

## DATORÖVNING 3 — FUNKTIONSANROP OCH OPTIMERING

### INSTRUKTIONER SYFTE OCH INNEHÅLL

Skapa en ny filkatalog (“directory”) `Lab3` för denna övning. Gör alltid uppgifterna i script-filer eller funktionsfiler om inget annat står. Lista gärna alla kommandon du använder under den här laborationen. Du kommer att ha nytta av samtliga i framtiden.

Laborationen syftar till att skapa ytterligare vana vid grafitning samt att förstå funktionsfilers uppbyggnad och användande av olika antal in- och utargument samt olika typer av argument. Du skall också bekanta dig med Matlabs färdiga funktioner för att hitta nollställen samt minima och maxima till funktioner. I synnerhet presenteras funktionsfilerna `roots.m`, `fzero.m`, `fminsearch.m`, `fminbnd.m`, `min.m` och `max.m`. Från förut dyker dessutom några elementära matematiska funktioner upp (t.ex `sin`, `exp`) samt `plot`, `legend.m`, `title.m`, `linspace.m`. Mera perifera saker är `nargin` och `num2str.m`. Vi presenterar också en hemmabygd funktion `sinfaf.m` och Matlabs `if-elseif-if`-konstruktion. Därvidlag dyker också olika jämförelseoperatorer upp.

### 1. UPPGIFTER

1.1. **Matematiska funktionsfiler.** Skapa en *funktionsfil* som tar ett inargument  $x$  och returnerar ett värde  $y = (x^2 - 1)e^x$ . Se till så att Matlab inte skriver ut uträkningen i kommandofönstret om man inte vill det. Använd semikolon. Plotta funktionsgrafan på lämpligt intervall. Vad är skillnaden mellan en (icke-anonym) funktion och en anonym sådan?

1.2. **Sinus med olika frekvens, amplitud och fas.** Det finns två syften med den här uppgiften: Att du ska förstå antalet arguments koppling till hur programmet är skrivet och att få intuition för sinusfunktioner och deras frekvens, amplitud och fas.

Ladda ned matlabfunktionen `sinfaf.m` från kurshemsidan eller Fredriks hemsida och spara den i katalogen `Lab3`. Skriv `help sinfaf` i kommandofönstret. Vad gör funktionen? Titta på hela filen med hjälp av `edit sinfaf`! Första ”aktiva” raden lyder

```
function [y,dy]=sinfaf(x,vfreq,amp,phase)
```

Hur många variabler kan man som mest få ut av funktionen? Hur många inargument kan man som mest skicka med? Försök förstå vad hela funktionsfilen gör för något! Testa följande kommandosekvenser i en script-fil. Separera sekvenserna med `%%` så att de bildar celler. Då kan du köra varje cell för sig genom att välja **cell** och sedan **evaluate current cell** i panelen i editorn. Tänk efter vad som kommer hända innan du exekverar (kör) programmet!

Fungerar ett in- och utargument?

```
x=linspace(-pi,pi);
freq=5; %Egentligen vinkelfrekvensen som är 2*pi*frekvensen
amp=2;
phase=pi/4;
freq2=10;
ut1=sinfaf(x);
plot(x,ut1)
hold on
```

Två inargument?

```
ut2=sinfaf(x,freq);
plot(x,ut2,'r')
```

Tre inargument?

```
ut3=sinfaf(x,freq,amp);
plot(x,ut3,'g')
```

Fyra inargument? Vad händer i den andra plotten?

```
ut4=sinfaf(x,freq,amp,phase);
plot(x,ut4,'k')
plot(ut4,x,'k')
```

Fem inargument?

```
ut5=sinfaf(x,freq,amp,phase,freq2);
```

Två utargument? Notera hur man kan skapa strängar av dubblarna och bygga ihop strängar t.ex. för att använda till `title`, `legend` etc.

```
figure(2)
[ut6,ut7]=sinfaf(x,freq,amp,phase);
plot(x,ut6,x,ut7)
str1=[num2str(amp) 'sin(' num2str(freq) '(x-' num2str(phase) ')')]'%!?!?
%help num2str
str2=[num2str(freq*amp) 'cos(' num2str(freq) '(x-' num2str(phase) ')')]'
legend(str1,str2)
```

Tre utargument?

```
[ut8,ut9,ut10]=sinfaf(x,freq2,amp,phase);
```

Jämför med en annan vinkelfrekvens.

```
[ut8,ut9]=sinfaf(x,freq2,amp,phase);
figure(3)
plot(x,ut8,'r:',x,ut9,'k-o')
str3=[num2str(amp) 'sin(' num2str(freq2) '(x-' num2str(phase) ')')]'
str4=[num2str(freq2*amp) 'cos(' num2str(freq2) '(x-' num2str(phase) ')')]'
legend(str3,str4)
```

Vi kastar om inargumenten. Det är väldigt viktigt att förstå vad som sker här!

```
figure(4)
[ut10,ut11]=sinfaf(x,amp,phase,freq2);
plot(x,ut10,x,ut11)
```

Ändra gärna lite på fas och amplitud också! Var säker på att du kan förklarar hur grafen förändras när de olika värdena minskar/ökar. Det är något särskilt när fasföskjutningen är  $\pi$  respektive  $\pi/2$ . Vad?

Läs igenom filen och se hur `if-elseif-else`-satser använd för att hantera olika antal argument!

**1.3. Derivata.** Hitintills har vi sett matlabfunktioner som har tagit inargument som är tal eller listor med tal (vektorer). I den här uppgiften ska du skriva en funktion som tar ett funktionshandtag (det vill säga en adress till en anonym- eller ickeanonym funktion) och använder det för att räkna ut derivatan av denna funktion. Du ska alltså skriva en funktionsfil som deriverar en given funktion med hjälp av den symmetriska differenskvoten

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Laborationshandledaren förklarar varför den ekvationen är bättre än den som man får direkt ur derivatans definition! Kalla funktionen `derivera.m` och låt den ta tre inargument, ett funktionshandtag  $f$ , en steglängd  $h$  samt punkten där derivatan ska beräknas  $x$ , och returnera derivatans värde  $df$ . Se till att funktionen är vektoriserad. Plotta resultatet för två olika frekvenser på en sinusfunktion! Överkurs är att sätta ett defaultvärde på  $h$  med hjälp av `nargin` och en `if-sats`. Ett bra värde är  $h = 10^{-6}$ . Titta hur det fungerar i `sinfaf.m`!

DEN HÄR FUNKTIONEN KOMMER DU BEHÖVA NÄSTA LABORATION! Om ni jobbar tillsammans, se till så att alla har den på sin dator ifall någon blir sjuk etc.



```
y=0:dx:5;  
z=f(y);  
[m,ind]=max(z);  
x=linspace(0,5,1e5);  
plot(x,f(x),y(ind),m,'*r')
```

Uppenbarligen är det hittade maximat inte funktionens maximum! Testa

```
hold on  
plot(y,z,'*k')
```

1.7. **Mera optimering!** Gör optimeringsuppgifterna från förra laborationen!

1.8. **Andraderivatan.** Implementera en funktion som beräknar andraderivatan av en funktion med hjälp av funktionen `derivera.m`.

1.9. **Newtons metod.** Kolla upp Newtons metod i Adams *Calculus* och skriv en funktion som tar ett steg med denna metod. Det är första steget mot att skriva egna funktioner som hittar nollställen och minima.