Matlabövning 2

En del av denna övning går ut på att skriva egna matlabfiler.Nedan beskrivs vad två speciella filer skall göra. Det gäller filerna

- **dynsys.m**
- **ritafun.m**

Redovisningen går sedan till så att ni antingen skriver ut filerna på papper eller skickar dem i textform i epost till
jea@math.chalmers.se
senast den 30 september. Glöm inte att i ebrevet skriva ut namnen (både för och efternamn) på dem som gjort uppgiften. Filerna skall vara testkörda så att de direkt går att använda.

1. Läs avsnitten 5.1-5.3 i handledningen om de två olika typerna av filer man kan skapa i MATLAB, m-filer. Titta speciellt på exemplen som finns där. Terminologin m-fil kommer av att den måste vara av formen *filnamn.m* . Man kan skapa sådana filer med vilken texteditor som helst. MATLAB innehåller en egen editor som man kan hitta under *File* -> *New* ->*M-Files*. Om inte Matlab hittar filen när du sparat den så se till att det i rutan *Current Directory* står det bibliotek där filen är sparad.

2. (Diskreta dynamiska system, funktionsfilen **dynsys.m**)
   Läs om slingor avsnitt 7.3 i handledningen och betrakta följande exempel där man för en given 2×2-matris A och en given kolonnvektor p beräknar $p, Ap, A^2p, ..., A^5p$. Det bildar också en vektor x och en vektor y så att

$$\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = A^{k-1}p$$

   för $k = 1, ..., 6$. Därefter ger den en tabell över x och y och en figur där punkterna $(x(k), y(k))$ ritas ut som * och sammanbinds med räta linjer.

```
»A=[1 0.4;-2 0.3];
»p=[-1 -1]';
»q=p;for k=1:6 x(k)=q(1);y(k)=q(2);q=A*q;end;disp([x;y]');plot(x,y,'*',x,y)
```

   Efter detta kan det vara problem att använda x och y igen så därför kan det vara lämpligt att frigöra dem med

```
» clear x;clear y
```

   (a) Den första uppgiften blir nu att göra en funktionsfil **dynsys.m** vilken skall anropas på formen

```
» dynsys(A,p,n)
```

   där A är en 2×2-matris, p är en kolonnvektor med två rader och n ett positivt heltal. Funktionen skall genomföra samma som exemplet ovan för den inmatade matrisen, startvektorn p och med slutpotensen 5 ersatt av en valbar slutpotens n. Dvs den skall ge motsvarande tabell och figur när man exekverar den.

(b) Testa din fil på några matriser exempelvis de som svarar mot följande problem.

Låt i fortsättningen bokstaven X associeras med ett bytesdjur och Y med rovdjur. Dessa rovdjur antas vara bytesdjuren enda fiender medan bytesdjuren också antas vara rovdjurens enda föda. Låt oss anta att X_k är avvikelsen i antalet (i tusental) bytesdjur månad k i förhållande till ett genomsnitt och Y_k avvikelsen i antalet rovdjur från ett genomsnitt samma månad. Vi antar att de båda populationerna utvecklas enligt ett samband

$$\begin{aligned} X_{k+1} &= 1.2X_k - 0.8Y_k \\ Y_{k+1} &= cX_k - 0.2Y_k \end{aligned}$$

där c är en parameter som preciseras nedan. Vi antar att när vi börjar studera processen är $X = 1$ och $Y = -1$.

Gör en tabell och en figur som visar hur utvecklingen är månad för månad de närmaste två åren i de tre fallen $c = 1.40$, $c = 1.55$ och $c = 1.70$.

3. Filen **ritafun.m**

Uppgiften handlar om att illustrera funktioner av två variabler dels genom funktionsyta dels genom nivåkurvor. Som hjälp finns nedan exempel (a)-(c) som det kan vara bra att göra först.

Själva uppgiften är att tillverka en funktionsfil **ritafun.m** som skall kunna anropas på formen

$>> ritafun(f, a, b, c, d)$

där f är en sträng i x och y och a, b, c, d är tal och som ger såväl funktionsytan som nivåkurvor för uttrycket i f uppritat på rektangeln $a \leq x \leq b$, $c \leq y \leq d$. Filen skall vara utformad så att om man skriver

$>>$ help ritafun

så får man upplysning om vad den gör. (Se 5.3 i handledningen.)

Testkör och redovisa filen.

Här följer exemplen.

(a) Detta del handlar om grafisk representation av funktioner av två variabler. För att rita funktionsytan till en funktion $z = f(x, y)$ över en rektangel $a \leq x \leq b$, $c \leq y \leq d$ gör man så här. Först måste man skapa ett rutnät i rektangeln på samma sätt som man gör en intervallindelning när det gäller funktioner av en variabel. Låt oss anta att vi vill rita funktionen $f(x, y) = x - y^2$ på rektangeln $0 \leq x \leq 3$, $-1 \leq y \leq 1$. Vi börjar med att skapa rutnätet med

$>> [x, y] = meshgrid(0 : 0.1 : 3, -1 : 0.1 : 1);$

Här är det verkligen att rekommendera att man slutar raden med ; annars skrivs en förfärlig massa ut. Därefter skall man räkna ut funktionsvärdena i rutnätspunkterna (kom ihåg att x och y kommer att behandlas som vektorer).

$>> z = x - y. * y;$

Därefter får man funktionsytan uppritad med kommandot *mesh*, kommandot *surf* ger också en bild av funktionsytan.

$>> mesh(x, y, z)$

Naturligtvis behöver man inte dela upp på olika rader som vi gjort här utan kan på en rad skriva

$>> [x, y] = meshgrid(0 : 0.1 : 3, -1 : 0.1 : 1); z = x-y.*y; mesh(x, y, z)$

När du fått upp funktionsytan i ett fönster går figuren att vrida på. Klicka på rotationspilen, sätt markören på figuren och håll någon musknapp nere.

Testa till exempel på följande funktioner.

i. $f(x, y) = x^2 + y^2$ på rektangeln $-1 \le x \le 1$, $-1 \le y \le 1$.
ii. $f(x, y) = x^2 - y^2$ på rektangeln $-1 \le x \le 1$, $-1 \le y \le 1$.
iii. $f(x, y) = x^2$ på rektangeln $-1 \le x \le 1$, $-1 \le y \le 1$.

I samband med funktioner av två variabler kan man också vara intresserad av att betrakta nivåkurvor. Det kan man få fram för funktionen $f(x, y) = x - y^2$ på rektangeln $0 \le x \le 3$, $-1 \le y \le 1$ genom kommandot *contour* så här

$>> [x, y] = meshgrid(0 : 0.1 : 3, -1 : 0.1 : 1); z = x - y.*y; contour(x, y, z)$

Man kan bestämma antalet nivåkurvor. I exemplet kan vi bestämma antalet kurvor till exempelvis 3 genom ersätta *contour(x,y,z)* med *contour(x,y,z,3)*.

Med kommandot $surfc$ kan man få en figur som innehåller både ytan och nivåkurvor

$>> [x, y] = meshgrid(0 : 0.1 : 3, -1 : 0.1 : 1); z = x - y.*y; contour(x, y, z)$

(b) MATLAB är inte symbolmanipulerande. Däremot kan man införa algebraiska uttryck med variabler som man senare ger numeriska värden och därefter bestämmer motsvarande algebraiska uttryck. Låt oss ta ett enkelt exempel. Vi vill arbeta med ett uttryck $x^2 + x + 1$. Vi inför detta i MATLAB som en "sträng" som vi kallar *f* genom

$>> f =' x. * x + x + 1'$

Observera fnuttarna (de skall vara raka)! Vill man sedan beräkna detta för ett visst numeriskt värde, säg för $x = 2$ så gör man det genom

$>> x = 2; eval(f)$

där eval är en förkortning för evaluate. Man kan också rita upp kurvan till exempel på intervallet $[-1, 1]$ genom

$>> x = -1 : 0.1 : 1; plot(x, eval(f))$

På sätt och vis kan man säga att man på detta sätt inför en egen funktion att arbeta med. Men det finns effektivare sätt att göra det. Vill man arbeta tillfälligt kan man göra det genom kommandot *inline*. Man kan istället för det tidigare definiera *f* som en funktion genom

$>> f = inline('x. * x + x + 1')$

Efter det kan vi bestämma värdet i $x = 2$ genom att skriva $f(2)$ och rita grafen genom

$>> x = -1 : 0.1 : 1; plot(x, f(x))$

(c) En funktionsfil behöver inte som resultat ha ett tal eller en vektor (se avsnitt 7.1 i handledningen). Här är exempel på en fil där man matar in ett uttryck som en sträng, anger ett intervall och där resultatet är att man får grafen uppritad på intervallet . Man skapar en fil, **ritakurva.m**, med exempelvis följande utseende.

```
function ritakurva(f,a,b)
% Funktionen ritar upp kurvan for strangen f i x pa intervallet [a,b]
h=(b-a)/30;
x=a:h:b;y=eval(f);
plot(x,y)
```

Här har vi bestämt oss redan från början för att dela in intervallet i 30 delningspunkter, man kunde förstås modifierat funktionen så att även detta kan styras.