MatLab

# Föreläsning 2

Det finns många olika variabeltyper i Matlab

Ex: p=3; Skalär variabel

Det finns också (väldigt vanligt förekommande) vektorer.

x=[0 2 4]; (3x1 matris)

Detta blir en radvektor med 3 element:

x(1)=0

x(2)=2

x(3)=4

Vi kan också använda ” ,” för att separera värden.

3,5 två element, 3 och 5

3.5 betyder 3 komma fem

(Om man vill testa kommandona på tavlan och spara dem för framtiden ”diary lecture2”)

En kolonnvektor ser ut såhär:

b=[3;2;1] (Semikolon ; betyder att man går ner en rad)

b= (1x3 matris)

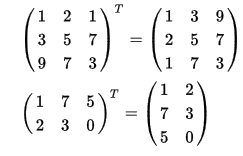
3

2

1

Fortfarande en vektor med 3 element, fortfarnde fås elementen med b(1), b(2) osv.

x\*b=8

x.\*b=fel

x.\*b’= 0 4 4 (b’ är transponaten av matrisen b)

I ett fall använder vi matrismultiplikationsregler:

3x-2y=0

x+y=2

3 -2 x 0

1 1 y 2

A=[3,-2;1,1]

B=[0;2]

xy=

A\B (Använd bara \ Backslash)

b=[0;2]

eller

b=[0 2]’

A=[1 2 3; 4 5 6; 3 1 9]

A(2,2) = 5

a=1:6;

>> b=8:11;

>> ab=[a b]; %konkatenera lista/vector

a=(1 2 3 4 5 6) b=(8 9 10 11)

>> c=1:11; %Skapar lista/vekor med ”1-steg”

>> c(ab)=[]; %Ta bort element genom tilldelning av tom []: “I listan/vektorn c, strykbort elementen som finns i listan ab”

>> c

c =

7

Vi kan också se på en vector som en lista med tal helt en lista med tal helt enkelt.

x= -2:0.1:2 Lista med x-punkter

y=x.\*x Lista med motsvarande x^2 värden

plot(x,y)

figure = öppnar ett nytt figurfönster istället för att skriva över det gamla.

# Föreläsning 3

* Plotta snyggt
* Logiska operationer
* If, while, for, break

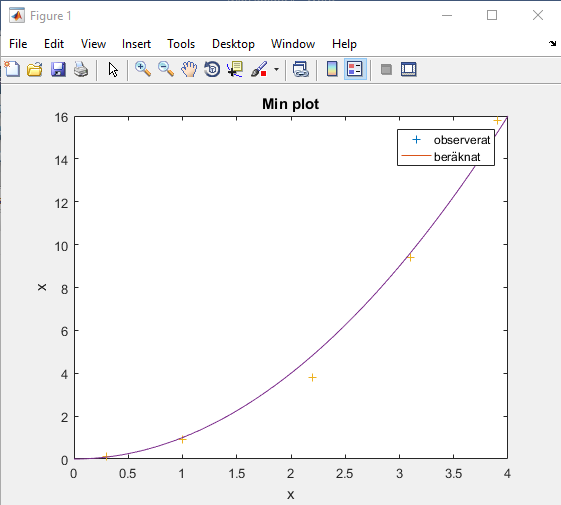
Att plotta: Exempel från avsnitt 5.1 i boken

X=[0.3 1.0 2.2 3.1 3.9]

Y=[0.1 0.9 3.8 9.4 15.8]

Hitta en matematisk modell som passar: finns funktioner för detta i MatLab, men nu kör vi ögonmått.

Ett tips (från boken): y=x^2. Vi har det som hypotes och testar.

Skapar script-fil: plotta x2.m

%Plottax2

X=[0.3 1.0 2.2 3.1 3.9];

Y=[0.1 0.9 3.8 9.4 15.8];

plot(X,Y, '+')

x=linspace(0,4,100); (100 är antal element)

y=x.^2;

hold on; %Betyder "rita i samma fönster

plot(x,y) (Defeult är heldragen linje)

legend('observerat','beräknat') (Det lilla fönstret uppe till höger)

xlabel('x')

ylabel('y')

title('Min plot')

(help plot): man kan ha upp till tre tecken i sitt tredje argument i plot-rutinen:

Färg Markör Linjetyp

(b,g,r,c-cyan,m,y,k-svart,w) (.,0,x,\*,s,d(diamant),v, >, <,p,h) (-,:(prickar),-.,--)

t.ex. plot(X,Y,’cd:’)

FORMATTERAD UTSKRIFT TILL COMMAND WINDOW

disp(’Hej!’ Resultatet är:’)

Ofta vill man ju också få med ett resultat

age=33;

Då är det FEL att skriva:

disp(’Jag är 33år gammal!’)

Så vi måste få in siffervärdet i vår disp!

1. Gör om variabeln ”age” till en textsträng
2. Konkatenera textsträngar

disp([’Jag är’ num2str(age) ’år gammal!’])

num2str = ”förvandla” numerisk variabel till text. (finns även något som heter ”fprintf”)

x=linspace(-3,3,200);

y=zeros(size(x));

for i=1:length(x)

if x(i) <= 0

y(i)=0;

elseif x(i) < 2

y(i)=1;

else

y(i)=0;

end

end

plot(x,y)

Logiska operationer:

Är a större än b? T.ex. MatLab svarar antingen 1 eller 0.

Jämförelseoperatorer

< Mindre än

<= Mindre än eller lika

== lika

>= Större än eller lika

~= inte lika med

”Kombinationsoperatorer”

&,&& Och

|,|| Eller

~ Inte

any Något element skiljt från 0 Ex. a<b || c>a; any([a>b c>a]); sin(x) [0 1]

all alla element skiljt från 0

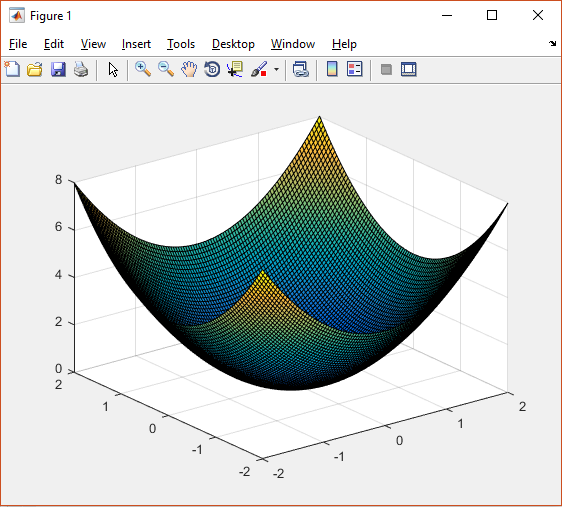
# Föreläsning 5

Programutveckling

* Förarbete
  + Definition: Formulera – vad är givet/sökt? Ambitionsnivå?
  + Strukturera: Dela upp i delproblem, funktioner
  + Algoritm: Beskriv lösning med ord eller flödesschema
* Programskrivning
  + Implementering: ”Översätt till MatLab”
  + Dokumentera: Beskriv ingående delar + övergripande målet
  + Träna koden!: Kan ni få till fel? Försök!

Struktur (3D-plot):

* Funktion: z=x.^2+y.^2
* 3D-plot verktyg: surf
* Indata: En matris Z och distanser x & y



%3D-plot

%I huvudprogram: ex5\_1.m

x=linspace(-2,2);

y=linspace(-2,2);

z=squareXY(x,y);

figure;

surf(x,y,z)

%I funktionsfil squareXY.m

function z=squareXY(x,y)

z=zeros(length(x),length(y));

for i=1:length(x)

for j=1:length(y)

z(i,j)=x(i)^2 + y(j)^2;

end

end

Hur snabbt går det i matlab?:

tic;

factorial(1000);

toc

Ofta bra att köra många gånger för stabila resultat

# Föreläsning 6

* Effektiv programmering
* PE5
* Inlämningsuppgift 2

Programeffektivitet (Kap.8 i boken)

Givet två lösningar på ett programeringsproblem så väljer vi den snabbare

Ex. Linjärt ekv.system

X1+3x2+4x3=7

X1-2x2+x3=0

X1-3x2-2x3=-15

Genom att definiera koefficientmatrisen och högerledet kan vi skriva detta på matrisform som:

Ax=B (Linjär algebra)

Formellt sett löses detta problem x=A-1B

Två sätt att lösa i MatLab:

Varför får man noll poäng om man använder inv?

Jo, om man har ett större problem är inv sämre numeriskt (dvs. det blir mer fel) och tar längre tid.

# Föreläsning 7

* Läsa och skriva filer
* Inlämningsuppgift 2
* Veckans problem

%T2013.3

fid=fopen('Allsvenskan2007.txt','r'); %r = read

Rubriker=fgetl(fid); %Skapar en textvariabel med rubriker, fgetL läser den första olästa raden i filen

Streck=fgetl(fid); %Läser andra raden ("Första olästa")

c=fgetl(fid);

a=1;

while c~=-1 %När det inte finns några rader kvar svarar fgetl "-1"

tabellIText(a,:)=c;

malSkillnad(a)=str2double(tabellIText(a,end-6:end-4)); %Hitta relevant data

c=fgetl(fid);

a=a+1

end

fclose(fid); %Har du fopen måste du ha fclose!

[~,I]=sort(malSkillnad,'descend') %flagga som avgör stigande eller fallande

% ~ innebär att utvariabel inte eftersökes

% [A,I]=sort(x); x=[3;1;2]

% x(I)=A I=[2;3;1] A=[1;2;3]

%Skriva till ny fil

fid2=fopen('Thernsvenskan2007.txt', 'w');

fprintf(fid2,[Rubriker '\n']);

fprintf(fid2,[Streck '\n'];

for a=1:length(malSkillnad)

fprintf(fid2,[tabellIText(I(a),:),'\n']);

end

fclose(fid2);

Formatkoder:

Används I fprintf och sprint för att få in värden i text.

fprintf: Skriver till fil eller CommandWindow

sprintf: Skriver till en textsträng (bra om man vill få in

text i en figur)

fprintf(‘Här är ett heltal: %i \n’,1) Utan ”fid” skriver vi till CommandWindow. i som i interger

olika lucktyper (conversion types)

%d %i heltal

%u heltal utan tecken

%f flyttal (fixed point)

%e exponentform (typ 3,14159e+00)

%s textsträng

Specialtecken

%% -> %

\\ -> \

\t=’tab’

%-s -> vänsterjusterad text

# Föreläsnning 8

* Funktionshandtag
* Numeriska metoder
* Veckans problem

Funktionshandtag (Behövs för att anropa en funktion med en funktion)

function y=myfun(x)

y=x.^2+sin(x)-32+18.\*x;

%För att få tag på den skapar vi sedan ett funktionhandtag

%I command window

f=@myfun;

f(0),f(1),f(2.34)

%Å vad smidigt!

%Syntax för olika variabeltyper

%Vektor: a(2) (Position i lista)

%Textsträng: mintext(2) (Position i lista)

%funktionsgandtag: f(2) (Precis som i funktionsfil)

Mer formellt definieras funktionshandtag

g=@sin;

g=@(t)(sin(t));

Men en matematisk funktion är oftast bara en rad. Vi kan definiera den i handtaget!

Lösa matematiska problem i Matlab:

(Vanliga problem: matrisalgebra x=A\b) (kap9)

Vad vill vi göra med funktioner?

* Nollställen (fzero)
* Extrempunkter (fminunc) (fmincor)
* Integrera (integral) äldre(quad)
* Interpolera (Interp1) (spline) (polyfit)
* Polynom (polyval, roots, polyder)är

%Skärningspunkt

clear all

clc

close all

f=@(x)(0.01.\*x.^(3)-0.02.\*x.^2-0.5);

g=@(x)(1-exp(1).^(-2.\*x)).\*sin(4.\*x);

fplot(f,[0,10],'k')

hold on

fplot(g,[0,10],'r--')

h=@(x)(f(x)-g(x)); %Där h==0 är f(x)=g(x)

xsol=fzero(h,5);

plot(xsol,f(xsol),'xb');

legend('f(x)','g(x)','f=g','Location','Best');

xlabel('x');

# Föreläsning 9

* Cellmatriser
* Strukturer

**Cellmatriser**

Används för att kunna blanda olika datatyper, eller texter med olika längd.

* En cellmatris kan blanda datatyper och vektorer av olika längd
* Deklareras med { }
* A{2} Hämtar element (ger datatypen i andra elementet i A)
* A(2) Tittar på element 2 (ger en cell)

Ex:

linjeTyp={'b-','r--','k:'};

hold on;

for a=1:3

fplot(@(x)sin(a.\*x),[0 2\*pi],linjeTyp{a});

end

Skillnad på linjeTyp{1} och linjeTyp{1}! Den förra blir en cell, den senare en textsträng.

Samma exempel igen, fast variant:

funktionsData={@(x)sin(x),@(x)sin(2\*x),@(x)sin(3\*x);

'b-', 'r--', 'k:'};

hold on;

for a=1:3

fplot(funktionsData{1,a},[0 2\*pi], funktionsData{2,a});

end

Ytterligare ett cell-exempel: Strukturerad Legend

clear variables

x=linspace(0,2\*pi,1000);

a=2:4:10; %2,6,10

legendString=cell(length(a),1);

for n=1:length(a)

y(n,:)=sin(a(n).\*x);

legendString{n}=sprintf('sin(%02i)',a(n)); %02="Zeropaddin" nollorframfor 2=storleken pa luckan

end

plot(x,y,'LineWidth',2);

legend(legendString,'Location','SouthWest');

**Strukturer**

Väldigt bra sätt att hålla reda på data som narturligt hör ihop. (s.78-80)

* Ett sätt att lagra data på hierarkiskt vis i en enda variabel
* En struktur består av s k fält vari en lagrar tillhörande data
* En kan göra listor av strukturer på samma sätt som med tal typ

Exempel: Persondatabas

person(1).name='Anders T. Johansson';

person(1).age=33;

person(1).title='Doktor';

%Strukturens namn = person, fältnamn = efter.

person(2).name='Prins Carl Philip Bernadotte';

person(2).age=37;

person(2).title='Hans Kungliga Höghet';

age=[person.age]; %hakparanteser får ut lista av siffor

meanAge=mean(age);

fprintf(['%s %s och %s %s är i genomsnitt '...

'%3.1f år gamla.\n'], person(2).title, person(2).name,...

person(1).title, person(1).name, meanAge);

Det går att få ut allt genom att skriva person(1) typ (Det gjorde jag i exemplet).

person=struct('name'{Anders T. Johansson', 'Prins C P Bernadotte',...

'age',{33,37}, 'title',{'Dr', 'HKH'});

%Observera att cellmatriser alltså används för att göra struct

Exempel: Skriva ut innehållet i alla filer på skärmen

filetree=dir

%length(filetree)

%filetree(3).name

%DON'T RUN THIS

for n=1:length(filetree)

if ~filetree(n).isdir

s=fileread(filetree(n).name);

fprintf('\*\*fil.nummer %03i (%s): \n \n %s \n',...

n,filetree(n).name,s);

end

end

# Föreläsning 10

* Felkoder
* Debugging
* Inlämningsuppgift 3

När det uppstår fel i MatLab-script eller funktioner meddelas detta med ett felmeddelande. Väldigt viktigt att kunna läsa dessa!

Vi övar på att göra fel. Skapa skript myerrorneosfile

%x=(1+2\*(5+4))/(3+4)) %för många paranteser

%a=1; %The expression to the left of the equals sign is not a valid target for assignment

%if a=1 %Ska vara 2st likhetstecken

% disp(a)

%end

%a=z+1;

%Undefined function or variable z

%A=rand(1,4);

%B=rand(1,4);

%Svar=A\*B

%Inner matrix dimensions must agree, måste ha punkt framför \* eftersom

%matrispultiplikation funkar inte med två vektorer

%Svar=A(-1)

%Subscript must be posotive intergers or logicls

Vid felsökning är debugverktyget bra som hjälp