Laborationer i kursmomentet Datoranvändning E1

Laboration nr 5: Matteverktyget MATLAB

92-11-05, 93-11-01, 94-10-31, 95-10-09, 96-10-31, 97-11-13 Magnus Bondesson
Uppdatering 98-09-26 Thomas Hallgren

1 Introduktion

Syftet med denna laboration är att ge dig en första praktisk kontakt med matematikverktyget MATLAB. Precis som i Laboration nr 3 (som handlade om Maple) avser vissa av uppgifterna dessutom att antyda hur dessa verktyg kan användas för att ge ökad förståelse kring något fenomen.

Som förberedelse måste du ha läst igenom MATLAB-kapitlet i Gula Boken (hoppa som vanligt över det som verkar konstigt). Gör uppgifterna i ordning, i första hand de uppgifter som är *-märkta. Några av uppgifterna utan * kräver kanske lite grundläggande kunskaper i programmering.

Det är lätt att blanda ihop Maples beteckningar med MATLABs. Observera speciellt att tilldelningstecknet i MATLAB är "=" och att kommandon normalt inte avslutas med "`;`" ("`;`" undertrycker utskrift av resultatet). π heter `pi` och inte som i Maple `Pi`, etc. MATLAB arbetar till skillnad mot Maple inte i ett eget fönster, utan i ett vanligt terminalfönster. Man kan därför inte använda musen vid redigering, däremot fungerar piltangenterna. Figurer i egna fönster (man kan ha flera, men normalt återanvändes ett enda automatiskt).

2 Uppgifter

*Uppgift 1. Starta MATLAB

Det är lämpligt — men inte nödvändigt — att först skapa en katalog (mapp) `matlab`, vilket du gör med UNIX-kommandot `mkdir matlab`. Starta sedan MATLAB med t ex ett programmeny-kommando.

*Uppgift 2. En inblick i MATLABs värld

Innan vi börjar med uppgifter som illustrerar de mer grundläggande funktionerna i MATLAB kan det kanske vara kul att ta en titt på lite mer avancerade exempel för att få en känsla för vad man kan använda MATLAB till.

Till MATLAB hör en massa demonstrationsprogram, som är skrivna i MATLAB (källkoden är tillgänglig). Du kommer in i demonstrationsdelen genom att ge kommandot *demo*. Gör det och pröva t ex Visualization/2D Plots, Visualization/3D Plots, Gallery/Hoops, Games/Minesweeper och Visualization/Lorenz attractor animation.

***Uppgift 3. Vektorer, ritning och redigering**

Vi skall rita upp kurvan $sin(x)$ på intervallet [0,16].

a. Bilda en vektor $x = [0, 1.0, 2.0, ..., 16]$ med litet skrivarbete.

b. Bilda en vektor *y* med motsvarande sinus-värden.

c. Rita kurvan med *plot(x,y)*.

d. Resultatet blev inte alldeles lyckat eftersom vår diskretisering var väl grov. Ändra med piltangenternas hjälp (se sid 169) så att $x = [0, 0.1, 0.2, ..., 16]$ i stället samt — även detta med piltangenterna — bilda motsvarande vektor *y* och rita på nytt.

***Uppgift 4. Addition och slumptal**

Rita upp kurvan $sin(x)$ störd med slumpmässigt brus i

a. intervallet (0.0,0.4)

b. intervallet (-0.2,0.2)

på samma intervall som i Uppgift 3, dvs lägg till varje sinus-värde ett slumpmässigt tal i det angivna intervallet. Vektorn *x* har du kvar sedan förra uppgiften, så skriv bara ett *plot*-kommando. **Ledning**: Slumptalsfunktionen *rand* beskrivs i Gula Boken, sid 170.

Uppgift 5. Vektorer och Lissajous-kurvor

Rita upp kurvan $x = cos 4t$, $y = sin(3t)$ på *t*-intervallet $[0,2\pi]$. Konstanten *pi* finns fördefinierad.

***Uppgift 6. Funktioner som arbetar på vektorer**

Betrakta uttrycket $\frac{4}{1+x^2}$.

a. Beräkna värdet för x=0.5.

b. Beräkna i ett svep uttryckets värde för $x = 0, 0.1, 0.2, ..., 1.0$.

c. Rita upp motsvarande kurva från $x = 0$ till $x = 1$.

***Uppgift 7. Rötter**

Bestäm rötterna till 16:e-gradsekvationen $17x^{16} + 16x^{15} + ... + 3x^2 + 2x + 1 = 0$. Rita upp dem i ett diagram (markera varje rot med + eller *). **Ledning**: Funktionen *roots* beskrivs i Gula Boken, sid 172. Om du händelsevis hann med Uppgift 24 i Laboration nr 3, notera hur mycket enklare detta problem löses i MATLAB än i Maple.

Uppgift 8. Rotvandring

Studera hur rötterna vandrar när nolltegrads-termen i förra uppgiften avtar från 1 till -1, genom att i samma diagram rita in rötterna (med markering `.`) när nolltegradstermen är 1, 0.9, 0.8, ..., –0.9, -1.0. Gör först *hold on* för att få ritning i samma diagram och ändra sedan nolltegradstermen successivt i en `FOR`-snurra (eller arbeta manuellt). Vad händer med de två rötter som ursprungligen låg längst till vänster i diagrammet? Hur många reella rötter har utgångsekvationen respektive slutekvationen? Gör *clf* (av Clear Figure) eller *hold off* innan du fortsätter med en ny uppgift.

Uppgift 9. Bild från MATLAB fogas in i FrameMaker eller HTML-dokument

Spara diagrammet från någon av de två föregående uppgifterna som en PostScript-fil `MATTE.epsi` (se sid 180). Kontrollera resultatet med t ex

```
ghostview MATTE.epsi.
```

Anm. Från Laboration nr 4 vet vi hur bilder kan importeras till FrameMaker. Även PostScript-bilder kan importeras.

När det gäller HTML är det numera en aning värre. Förut kunde man från MATLAB spara på GIF-format, som ju hanteras direkt av webbläsare, men man tycks ha övergivit det formatet för gott. I Gula Boken (sid 181 överst) står hur man kan göra. Det sättet fungerade inte när texten skrevs. Nu går det, men dels blir PPM-filen väldigt stor, dels utvidgas bilden till A4-format. Bättre är att i MATLAB skriva

```
print -dppmraw filnamn
```

och att senare med `xpaint` eller `xv` ta bort onödig utfyllnad i `filnamn.ppm` och till sist spara på GIF-format. I version 5.1 (som ännu inte finns på E) kan man även spara på JPEG-format - som webbläsarna ju också klarar - med (*help print* ger detaljerna) *print -djpeg filnamn*. Bilden blir dock stor (som A4).

*Uppgift 10. Komplexa tal

Låt $z = 1 + 2i$. Beräkna $z^3 + z^5$. Beräkna sedan i ett svep detta uttryck för $z = 1 + 2i, 1 + 3i, ..., 1 + 9i$.

Uppgift 11. Komplexa vektorer

Låt z vara en vektor av punkter som ligger på enhetscirkeln i det komplexa talplanet. En sådan vektor kan bildas t ex så här:

```
t=0:0.1:2*pi; z=cos(t)+i*sin(t).
```

Var hamnar då de punkter som utgörs av $w = \frac{1-z}{1+z}$? Vill du visualisera resultatet är det lämpligt att förfara enligt (>> är redotecknet som ej skall skrivas):

```
>> axis([-10,10,-10,10]);  % Skala på axlarna
>> plot(z,'.')             % Rita punkterna på cirkeln
>> hold on                 % Rita utan att sudda föregående
>> plot(uttryck för w,'.') % Rita de nya punkterna
```

```
>> clf                          % Innan du går vidare
```

Svar: ____________________________

Uppgift 12. Summation

Beräkna följande summor numeriskt. (**Ledning**: *sum*, se sid 171.)

a. $\sum_{k=1}^{60} \frac{1}{k^4}$. Jämför gärna med summan $\frac{\pi^4}{90}$ för motsvarande oändliga serie.

b. $\sum_{k=1}^{30} sin\frac{1}{k}$. Svar: ____________________

***Uppgift 13. Egna funktioner. Integration.**

a. Tag med hjälp av *help* eller *helpdesk* reda på vad verktygslådefunktionen *quad* gör (läs bara de 5 första raderna). Beräkna sedan en approximation till integralen $\int_0^{\pi/2} cos x dx$ och jämför med det exakta värdet 1.

b. Skriv en MATLAB-funktion *first*, dvs gör en m-fil (sid 183) `first.m` utanför MATLAB med ditt favoritredigeringsprogram (t ex NEdit), som beräknar $\frac{4}{1+x^2}$ för en **godtycklig vektor** *x*. Spara lämpligen i katalogen `matlab` som du skapade inledningsvis. En sådan funktion har formen

```
function y=first(x)
y=?????
```

Testa den och verifiera att den ger rätt resultat.

c. Beräkna nu med *quad* en approximation till integralen $\int_0^1 \frac{4}{1+x^2}dx$ och jämför med det exakta värdet som ju är π.

Uppgift 14. Egna funktioner. Ekvationslösning.

a. Tag med hjälp av *help* reda på vad verktygslådefunktionen *fsolve* gör. Beräkna sedan en approximation till det nollställe till $cos x$ som ligger i närheten av 1.5. Jämför med det exakta värdet som ju är $\pi/2$.

b. Funktionen $5xe^{-x} - 1$ har två reella nollställen på intervallet $[0,5]$. Visa det genom att rita upp funktionen.

c. Skriv en MATLAB-funktion *second*, dvs gör en m-fil `second.m` utanför MATLAB med ett redigeringsprogram, som beräknar funktionsvärdena för en godtycklig vektor. En sådan funktion har formen

```
function y=second(x)
y=?????
```

d. Beräkna nu med *fsolve* det största av de två nollställena i b. Vad blev det? ______________

Uppgift 15. Egna funktioner. Interaktion.

Skriv en MATLAB-funktion *inter*, dvs gör en m-fil `inter.m`, som låter dig markera 10 punkter i ritfönstret (som förutsätts finnas framme) och när en punkt markerats förbinder den med närmast föregående (utom för den första). En sådan funktion har formen

```
function inter
[xold,yold]=ginput(1);
for i=1:9
      ???
end
```

där `???` står för maximalt 4 rader kod.

Gör *clf* (suddar ritfönstret) och *hold on* och testa sedan din funktion.

***Uppgift 16. Ett besök till i fraktalernas underbara värld**

Låt oss välja en punkt (x_0,y_0) i x,y-planet och sedan bilda nya punkter enligt formeln,

$$\begin{cases} x_{n+1} = 1 - y_n + |x_n| \\ y_{n+1} = x_n \end{cases}$$

för $n = 0, 1, 2, \ldots$. Var hamnar dessa punkter? Följande funktion i form av en m-fil gör 1000 iterationer utifrån en startpunkt och ritar samtliga punkter. Skriv in funktionen (texten som inleds med `%` kan du strunta i) med valfritt redigeringsprogram och spara den som filen `ginger.m` i din katalog `matlab`.

```
% GINGER(x,y) draws PEPPARKAKSGUBBEN if e g x=-1.3,y=2
function ginger(x,y)
clf                        %erase previous figure and forget hold
axis([-10,10,-10,10])      %our coordinate system
plot([-10,10],[0,0])       %draw x-axis
hold on                    %never erase,but autoscale still in effect
plot([0,0],[-10,10])       %draw y-axis
for i=1:1000               %iterate and plot 1000 times
    x1=1-y+abs(x);y=x;x=x1;
    plot(x,y,'.')
end
```

Anropa den med *ginger(-1.3,2)*. Experimentera med andra startvärden. Resten är klar överkurs. Ändra i funktionen *abs(...)* till *sqrt(abs(...))* eller *sin(...)*. Gör eventuellt om funktionen så att startpunkt väljs interaktivt med musen (jämför med föregående uppgift).

Uppgift 17. 3D-ritning

Hitta på en trevlig funktion f(x,y) av två variabler. Skriv den som en m-fil och rita upp den med specialfunktionen *myplot3d* på ett uttryck (se sid 175 och sid 186) eller gör stegen i den funktionen.

Uppgift 18. Även solen har sina fläckar

Följande lilla problem i MATLAB kan lätt hamna i och bli förundrad över. Säg att vi vill rita upp roten till ekvationen $x^2 - 1 = 0$. Vi vet vid det här laget att det kan åstadkommas med *plot(roots([1,0,-1]),'*')*. Pröva det. Konstatera att resultatet är käpprätt åt skogen. Förklaringen kan du läsa dig till med *help plot* (andra stycket). Det egentliga skälet är att man givit *plot* olika betydelse beroende på vad första parametern är. Och det är uppenbarligen inte bra. Det här problemet dyker upp när samtliga rötter är reella. En nödlösning är att lägga till ett litet rent imaginärt tal (och hoppas att inte alla rötterna har en imaginärdel som är lika med -detta tal), dvs *plot(roots([1,0,-1])+0.001*i,'*')*. Förvissa dig om att detta fungerar.

Uppgift 19. Zooma

MATLAB har vissa interaktiva inslag som saknas i Maple (och vice versa för den delen). Rita upp kurvan från Uppgift 14b. Tag med *help zoom* reda på vad *zoom*-kommandot uträttar. Tag sedan med dess hjälp reda på ett av nollställena med litet bättre noggrannhet än vad den ursprungliga figuren tillåter.

***Uppgift 20. WWW-hjälp**

Dokumentationen till MATLAB finns även som WWW-sidor (se sid 177). Tag den vägen reda på information om *ezplot*-kommandot.

Uppgift 21. Rotvandring med hemmagjord funktion

De flesta har nog inte hunnit med någon av de tidigare förekommande rotvandringsproblemen. Själva grundidén med denna typ av uppgift är att man vill visualisera hur ändringar i indata påverkar resultaten i fall där beroendet är komplicerat. T ex vilka rötter som påverkas mycket.

Mata in m-funktionen nedan (kommentarerna kan du ju strunta i) och kalla filen `animate_roots.m` (eller kopiera den från `~hallgren/Intro/`). Sätt dig även in i hur den fungerar.

```
function animate_roots(p,i,a)
   % p is a vector with the coeffients of a polynomial as for roots
   % i is an order, e g i=0 means the zero order coefficient
   % a is a vector with values for that coefficient
   % animate_roots will draw the roots for the different values of a
   % in a common window
   % the user has to press any key between the cases
   % E g animate_roots([4,3,2,1],1,2:0.1:3);
```

```
% will draw the roots when the first order coeff varies from 2 to 3
  clf
  hold off
  for var=a
       p(length(p)-i)=var;
       r=roots(p);
       plot(r,'+');
       hold on;
       pause
  end
  hold off
```

Pröva den sedan för det fall som nämns bland kommentarerna och för fallet i Uppgift 8.

Uppgift 22. Extra-uppgift: Program med knappar och menyer och sånt

Tydligen kan man inom MATLABs ram skapa trevliga interaktiva program. Vi skall belysa den tekniken litet grand. Mata in följande som en m-fil `USER.m` (eller kopiera från den vanliga platsen). Förklaringar har jag förhoppningsvis presenterat under föreläsningen.

```
% This is a short m-file with commands only, they are executed
% when the name of the file (.m excluded) is written as a command

% Create a window for graphics
fig = figure;
% Create a button control
h1 = uicontrol('Position',[10,20,200,60]);
set(h1,'Callback','s=33; s+2');
set(h1,'String','Tryck här');

% Create another button control for ending the experiment
h2 = uicontrol('Position',[210,20,200,60]);
set(h2,'Callback','delete(fig)');
set(h2,'String','Avsluta');

% Create a scroll-bar control; the value is written
h3 = uicontrol('Position',[150,90,200,60]);
set(h3,'Callback','s=get(h3,''value'')');
set(h3,'Style','slider');
```

Anropa den med *USER*. Pröva de tre styrorganen. Notera att hissen ger ett värde s mellan 0 och 1. Modifiera sedan den så att kurvan $y = e^{sx} - s$ eller eventuellt kurvorna $y = e^{sx}$ och $y = x$ ritas upp på intervallet när man dragit i hissen. Använd alternativt verktyget *guide* för att ändra användarsnittet.

3 MATLAB på webben

Företaget som gör MATLAB har webbadressen `www.mathworks.com`. Där kan man hitta en del reklam för MATLAB. Man kan hitta mer nyttig information genom att söka efter MATLAB i YAHOOs webkatalog på `www.yahoo.com`.