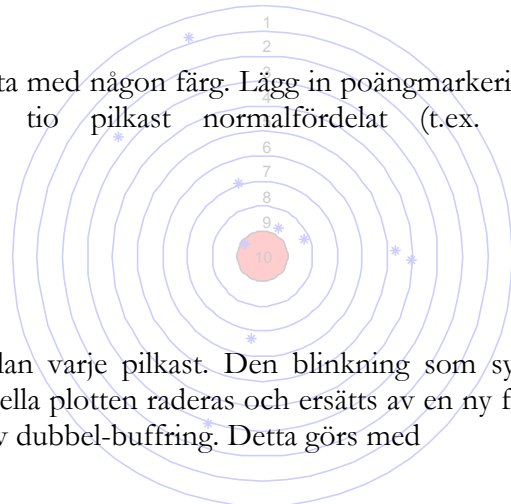
Rita tio cirklar med radie 1,2, ..., 10. Fyll den innersta med någon färg. Lägg in poängmarkeringar på piltavlan med `text`-funktionen. Slumpa tio pilkast normalfördelat (t.ex. med standardavvikelsen 5 i x-led och 4 i y-led):

```
for pil=1:10
    plot(5*randn,4*randn,'*')
    pause(0.7)
end
```

`pause`-satsen ger 0.7 sekunders fördröjning mellan varje pilkast. Den blinkning som syns i grafikfönstret vid varje pilkast beror på att den aktuella plotten raderas och ersätts av en ny figur. Du kan slippa blinkningen genom att använda dig av dubbel-buffring. Detta görs med

```
set(gcf,'DoubleBuffer','on')
```

Egenskapen `DoubleBuffer` styr hur grafiken ska uppdateras hos grafikobjektet `figure` i MATLAB. Beräkna poängen för varje kast, summera och skriv ut totalpoängen. Exekvera flera gånger. Simulera sedan en sämre pilkastare genom att öka standardavvikelsen (se `randn`)



Användbara Matlab -kommandon

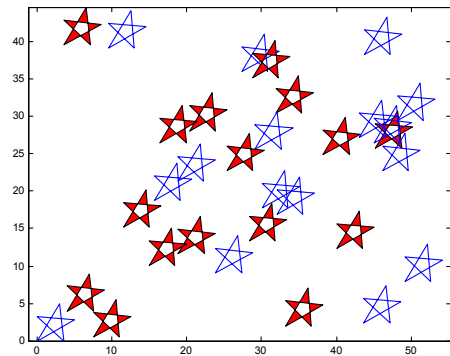
pause for randn (gcf
 set(gcf if - else axis off text

16. Femuddiga stjärnor femudd.m

Femuddstjärnor kan man ju rita utan att lyfta på pennan. Rita en sådan i MATLAB. Låt en udd ligga i origo och resten på lämpliga koordinatvärden.

Rita också en femudd som är förskjuten fyra enheter i både x och y-led. Använd `fill` istället för `plot` för denna.

Gör sedan en stjärnhimmel genom att slumpa ut 15 fyllda och 15 ofyllda stjärnor, likformigt fördelat i området $0 \leq x \leq 60$ och $0 \leq y \leq 45$



Användbara MATLAB -kommandon

`pause`

17. Klicka in punkter

Modifiera uppgiften med tre cirklar så att användaren får ange tre önskade hörnpunkter till en triangel. Detta skall ske genom att användaren klickar i grafikfönstret. Börja med ett `axis`-kommando med lämpliga värden för x- och y-axlarna, t.ex.

```
clear, clf, axis([0 10 0 8]), hold on
```

Klickning av tre punkter kan göras med `[x,y]=ginput(3); plot(x,y,'o')`. Nackdelen är att punkterna inte markeras förrän alla tre matats in. Prova! Det är bättre att klicka in en punkt i taget, markera den och uppdatera x och y vektorerna.

```
x=[]; y=[];
for i=1:3
    [xp,yp]=ginput(1); plot(xp,yp,'*'); x=[x; xp]; y=[y; yp];
end
```

Användbara MATLAB -kommandon

`ginput` `axis` `disp` `clf`

18. Polynom genom givna punkter polynom.m

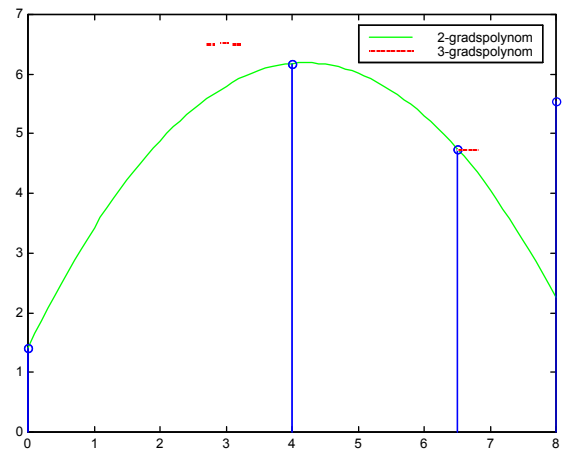
Genom fyra givna punkter går ett entydigt bestämt tredjegradspolynom. Den enklaste ansatsen för att representera ett tredjegradspolynom är $P(x) = c_0 + c_1x + c_2x^2 + c_3x^3$

Givet fyra punkter (x_i, y_i) $i = 1, \dots, 4$ leder detta till ett ekvationssystem $Ac = y$ för bestämning av de fyra koefficienterna c_0, c_1, c_2, c_3 .

Systemmatrisen kan skrivas $A = [\text{ones}(\text{size}(x)) \quad x \quad x.^2 \quad x.^3]$ där x är en kolumnvektor med x -värdena.

a) Till att börja med har vi bara tillgång till tre punkter, nämligen triangelhörnen i uppgift 4. Beräkna och rita den entydigt bestämda andragradskurva (parabel) som satisfierar punkterna.

b) Lägg till punkten $(8, 5.55)$ och beräkna och rita tredjegradspolynomet som passerar de fyra punkterna. Punkterna skall naturligtvis också markeras i plotten, prova hur `stem(x, y)` fungerar.



Användbara MATLAB -kommandon

ones stem xlabel legend figure(2)

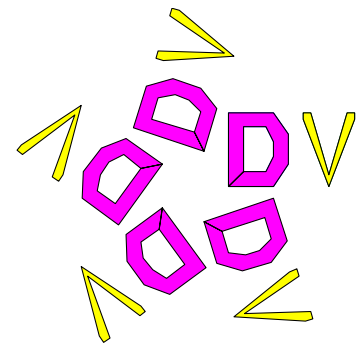
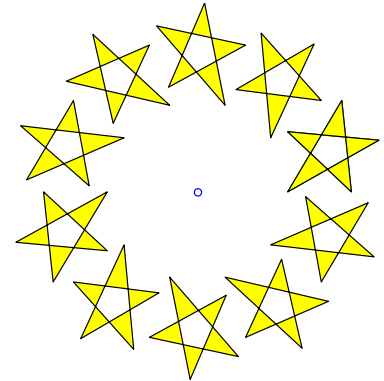
19. Vridna figurer

Med hjälp av en transformationsmatris $S = \begin{pmatrix} \cos(v) & -\sin(v) \\ \sin(v) & \cos(v) \end{pmatrix}$ kan man enkelt rotera figurer.

Här startar vi med femudden från tidigare, flyttar den fem enheter åt höger samt roterar allt kring origo.

```
% femuddsnurr
clear, clf
x=5+[0 5 0 3.5 3 0];
y=[0 2.8 3.3 0 5 0];

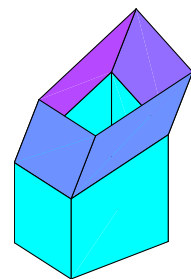
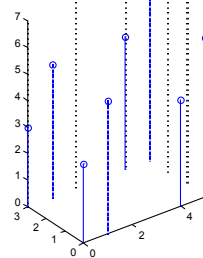
plot(0,0,'o', x,y)
axis equal, hold on, axis off
n=10; v=2*pi/n;
S=[cos(v) -sin(v);
   sin(v)  cos(v)];
for k=1:n
    P=S*[x;y];
    x=P(1,:);
    y=P(2,:);
    fill(x,y,'y')
end
```



Modifiera mallen så att en fylld stor bokstav vrids i n steg, där n läses in från kommandofönstret.

20. Sned låda, 3D-ritning

Den sneda lådan i den högra figuren är konstruerad av stolparna till vänster som ritas med `stem3(x,y,z)`. Rektangeln i botten är fyra enheter i x-led och tre enheter i y-led. De heldragna stolphöjderna är `[3 5 5 3]`. De streckade stolparna har förflyttats en enhet i x-led och höjderna ökats med två enheter. Rita figuren med `surf`-kommandot.



Användbara MATLAB-kommandon



stem3



surf



colormap



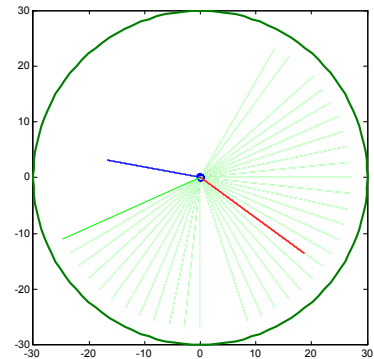
rotate3d

21. Klocka

Konstruera en klocka med cirkel-formad urtavla och färgglada tim-, minut- och sekundvisare.

Du behöver inte sätta ut markeringar på urtavlan. Rita om visarna varje hel sekund, utnyttja `drawnow` efter `plot`-satsen. MATLAB har en funktion `clock` som retrurnerar en vektor med följande innehåll `[year month day hour minute seconds]`, där de fem första är heltal och den sista, `seconds`, har några få decimaler. Detta kan man lösa med `fix(clock)`. För att slippa flimmar kan du utnyttja samma teknik som i pilkastningsuppgiften.

Tänk på att låta timvisaren följa minutvisaren så att den inte "skuttar" fram en timme i taget.



Användbara MATLAB -kommandon

`clock` `fix` `plot(..., 'Linewidth'` `plot(..., 'b`