TMV156 Inledande matematik E, 2010

# DATORÖVNING 2 — ANONYMA FUNKTIONER, FUNKTIONSGRAFER OCH LITE OPTIMERING

## 1. Syfte och mål

I den här laborationen skall du lära dig att definera och använda anononyma funktioner för att beräkna värden på matematiska funktioner som inte är fördefinerade i Matlab. Vi inför också begreppet vektorisering av funktioner som är intimt sammankopplat med elementvisa operationer och väldigt bra när man t.ex. skall rita funktionsgrafer. I laborationen får du lära dig rita snygga och informativa sådana. Den syftar också till att göra dig medveten om de fallgropar du kan ramla i vid all användning av datorgrafik. En viktig sensmoral är att datorer kan vara en stor hjälp när man vill förstå ett matematiskt (och därmed ett tekniskt) problem men att man för att kunna använda datorn på ett effektivt sätt också måste ha koll på matematiken. Kunskaperna vi förvärvar används sedan till att bättre förstå olika funktioners betydelse och att studera grafers förändringar med avseende på olika tänkbara parametrar. Den som hinner får dessutom chansen att börja studera de verktyg som finns i Matlab för att hitta funktioners optimala punkter. I den sista laborationen i den här kursen skall vi skriva egna program som gör detta åt oss.

## 2. Instruktioner

Välj plottningsintervall och skalor på $x, y$-axlarna så att figuren blir bra. Se sidan 26–27 i Adams. Vissa funktioner har parametrar, $a, b, c, ....$ Plotta då i en figur flera grafer med lämpligt valda parametervärden.

Använd "label", "title", "legend" samt olika färg och linjetyp så att figuren blir informativ.

Skapa en ny filkatalog ("directory") `Lab2` för denna övning. Gör alltid uppgifterna i script-filer. För rituppgifterna, arbeta med script-filer som innehåller kommandon av typen:

```
A=0; B=1;N=50; %Bara exempel!
f=@(x)...  %Gör alltid implicita funktioner på det här sättet!
g=@(x)...
x=linspace(A,B,N);
y1=f(x);
y2=g(x);
plot(x, y1,'go-', x, y2,'rd:')
legend('y=f(x)', 'y=g(x)')
```

Spara gärna någon figur i Matlabs eget format `.fig` så att du kan öppna den igen. Testa också att spara den till i `.eps` format ("encapsulated postscript") eller `.png` så att du kan importera den till något ordbehandlingsprogram. Notera att det går att exportera till många olika format. På så sätt kan man även manipulera figurer i rit- eller bildbehandlingsprogram om man vill.

## 3. Uppgifter

### 3.1. Anonyma funktioner.

3.1.1. *Att skapa anonyma funktioner.* Definera fyra anonyma funktioner:

```
>>g1=@(z)(z^2+z-2)
>>g2=@(y)(z^2-3*z+2)
>>f1=@(x)(g1(x)/g2(x))
>>f2=@(q)(g2(q)/g1(q))
```

*Date*: 2011–09–17, Fredrik Lindgren, Matematiska vetenskaper, Chalmers tekniska högskola.

Vilka matematiska funktioner definerar $f1(x)$ och $f2(x)$? Vad har de för definitionsmängd? Testa och anropa dem med några olika värden, i synnerhet $x = -2, -1, 1, 2$. Händer det som borde hända?

Notera att namnen på inparametrarna i de olika anonyma funktionerna är olika: z, y, x respektive q. Vilken funktion fyller dessa namn? Spelar det någon roll att de är olika? Antag att vi vill ändra namnet på inparametern i funktionen med funktionshandtag `g1` till w. Hur deklarerar vi då `g1`? Behöver vi ändra någon av de andra funktionerna efter att vi har ändrat `g1`?

3.1.2. *Vektorisera de anonyma funktionerna.* Skapa en vektor `x1=a:dx:b` för några lämpliga värden på $a$, $dx$ och $b$ och anropa de anonyma funktionerna från den förra uppgiften med denna. Du borde då få ett felmeddelande. Detta beror på att de inte är *vektoriserade*. En vektoriserad funktion kan ta en vextor $x = (x_1, x_2, \ldots, x_n)$ och om man skriver `y=f(x)` så är $y_i = f(x_i)$ (notera att vi här använder matematiska notation, inte Matlabnotation. I det senare fallet skulle vi skriva `y(i)=f(x(i))`). Skriv om $f_1, f_2, g_1$ och $g_2$ så att de är vektoriserade. Testa på vektorn `x1` igen och på $x_2 = (-2, -1, 0, 1, 2, 3)$. Hur skapar man den senare på enklast sätt?

## 3.2. **Några rituppgifter.**

3.2.1. *Ett tredjegradspolynom med olika upplösning.* Gör följande för $N = 10, 30, 50, 100$ för att se hur kvalitén på figuren förbättras med ökat $N$. Börja med att skapa en m-fil poly3deg.m genom att skriva `edit poly3deg.m` vid matlabprompten. Kopiera sedan in nedandstående och förfärdiga det. *Kommentera* koden och skriv efter varje rad vad som händer. (Kommentera gör du genom att skriva ett procenttecken följt av din kommentar. Allt som står efter ett procenttecken i Matlab hoppas över av programmet: det är bara synligt för det mänskliga ögat och är ett sätt för programmeraren att själv komma ihåg hur hon tänkte när hon skrev programmet och att kommunicera det till andra användare och utvecklare. Det är ofta bra när man ska leta fel i kod att det är välkommenterat.)

```
f=@(t)(t.^3-2t.^2+t-1);
a=?
b=?
N=10;
x=linspace(a,b,N);
y1=f(x);
plot(x,y1,'*-')
hold on
N=30;
y2=f(x);
plot(x,y2,'kd-')

        ⋮

plot(x,y4,'g:o')
legend('N=10','N=30','N=50','N=100')
title('Grafen till f(x)=x.^3-2x.^2+x-1')
xlabel('x')
ylabel('f(x)')
```

Hitta sedan nollställena till polynomet med funktionen `roots`, se `help roots` och plotta dem som cirklar i figuren genom att skriva

```
hej=roots(?) %Hur ska man använda roots?
stlk=size(hej) %Vad gör size?
y=zeros(stlk) %Vad gör zeros?
hold on       %hold on?
plot(hej,y,'o')
```

3.2.2. *Sinusfunktioner och samplingssatsen.* Om man har för mycket punkter i plotten så blir det långsamt och tar massa minne utan att man vinner så mycket extra insikter jämfört om man hade haft, säg hälften så många. Har man för lite så kan det dels bli fult, men man kan också missa viktig information.

Öppna en ny m-fil och plotta funktionen $\sin(\omega t)$ med `linspace(-pi,pi,N)` med $N = 10$, 100, 500, 1000 och 10000 punkter för $\omega = 100$ och försök förstå vad som händer. Uppenbarligen blir det alldeles galet när man har för få punkter, trots att det ser fint ut! Sensmoralen är att man bör försöka skapa sig en förståelse av funktionen man ska rita. *Datorn kräver sin matematiker.*

Det här är för övrigt ett exempel på Nyquists samplingssats som säger att man måste mäta en signal minst två gånger per period för att det inte ska bli fel. Man kan se plottningen som att man mäter funktionsvärdet i de punkter som finns i vektorn $x$ i det här fallet och sedan försöker återskapa funktionen med linjär *interpolation* mellan mätpunkterna.

- Vad är periodlängden för funktionen $\sin(100t)$?
- Om vi skulle skapat $x$ med syntaxen $-p : dx : pi$, hur stort skulle $dx$ kunna vara för att uppfylla villkoret i Nyquist samplingssats?

Nyquists samplinssats är viktig för en elektroingenjör att känna till och den kommer dyka upp i många kurser i senare årskurser.

3.2.3. *Olika funktioner.* Plotta följande funktioner på intervallet $[-\pi/2, \pi/2]$. Använd olika linjefärger och ”legend” så man ser vilken graf som hör till vilken figur.

$$
\begin{aligned}
y &= x^2 \\
y &= \tan(x) \\
y &= \cot(x) \\
y &= \ln(|x|) \\
y &= \exp(x)
\end{aligned}
$$

För flera matematiska funktioner, skriv

```
help elfun.
```

3.2.4. *Komplexa tal ger problem.* Rita grafen till funktionen $y = \sqrt{x}$ med definitionsområdet $-3 \leq x \leq 3$. Läs varningen som skrivs ut i kommandofönstret. Skriv sedan

```
figure(2)
plot(y,'*') %Eller möjligen plot(y,'*-r')
```

i din skriptfil. Vad visar denna plot? Här finns tydligen anledning att vara vaksam! Zooma slutligen in origo i denna plot genom att skriva

```
axis([-.5 .5 -.1 .5]).
```

Vad gör kommandot ”axis”?

3.2.5. *Translation och skalning 1.* Antag att vi har en funktion $f(x)$. Vi har sett att grafen av $f(x-b)$ ligger $b$ steg till höger om grafen till $f(x)$. Dessutom har vi sett att grafen av $f(ax)$, $a > 0$ är en version av grafen som är ihoptryckt så att ”bredden” i någon mening är $1/a$ av den ursprungliga. För övrigt så gäller att $Af(x)$, $A > 0$ gör grafen $A$ gånger så hög. Ett särkilt intressant fall (i synnerhet för en elektroingenjör) är funktionen

$$y = A\sin(kx - a) \quad (A > 0,\ k > 0)$$

Vad kallas $A$, $a$ och $k$? Se Adams sid 207. Variera en parameter i taget och plotta de olika grafer du får i tre olika fönster i samma figur med hjälp av ”subplot”. Se till att parametern antar ett värde som är mindre än 1, ett som är större och att den också antar värdet 1. Studera noggrant de fenomen som beskrivs ovan. Gör titlar etc. som i uppgiften ovan.
Tips: Definiera en implicit funktion som tar fyra inargument: $k, a, A$ och $x$. enligt

```
funk=@(x,a,A,k)(A*sin(k*x-a))
```

3.2.6. *Translation och skalning 2.* Gör om uppgift 3.2.5 hela för funktionen $g(x) = \exp(-x^2)$. Den här funktionen spelar en alldeles särskild roll i sannolikhetsteorin med det speciella valet $A = k/\sqrt{\pi}$. Här kallas talet $a$ för medelvärdet och $\frac{1}{\sqrt{2}k}$ är standardavvikelsen. Grafen är den berömda normalfördelningskurvan. Ni borde alltså kunna återskapa bilden på

`http://sv.wikipedia.org/wiki/Fil:Normal_distribution_pdf.png.`

Läs alltid Wikipedia-artiklar med kritiska ögon, men läs dem gärna. Det finns förvånansvärt mycket om matematik och teknik!

3.2.7. *Rationella funktioner 1.* Plotta grafen till funktionen

$$y = \frac{1}{x}$$

i ett område runt origo. Plotta också den lodräta asymptoten med en annan linjetyp.

3.2.8. *Rationella funktioner 2.* På räkneövningarna har vi ritat rationella funktioner. Nu ska vi göra det med hjälp av Matlab också. Betrakta funktionen

$$f(x) = \frac{x^3 - 4x}{x^2 - 1}.$$

Man kan visa att den har singulariteter i $x = -1$ och $x = 1$ samt en inflektionspunkt i $x = 0$. Använd på ett smart sätt `roots` för att hitta dessa singulariteter. Hitta också nollställen med hjälp av samma funktion! Vi vet också att den beter sig som $y = x$ då $x \to \infty$. Rita funktionen genom att t.ex. plotta den i tre steg. Först på intervallet $[-K, -1)$, sedan på $(-1, 1)$ och slutligen på $(1, K]$ där $K$ är något lämpligt tal som är större än 2. Markera sedan inflektionspunkten med en cirkel. Slutligen plottar du de lodräta asymptoterna och grafen av $y = x$ med en annan linjetyp än funktionsgrafen. Notera att räta linjer alltid kan plottass med bara två punkter! Sätt ut namn på linjerna med `legend`, titel på figuren med `title` samt namn på axlarna med `xlabel` och `ylabel`.

Ett snyggare sätt att få till fina grafer av funktioner vars singulariteter man har konstaterat ligger i intervallet $(\hat{x} - \epsilon, \hat{x} + \epsilon)$ för något litet positivt $\epsilon$ fås av följande exempel.

```
funk=@(hej)(?) % En anonym funktion som definerar en funktion med
               % singularitet.
xhat=? %Den punkt där vi tror singulariteten finns
epsilon=? % Ett avstånd till xhat så att vi är säkra på att
          % singulariteten inte ligger utanför intevallet
          % (xhat-epsilon,xhat+epsilon)
N=?% Antal punkter i varje delintervall
Left=? %Intervallets vänstergräns
Right=? %Intevallets högergräns
x_left=linspace(Left,xhat-epsilon,N);  % Gör punkter till vänster om
                                       % singulariteten.
x_right=linspace(xhat+epsilon,Right,N);% Gör punkter till höger om
                                       % singulariteten.
x=[x_left NaN x_right]; % Sätter ihop alla dessa punkter med
                        % ''Not a Number'' där singulariteten finns.
y=f(x); %Beräknar funktionsvärdena av punkterna i vektorn x
plot(x,y) % Nu riter Matlab grafen men ritar inte punkten med NaN.
          % Där får grafen ett hål och vi slipper felaktiga,
          % fula linjer.
```

3.3. **Optimering.**

3.3.1. *Anrop av färdiga funktioner, nollställen till generella funktioner.* Funktionen ”roots” ovan hittar bara nollställen till polynom. För att lösa $f(x) = 0$ för en mer generell funktion $f$ så används ”fzero”. Läs igenom hjälpfilen och använd funktionen på polynomen i föregående uppgift. Jämför lösningarna. Det finns minst en viktig skillnad! Försök också med (den matematiska) funktionen $g(x) = (x^2 - 1)\exp(x)$. Var har $g$ sina nollställen?

3.3.2. *Anrop av färdiga funktioner, min- och maxvärden.* I matlab finns två standardlösare för att minimera funktioner, ”fminsearch” och ”fminbnd”. ( Dessutom finns bättre lösare för olika speciella problem i ”Optimization Toolbox”.) Vad är skillnaden mellan ovan nämda funktioner? Testa båda på funktionen $g(x)$ i uppgiften ovan. Funktionen $\tilde{g} = g(x) * (x^2 - 4)$ har flera lokala minima. Plotta funktionen och välj gränser respektive startvärde så att du hittar alla! Det krävs alltså en sökning per lokalt minimum.

Hur gör man för att hitta ett maximum med hjälp av dessa Matlab-funktioner?

3.3.3. *Mera optimering.* Betrakta funktionen

$$f(x) = -\exp(0,01x)(1 + \sin(x))$$

. Försök hitta det globala maximat på $\mathbb{R}^+$!

3.3.4. *Lurad av grafen?* Plotta funktionen $f(x) = (x\ln(x) + 3\sin(x) + 7x)/\sqrt{x}$ först på intervallet $[0, 1]$ och sedan på $[0, 5 \cdot 10^{-5}]$. I den första plotten missar man tydligen ett globalt minimum. Försök hitta det med optimeringsfunktionerna från föregående övning. Svårt, eller hur?! Testa med $g(x) = 1000f(x/1000)$. Kan du med hjälp av lösningen till det problemet räkna ut ungefär vad lösningen är till det första?

3.4. **Extrauppgifter.** Gör det du inte hann med i den förra laborationen!