Intro Math K + Kf + Bt

Lecture 2:

Matlab

Vad är Matlab

- Matrix Laboratory (MathWorks, Comsol)
- avancerad grafritande räknare
- programmerbar
- “typenkelt” och interpreterande
- finns på biblioteket för din hemdator!
- onlinehjälp: help, help *ämne*, helpwin, helpdesk
- toolboxar
- andra: mathematica, maple, derive, excel, mathcad,..

tal, format, operatorer

- heltal: 2, -7
- rationella tal: 0.4286 (=3/7)
- finns fler decimaler: format long => 0.42857142857143
- 1.2345e+9 (=123450000)
- $+$, $-$, $*$ (mult), $/$, ^ (potens)
- $\exp(-1)$, $\sin(pi/3)$
- help ops, elfun

sant, falskt, villkor

- (2<5)=1, (2>5)=0
- 0 = falskt, allt annat sant
- exempel: `if (2>5) beep; end` (ger *ej* pip)
- exempel: `if 0.1 beep; end` (ger pip)
-
```
if villkor
  ..;
elseif nytt villkor
  ..;
end
```
- Om *det blir sol* så ..; annars ..; end
- help lang

repetitionssatser, loopar

- while *villkor* ..; end
- typexempel:

```
wv = 1;
while wv
  ..
  if villkor
   wv = 0;
  end
end
```


repetitionssatser, loopar

- alternativ:

```
while 1
  ..
  if villkor
    break
  end
end
```


repetitionssatser, loopar

- for *index=index_range* ..; end
- index=startvärde:steg:slutvärde (definierar indexlista!)
- exempel:

```
for i=9:-2:3
  disp(i);
end

s=0; for i=1:100
  s=s+i;
end
```


funktioner

- eval
- inline
- funktionsfil
- exempel:

```
f='3*x+4';
x=2;
eval(f)
```
(ger värdet 12)

inlinefunktioner

- inlineing: (temporära, help inline)

```
f=inline('3*x+4');
f(2) (ger värdet 12)

f=inline('x^2*h','h','x');
f(2,3) (ger värdet 18)
```


funktionsfiler

- skrivs i textfiler på disk (=bestående)
- function utvariabler=funktionsnamn(invariabler)
- skriver i fil MyFunction.m

```
function y=MyFunction(x)
y=3*x+4;
```

- >> MyFunction(2) (ger värdet 12)
- >> feval('MyFunction',2) (ger värdet 12)

plottning

- Exempel:

```
x=0:.1:1; (ger x-koordinater mellan 0 och 1)
y=3*x+4;
plot(x,y) (plottar kurvan y=3*x+4)
```

- title('..')
- xlabel('..'), ylabel('..')
- grid on/off
- plot(x,y,'r')
- hold on/off

problemlösning

subsection*Skinnarmos problem

- Hur många dar tar det till nordpolen?
- och hur mycket proviant behövs?
- model:
- s=v*t (sträcka = hastighet * tid)
- m (proviant kvar), dm (proviantbehov per dag)
- v=1/(1+m) (hastighet beroende av resterande last)

problemlösning

- resulterande kod (i scriptfil):

```
m=100; (utgångsproviant)
dm=1; (proviantbehov per tidsenhet/dag)
dt=1; (tidssteg, dagar)
s=0; (utgångssträcka)
while m>0
  v=1/(1+m); (aktuell hastighet)
  s=s+v*dt; (tillkommer sträcka)
  m=m-dm*dt; (provianten minskar)
  t=t+dt; (tiden går)
end
```


förenklingar

- resulterande kod:

```
m=100:-1:1;
v=1./(1+m);
s=sum(v);
```

- Resulterande sträcka:

$$s = \frac{1}{101} + \frac{1}{100} + \frac{1}{99} + .. + \frac{1}{2}$$

sökvägar

- workspace variabel -> builtin funktion -> funktionsfil
- path (listar aktuella sökvägarna)
- addpath 'C:/..' -end alt. path(path, 'C:/..') (lägger till sökbibliotek)
- pwd (skriver ut aktuellt bibliotek/mapp)
- ls alt. dir (listar filer i aktuellt bibliotek)
- cd .., cd *biblioteksnamn*
- ! *systemkommando* (skickar *systemkommando* till systemet)

felsökning

- utveckla koden i små steg
- skriv ut
- se: help debug
- testfall med känd lösning
- samla erfarenhet