Lab 2, Funktioner, funktionsfiler och grafer.

Laborationen innehåller 8 deluppgifter.
Uppg. 1-3: behandlar Matlabs grundläggande operationer
Uppg. 4-5: behandlar kurvritning
Uppg. 6-8: behandlar funktionsfiler och något om programering.

1 Grundläggande operationer

1.1 Aritmetiska operationer, (PS 2.5).

De vanliga aritmetiska operationerna mellan tal ser ut så här :

- \+ addition
- − subtraktion
- * multiplikation
- / division
- ^ exponentiering (OBS! För att få denna symbol skriver man ^ följt av mellanslag!)

Talet π skrivs `pi` medan talet e skrivs `exp(1)`
Några exempel:
Skriver du
`>> 100*pi` (utan ; före avslutande RETURN)
blir svaret
```
ans =
314.1593
```
(`ans` står för det senaste svaret).
Som påpekats ovan skall man ”alltid” ge namn till beräkningarna. I ovanstående exempel skriver man då
`>> a=100*pi`
och får svaret
```
a =
314.1593
```

MATLAB tillämpar den vanliga prioriteringsordningen mellan de aritmetiska operationerna :
`>> 8^1/3`
ger svaret `2.6667`, medan
`>> 8^(1/3)`
ger svaret 2.
Man får allmän hjälp om dessa begrepp i `helpwin` genom att gå till biblioteket `matlab/ops` och i `helpdesk` genom att välja `functions by subject` och sedan `Operators and Special Characters`.

1.2 Elementära funktioner, (PS 2.5).

MATLAB har alla de vanliga elementära grundfunktionerna, alltså exponential- och logaritmfunktionerna, de trigonometriska funktionerna och deras inverser, absolutbelopp, kvadratrot, och flera andra. Här följer en lista på några av MATLABs funktioner:
`exp, log` (= ln), `log10` (= 10-logaritmen), `sin, cos, tan, atan` (= arctan),
`asin, abs, sqrt sinh, cosh, tanh`
Observera att man alltid måste ha () runt variabeln som i `sin(pi/3)`.
Det finns ytterligare funktioner t.ex.
`sign , round , floor , ceil`
Man får en fullständig lista i `helpwin` under `matlab/elfun`.

Exempel: Vi beräknar $\ln(\sqrt{e})$:
```
>>y = log(sqrt(exp(1)))
y = 0.5000
```
Vi löser ekvationen $e \tan x = 5, \quad -\pi/2 < x < \pi/2$:
```
>> x = atan(5/exp(1))
x =1.0728
```
Notera att man inte kan skriva `e^x` för exponentialfunktionen.
(Läs mer i P-E,S 2.5)

1.3 Formatering av utskrift, (PS 2.6).

Man kan dirigera antal decimaler som skrivs ut och formen på utskriften med hjälp av kommandot `format`. De vanligaste varianterna är

format short	ger fem signifikanta siffror
format long	ger femton signifikanta siffror
format short e	ger fem signifikanta siffror i flyttalsnotering
format long e	ger femton signifikanta siffror i flyttalsnotering

En fullständig lista av de olika utskriftsformaten finner man under `format` i `matlab/general`.
Prova med att skriva ut 10π i de olika utskriftsformaten, t.ex.
```
>> format long
>> 10*pi
ans =  31.41592653589793
```

OBSERVERA :
I vissa av följande uppgifter förekommer variablerna $p_1, \ldots, p_{10}$. De avser siffrorna i ditt personnummer $p_1p_2p_3p_4p_5p_6 - p_7p_8p_9p_{10}$ (välj ett av era personnummer om ni är två som gör övningen tillsammans).

Uppgift 1: Låt c vara $p_7p_8p_9p_{10}/1000$ och skriv in detta i Matlabs kommandofönster:
```
>>c=ditt värde;
```
Beräkna därefter med hjälp av MATLAB

a) roten till ekvationen $10^x = 21$. (Se ovan hur ekv $e \tan x = 5, \quad -\pi/2 < x < \pi/2$ kunde lösas.) Ge roten namnet rot1a.

b) den positiva roten till ekvationen $ln(1 + x^2) = 1/c$. Ge roten namnet rot1b.

1.4 Operationer med radmatriser, (PS 3.1-7).

För att utnyttja MATLAB effektivt skall nästan alla variabler man använder vara radmatriser eller större matriser. Detta innebär en viss komplikation. Multiplikationen `x*y` betyder matrismultiplikation. Om `x` och `y` är enstaka tal så är det den vanliga produkten. Är `x` och `y` matriser så finns inte produkten såvida inte typerna stämmer överens.
Vi kan som nämnts ovan föreställa oss en matris lagrad som en lång lista av tal tillsammans med uppgift om matrisens typ. Ofta vill vi beräkna elementvisa (**punkt**-visa) produkter av tal i sådana listor. Den produkten skrivs `.*` alltså med en **punkt** framför `*`-tecknet. Med `x = [1,2,3]` är `x.*x= [1,4,9]`. På samma sätt skrivs division `x./y` och potenser `x.^y`.
Man kan även använda de elementära funktionerna på matriser. Allmän hjälp om detta område får du i `helpwin` under `matlab/elmat` och `matlab/ops`.

Exempel (med utskrift i `format short`):

`>>x=[1 2 3]; y=[4 5 6];`	ger
`x+y`	`5 7 9`
`x.*y`	`4 10 18`
`x./y`	`0.2500 0.4000 0.5000`
`x.^y`	`1 32 729`
`exp(x)`	`2.7183 7.3891 20.0855`

Viktiga specialkommandon är `ones` och `zeros`. De användes för att generera matriser bestående av enbart ettor respektive nollor. T.ex. ger kommandot `ones(1,3)` eller `ones(size(x))` svaret `1 1 1` om `x` är en radmatris av längden 3.

Exempel:

```
>>x=[1 2 3 4]; z=ones(size(x))./x
```

ger svaret `>>z = 1.0000 0.5000 0.3333 0.2500`

`>>x+ones(size(x))` ger svaret `2 3 4 5`

`>>2*ones(size(x))` ger svaret `2 2 2 2`

Det näst sista svaret hade man också kunnat få genom att skriva `x+1`.
`z` kan man erhålla med `z = 1./x`.

Uppgift 2: Skriv in radmatriserna $x = [p_1\ p_2\ p_3]$ och $y = [p_4\ p_5\ p_6]$.
Beräkna `x+y, x-y, x.*y, x.^y, x./y, x.\y, x.*sin(y).`
Vad svarar Matlab på $x * y$.

1.5 Generering av aritmetiska följder, (PS 4.3).

Om a, h och b är givna tal kan man bilda radmatrisen `x=[a,a+h,a+2h,... ,b]` med hjälp av kommandot `x=a:h:b`.

`>>x=-5:2:5` ger

```
x =
    -5 -3 -1 1 3  5
```

Analogt ger kommandot

```
>>x=-pi:0.1:pi;
```

radmatrisen `x=[-3.1416 -3.0416 -2.9416 ... 2.9584 3.0584]`, vilket är en matris av längden 63. Kolla genom att mata in `x` enligt ovan och sedan skriva

```
>>length(x)
```

Låt också datorn skriva matrisen `x` på skärmen genom att skriva
`>>x` (utan semi-kolon)
Vill man ha steglängden `h = 1` räcker det att skriva `>>x=a:b` t.ex.

```
>>x=0:10
x = 0 1 2 3 4 5 6 7 8 9 10
```

Om `b < a` så kan man ha `h < 0` .
Ytterligare information får du under `matlab/ops`.
Läs den informationen, kolon-operatorn är en mycket viktig ingrediens i Matlab.

Uppgift 3: Låt n vara ett givet positivt heltal. Generera radmatriserna `m=[-1,-2,...,-n]` och
`x=[2^(-1),2^(-2), ... , 2^(-n)]`. Använd sedan kommandot `sum(x)` för att beräkna den geometriska summan

$$s = 2^{-1} + 2^{-2} + \cdots + 2^{-n}$$

Vilket är det första n-värde för vilket $s > 0.999999$?

2 Kurvritning

2.1 Allmänt om plotkommandot, (PS 13.1-2).

Om `x = [x(1), ... ,x(n)]` och `y = [y(1), ... ,y(n)]` är två radmatriser av samma längd så kommer ritkommandot `plot(x,y)` att rita en kurva som sammanbinder de `n` stycken punkterna
`(x(1),y(1)), ..., (x(n),y(n))`. Om `x` eller `y` innehåller flera rader så kommer flera grafer att ritas i samma figur.
Om man inte ger ett särskilt kommando kommer MATLAB att välja koordinataxlarna så att samtliga punkter syns. (Detta kallas auto-scaling.) Man kan dirigera på vilket sätt kurvan ritas genom att ge order om särskild linjetyp. Man kan också rita ut punkterna utan att sammanbinda dem genom att bestämma vilken punkttyp som skall användas.
Exempel:

```
>>x=-3:0.5:3;y=sin(x);
>>plot(x,y)
>>plot(x,y,':')          (Prickad sammanbunden kurva.)
>>plot(x,y,'.')          (Punkterna utritade som små prickar.)
>>plot(x,y,'o')          (Punkterna utritade som små cirklar.)
```

Allmän hjälp för tvådimensionell grafik ges i `matlab/graph2d`.

2.2 Funktionskurvor

Av exemplen ovan framgår det redan hur man kan rita funktionskurvor. Säg att vi vill rita funktionen $f(x)$, på intervallet $a \leq x \leq b$. Man börjar då med att välja en lämplig steglängd h och bildar sedan radmatrisen `x=a:h:b`. Därefter låter man MATLAB beräkna funktionsvärden samt ger dessa ett namn. . Man får funktionskurvan uppritad med kommandot `plot(x,funktionsnamn)`.
Exempel:

```
>> x=-2*pi:0.1:2*pi;
>> y=sin(3*x)+cos(5*x);plot(x,y)
```

Här ritas funktionen $f(x) = \sin 3x + \cos 5x$ på intervallet $[-2\pi, 2\pi]$ På samma sätt ritar man funktionen $g(x) = x \sin x^2$ på samma intervall med hjälp av kommandot

```
>>g=x.*sin(x.*x);plot(x,g)
```

Uppgift 4: Rita ovanstående figurer.
Vid kurvritning bör man tänka på att det blir bättre graf om man har fler punkter, men bara upp till en viss gräns. Radmatriserna bör inte ha fler än ca 400 element. Däröver blir det bara slöseri med beräkningstid och minnesutrymme. I synnerhet om bilden skall skrivas ut på papper så är detta mycket viktigt.

2.3 Flera kurvor i samma figur, (PS 13.4).

Antag att vi vill rita funktionerna f och g ovan i samma figur. Vi kan då ge kommandot
`plot(x,f,'-',x,g,'-.')` eller bara `plot(x,f,x,g)`
Det finns också en annan möjlighet. Antag att vi först ritar funktionen f med hjälp av kommandot

```
>>plot(x,f)
```

Man kan sedan ge kommandot

```
>>hold on
```

Då kommer de gamla kurvorna att behållas när nya kurvor ritas. Till exempel

```
>>plot(x,g)
```

När man inte längre vill behålla gamla kurvor ger man kommandot

```
>>hold off
```
Kommandot `hold` ensamt innebär att man skiftar från "hold on-läge" till "hold off-läge", (om man tidigare gett kommandot `hold on`) eller från "hold off-läge" till "hold-on-läge".

2.4 Dimensionering av koordinataxlarna, (PS 13.4).

Normalt väljer MATLAB ett koordinatsystem så att alla punkter som skall ritas syns på skärmen. Man kan styra valet av koordinataxlar med hjälp av kommandot `axis`. För att ta reda på hur `axis` fungerar kan du läsa hjälptexten i `matlab/graph2d`.

Uppgift 5: Ett bekvämt sätt att ge ett plotkommando där det är lätt att ändra är att på en enda rad skriva enligt följande exempel.
```
>> a=0; b=1; h=(b-a)/400; x = a:h:b; y = sin(1./x); plot(x,y)
```
Med vänster och höger piltangenterna kan du flytta markören till den plats du vill ändra. Backspace och del raderar, pröva vad de tar bort.
Rita en figur i vilken de tre kurvorna $y = sin(x)/x$, $y = cos(x)$ och $y = 1$ förekommer samtidigt. Den sista får du med `y=ones(size(x))`.
Zooma in för att titta på gränsvärdet $\lim_{x\to 0} \dfrac{\sin(x)}{x}$.

3 Funktionsfiler.

3.1 Hur man skriver en funktionsfil, (PS 2.8).

Programering i matlab sker med hjälp av m-filer. dessa filer skall ges namn som slutar med .m (t.ex. funk.m, kvadrat.m etc.) Det finns två sorters m-filer: *funktionsfiler* och *kommandofiler (alt. scriptfiler*. Vi kommer i denna lab att koncentrera oss på funktionsfiler.
Du skall nu skriva din första m-fil. Starta MATLABs texteditor. Skriv sedan av exakt enligt nedanstående och spara det du skrivit med namnet `fun.m` i `matlab/lab2`.
Första två raderna skall vara så här:
```
function y=fun(x)
y = x.*x -1;
```

Nu kan du kontrollera att filen finns i det aktuella biblioteket med kommandot `what`. Testa sedan filen med
```
>> fun(1)
```
och
```
>> fun([1,2,3])
```
Fungerade detta så kan du rita grafen med
```
>> fplot('fun',[-1,1])
```
Dessa tre steg är ett bra sätt att kontrollera en sådan här funktonsfil fungerar (`fplot = funktionsplot`).
Vi väntar till lite senare i kursen med att förklara hur funktionsfiler fungerar. En viktig detalj är emellertid att filnamnet *fun*.m och texten i första raden y = *fun*(x) skall stämma överens.

Uppgift 6: Gör en funktionsfil *fun2.m* som ger funktionen $y = sin(x)/x$. Spara den som `fun2.m`. Testa den som ovan.

Uppgift 7: Gör en funktionsfil *summa.m* som beräknar summan $1^2 + 2^2 + 3^2 \cdots + n^2$. (Här är n variabeln.) Testa genom `>> summa(3)` ,
```
>> summa(4)
```

3.2 Loopar, (PS 12.4).

Exempel: Vi löser ekvationen $sin\sqrt{x} = x$ approximativt genom att göra en funktionsfil *roten.m* som itererar ett givet antal gånger, n. Här behövs en loop:

```
function x=roten(n)
x=1/2;
for i=1:n
x=sin(x^0.5);
end
```

I nästa uppgift behöver du/ni två personliga heltalsparametrar. Låt därför a och b vara sista siffrorna som är skilda från noll i era respektive personnummer. Om du labbar ensam så låter du a och b var sista respektive näst sista siffran skild från noll.

Uppgift 8: Skriv en funktionsfil *macl1(x,n)* som ger Maclaurinpolynomet av grad $2n$ till funktionen $f(x) = \frac{\sin(ax)}{x}$.
Tänk på att funktionen skall klara av x som vektor. (n är däremot alltid en skalär.)
Tips:
Utgå från polynomet av grad 0 och bygg succesivt upp polynomet i en loop genom att varje varv lägga till nästa term i polynomet. Obsevera att $n!$ heter *factorial(n)*.
Plotta *macl1(x,n)* tillsammans med $y = f(x)$ för $n = 1$, $n = b + 5$ och $n = b + 20$. Det skall alltså vara tre separata plottar med både polynomet och $y = f(x)$ på var och en.
Tips:
Välj axlar så att det ser bra ut. T.ex. med x från $-(10 + n)/a$ till $(10 + n)/a$ och y från $-a$ till $a + 1$.