

PC les 1: MATLAB gebruiken

In deze les frissen we het gebruik van MATLAB op. We herhalen enkele commando's, en de basisbegrippen om numerieke algorithmen via MATLAB te schrijven. We doen dit aan de hand van enkele oefeningen. Voor dieper begrip, lees Appendix A van het cursusboek.

TODO: 10 min extra materiaal toevoegen

1 De MATLAB interface en werkwijzen

Start Matlab op en verken de standaard interface, Desktop->Desktop Layout -> default. Links staat de werkdirectory. In het midden het commando scherm.

Begin steeds met de werkdirectory te plaatsen op een nieuwe directory. Op het einde van de les kun je dan deze directory zippen en naar jezelf mailen.

Er zijn 3 manieren om met Matlab te werken:

1. Geef commando's in het command window. Tik bevoorbeeld:

```
>> syms x
>> taylor(sin(x),2,0)
```

```
ans =
```

```
x
```

```
>> taylor(sin(x),4,0)
```

```
ans =
```

```
x - x^3/6
```

Je heb nu de taylor ontwikkeling van sinus berekend rond 0. Druk F1 op taylor om de help te zien.

2. Een tweede manier is om de bovenstaande commando's op te slaan in een file. Druk op het icoon voor een nieuwe M-file, en paste de commando's in de M-file:

```
syms x
taylor(sin(x),2,0)
taylor(sin(x),4,0)
```

Sla op als file `oef1.m`. Je kan nu de oefening uitvoeren door in het commando scherm dit m file op te roepen. Je doet dat door gewoon `oef1` te tikken:

```
>> oef1

ans =

x

ans =

x - x^3/6
```

Alle lijnen die niet eindigen met `;` verschijnen als output in het commandoscherm.

Opdracht: Voeg `;` toe aan een lijn en verifieer dit.

3. De derde mogelijkheid is de definitie van functies. We kunnen bijvoorbeeld een functie maken om altijd de Taylor ontwikkeling van de sinus functie te berekenen. Maak een nieuwe M-file met naam `oef2.m`, en plaats volgende output in dit file:

```
function out = oef2(orde)
% oef2 is een functie die de Taylor ontwikkeling van
% sinus teruggeeft rond
% de oorsprong, tot op de gevraagde orde
    syms x
    out = taylor(sin(x), orde, 0);
end
```

Roep nu deze functie op door in het commandoscherm `oef2(4)` te schrijven. Merk op dat het voor functies verplicht is de naam van de

functie en de naam van het m file identiek te nemen.

Opdracht: Gebruik de oef2 functie in het commandoschermb. Ken het toe aan de variabele a zonder dat er output in het commandoschermb verschijnt.

Ga als volgt te werk voor de PC-oefeningen: probeer zaken uit in het commandoschermb, schrijf voor elke oefening een m file met alle commando's. Schrijf functies om in een oefening te gebruiken om alles overzichtelijk te houden.

2 Arrays

Het kernbegrip van Matlab zijn rijen van getallen, zogenaamde arrays. Matlab is geroemd voor zijn snelle bewerkingen op arrays. Neem de tijd om dit begrip goed te verstaan. Matlab is geen symbolisch pakket zoals Maple, het is een numeriek pakket, en de belangrijkste datastructuur is de array.

Oefening 3 Tik volgende commando's om te leren wat ze betekenen

```
a = 1:10
a(1)
a(7)
a = 1:2:10
a = linspace(1,10,10)
a = [1 2 3]
a = [1;2;3]
a(2)
a = [1 2 3 ; 4 5 6; 7 8 9]
a(2,2)
a(1,:)
a(:,1)
a(1:2,1:2)
size(a)
a = zeros(2,2)
a = ones(2,2)
a = eye(2)
```

Oefening 4 De standaard berekeningen op arrays bestaan ook elementsgewijze, wat vaak nodig is voor datamanipulatie. Voer volgende commando's uit:

```

a = [5 -1 8]
b = [9;2;10]
A = [5 -3 8; 9 2 10]
B = [10 7 4; -11 -15 11]
C = A + B
C = A * B'
a * b
b * a
dot(a,b)
cross(a,b)
3 * A
3 + A
A .* B
A ./ B
A .^ B

```

We hebben hier dus * voor het lineaire algebra matrix product, terwijl de operatoren met een . elementsgewijze zijn.

3 Controle

We zullen vaak controlestructuren nodig hebben zoals if, for, Eerst moeten we de logische operatoren invoeren. In Matlab betekent het getal 1 de boolean waarde True, en 0 de boolean waarde False. Verder is evenwel elke niet 0 waarde ook gelijk aan True

Oefening 5 Voor volgende op commando's uit:

```

5 > 4
5 <=5
5>6
4 == 6
4 ~= 6
4|0
4&0
~5

```

Oefening 6 Tik if in het commandoscherm en druk F1. Doe hetzelfde met for. Lees de help.

Oefening 7 De waarde van π kan berekend worden met de reeks

$$\pi = 4 \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{2n-1} = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right).$$

Schrijf een matlab functie dat de waarde van π berekent gebruik makende van de eerste n termen en dat de echte relatieve fout ook weergeeft. Om de echte waarde van π te hebben, gebruik de matlab variabele pi . Gebruik dan je eigen programma om π en de relatieve fout te berekenen voor $n = 10$, $n = 20$, en $n = 40$

4 Plotten

Voorbeeld Een voorbeeld van een plot van onze taylorontwikkeling. Maak een m file met volgende commando's:

```
>> x=linspace(0,pi,40);
>> y=sin(x);
>> y2=x;
>> y4=x - x.^3/6;
>> y6=x.^5/120 - x.^3/6 + x;
>> y8=x.^5/120 - x.^7/5040 - x.^3/6 + x;
>> plot(x,y,'r',x,y2,'k--',x,y4,'k-',x,y6,'r--',x,y8,'b-*')
>> axis([0,4,-2,2])
>> legend('Exact','2 termen','4 termen','6 termen','8 termen')
>> xlabel('x'); ylabel('y')
```

Hetzelfde via een functie:

```
function taylorplot(f,a,left,right,n)
%TAYLORPLOT: MATLAB function M-le that takes as input
%a function in inline form, a center point a, a left endpoint,
%a right endpoint, and an order, and plots
%the Taylor polynomial along with the function at the given order.
syms x
p = vectorize(taylor(f(x),n+1,a));
x=linspace(left,right,100);
f=f(x);
p=eval(p);
plot(x,f,x,p,'r')
```

De plots kunnen dan als volgt verkregen worden:

```

>> f=inline('sin(x)')
>> taylorplot(f,0,0,pi,1)
>> taylorplot(f,0,0,pi,2)
>> taylorplot(f,0,0,pi,4)
>> taylorplot(f,0,0,pi,6)
>> taylorplot(f,0,0,pi,8)

```

5 Symbolisch rekenen, oftewel, Maple in Matlab

Matlab kan ook symbolisch rekenen, het is evenwel geen vervanging van Maple op dat vlak. Enkele voorbeelden:

% Hoe afleiden van een functie in MATLAB?

```

syms x
y=7*exp(3*x)
dfdy=diff(y,x,1)
dfval=subs(dfdy,x,0.5)
xval = linspace(0,1,50);
plot(xval,subs(y,x,xval),'r',xval,subs(dfdy,x,xval),'b')

```

%hoe een vergelijking oplossen

```

syms x
eqn1=x^3+x^2-6*x+1
soln=solve(eqn1,x)
soln=double(soln)

```

%hoe integreren

```

syms x
y=x^3
val=int(y, x, 0.2, 0.3)
val=eval(val)
int('x^3', 0.2, 0.3)

```

6 Extra's

- Oef A.8

- Oef A.17