

Funktioner i MATLAB

Uge 5

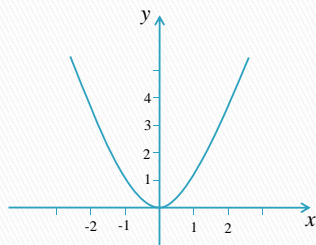
JRF 2009

Træt af at gentage kode?

- ▶ Samme beregning – forskellige input
 - Eks.: Φ , Ψ , A for forskellig v i Rapport 1.
- ▶ Samme simulering – forskellige tidsskridt
 - Eks.: Skridtlængde Δt og antal skridt N i Rapport 1.
- ▶ Samme model – forskellige startværdier
 - Eks.: u -vektor for Tumor 1 og Tumor 2 i Rapport 1.
- ▶ Samme formel – forskellige konstanter
 - Eks.: Forskellige cancer-typer i Rapport 1.
 - Eks.: A -matrix for forskellige retningsfelter.
 - Eks.: Leslie-matrix for forskellige befolkninger.

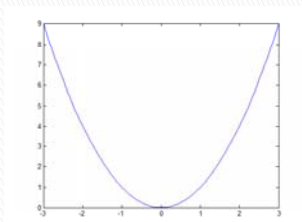
Funktioner er løsningen

▶ $y = f(x) = x^2$



Matematik

▶ MATLAB file: f.m
`function [y] = f(x)`
`y = x.^2;`
`end`



MATLAB

Øvelse: input – output

▶ $f(x) = x^2$

▶ MATLAB file: xsquare.m
`function [y] = xsquare(x)`
`y = x.^2;`
`end`

▶ $f(v) = \|v\|$, $v \in \mathbb{R}^2$

▶ MATLAB file: veclength.m
`function [len] = veclength(v)`
`len = sqrt(v(1)^2 + v(2)^2);`
`end`

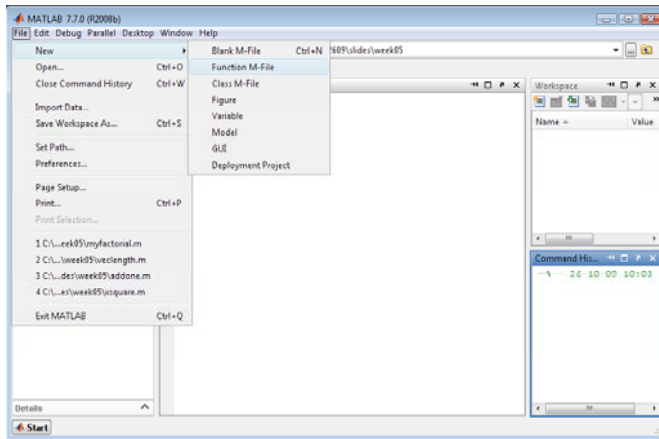
▶ $f(n) = n!$, $n \in \mathbb{N}$

▶ MATLAB file: myfactorial.m
`function [fac] = myfactorial(n)`
`fac = 1;`
`for k=n:-1:2`
`fac = fac*k;`
`end`
`end`

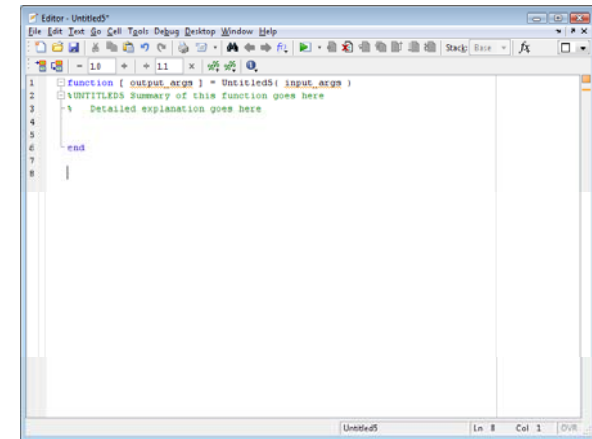
Matematik

MATLAB

MATLAB funktioners definition





MATLAB funktioners definition



Eksempel på MATLAB funktion

```
function [fac] = myfactorial(n)
%myfactorial(n) - Returns the factorial of an unsigned integer n
% In mathematics, the factorial is written n! and it is
% defined by n! = n*(n-1)*...*1 .
% This function is called "myfactorial" because MATLAB has a
% built-in function called "factorial".
fac = 1;
for k=n:-1:2
    fac = fac*k;
end
end
```

MATLAB funktioners egenskaber

- Man kan have flere input (komma-adskilt liste)
- Man kan have flere output (komma-adskilt liste)
- Variable der erklæres i en funktion, kan ikke bruges udenfor funktionen (scope)
- Man kan ikke *køre* en funktion!
 - F5 i editor
 - Grøn pil  eller skridtvis kørsel 
- Man kan kun *kalde* en funktion!
 - Eksempel: `veclength([4, 3])`

Brug af indbyggede funktioner

Eksempel: En vektors længde

```
>> v = [4 3];  
>> (v(1)^2 + v(2)^2)^0.5  
ans =  
    5  
>> sqrt(v(1)^2 + v(2)^2)  
ans =  
    5  
>> sqrt(dot(v, v))  
ans =  
    5  
>> norm(v)  
ans =  
    5
```

Generalisering

► En funktion generaliseres ved at kunne håndtere flere typer input

```
► function [len] = veclength(v)  
    len = sqrt(v(1)^2 + v(2)^2);  
end
```

```
► function [len] = veclength(v)  
    len = sqrt(dot(v, v));  
end
```

```
► function [len] = veclength(v)  
    len = norm(v);  
end
```