# Oefeningen Numerieke Wiskunde

Oefenzitting 1: Een eerste kennismaking met MATLAB

## 1 Inleiding

MATLAB is een pakket dat interactief wordt gebruikt en dat vooral geschikt is voor het werken met matrices. Vele functies uit welgekende pakketten voor lineaire algebra (LINPACK, EISPACK, ...) worden in MATLAB aangeboden. Zo zijn er bv. commando's beschikbaar voor het berekenen van de eigenwaarden en eigenvectoren van een matrix of voor de LU-decompositie van een matrix.

Deze tekst wil een inleiding zijn voor MATLAB en bevat enkel die functionaliteiten van MATLAB waarvan wij denken dat ze gekend moeten zijn om enigszins vlot met het pakket te kunnen werken. Voor verdere informatie verwijzen we naar de uitgebreide helpfunctie. Meer informatie, onder andere bibliotheken met functies, kan je ook altijd vinden op de homepage van The Mathworks: [http://www-europe.mathworks.com/].

MATLAB bevat een uitgebreide helpfunctie. Door het intypen van

```
>> doc
```

verschijnt het helpvenster, dat ook een ingebouwde zoekfunctie heeft. Verdere informatie over bijvoorbeeld de verschillende elementaire functies in MATLAB vindt men met het commando

```
>> doc elfun
```

Om te weten te komen wat de functie exp precies doet, typt men

```
>> doc exp
```

Een nadeel van deze manier van werken is dat men de naam van het commando moet kennen vooraleer men de documentatie hierover kan raadplegen. Als je het exacte commando niet kent, kan je de zoekfunctie gebruiken in het helpvenster of je kan beroep doen op lookfor.

```
>> lookfor eigenvalue
```

zoekt van alle m-files (zie Sectie 5) de eerste commentaarregel af en vermeldt diegene die het opgegeven woord bevatten. Als je dus zelfgeschreven m-files hebt en je voorziet die van nuttige commentaar, dan kan het commando lookfor die terugvinden. Verder kan je in plaats van doc ook het commando

```
>> help exp
```

gebruiken. De documentatie verschijnt hierdoor in de Command Window waarin je aan het werken bent, wat vaak handiger is.

# 2 Basiscommando's

Bij het werken met MATLAB krijg je op het scherm telkens een prompt. Dit is: >> . Hierachter kan je de gewenste bevelen intikken. Als je op de enter-toets drukt, wordt het bevel uitgevoerd. Merk op dat je enkel achtereenvolgens commando's kan ingeven. Het is niet mogelijk om bijvoorbeeld met de muis op een eerder uitgevoerd commando te klikken om dit te wijzigen. Er bestaat wel de mogelijkheid om met de pijltjes toetsen te navigeren doorheen de commando's die je eerder ingegeven hebt. Zo kan je eenvoudig een commando opnieuw onder de cursor halen en eventueel aanpassen.

## 2.1 Het invoeren van variabelen

### Getallen

```
< naam-van-de-variabele > = < waarde >
```

Voorbeeld:

```
>> x = 0.65
```

heeft als resultaat dat MATLAB aan de variabele x de waarde 0.65 toekent. MATLAB antwoordt :

```
x =

    0.6500
```

### Vectoren

Vectoren worden aangemaakt door getallen gescheiden door een spatie, een komma of een puntkomma te plaatsen tussen vierkante haken. Als de getallen gescheiden worden door een puntkomma, is de vector een kolomvector, anders is het een rijvector.

```
 >> v = [1 8 6 7]

v =
1 8 6 7

 >> v = [1; 8 ;6 ;7]

v =
1
8
6
7
```

Vectoren die bestaan uit equidistante elementen kan je aanmaken door gebruik te maken van de dubbelpuntoperator. Voorbeeld:

```
 a:b:c
```

maakt een vector aan met als elementen:

```
 a, a+b, a+2b, ..., c
```

**Matrices**

Matrices worden rij per rij gedefinieerd. Een vierkante haak [ maakt MATLAB duidelijk dat er een matrix volgt. Een matrix wordt afgesloten met ]. Je kan een matrix op 2 manieren invoeren. Voorbeeld: Je wil

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}$$

dan tik je ofwel

```
>> A = [ 0 1 2 ; 3 4 5 ; 6 7 8 ]
```

ofwel

```
>> A = [ 0 1 2
         3 4 5
         6 7 8 ]
```

In beide gevallen antwoordt MATLAB :

```
A =

     0     1     2
     3     4     5
     6     7     8
```

De rijen worden dus gescheiden door een ; of door een enter, de verschillende elementen van een rij worden gescheiden door een blanco of eventueel door een komma.

## 2.2 De uitvoer

De inhoud van een variabele kan je opvragen door gewoon de naam van de variabele in te tikken, en dan op de enter-toets te drukken.

```
>> x
```

MATLAB antwoordt :

```
x =

    0.6500
```

Dit is het standaard uitvoerformaat van MATLAB. Wetenschappelijke notatie verkrijg je met het bevel:

```
>> format short e
```

Wil je meer cijfers van je getal zien, dan kan dat met de bevelen

```
>> format long
```

of

```
>> format long e
```

Na het intikken van dit laatste bevel, beantwoordt MATLAB

```
>> x
```

met

```
x =

     6.500000000000000e-01
```

Meerdere opties wat uitvoer van getallen betreft, kan je opvragen met het commando

```
>> doc format
```

Merk op dat MATLAB steeds uitvoer geeft als een commando wordt ingetypt. Om dit te vermijden (wat bijvoorbeeld handig is als men grote matrices als uitvoer verwacht) kan je achter het commando een ; plaatsen.

## 2.3 Selecteren van elementen

Je kan elementen uit een vector selecteren door gebruik te maken van ronde haakjes:

| | |
|---|---|
| `v(i)` | geeft het i-de element van een rij- of kolom-vector terug. |
| `v([1,3,4])` | geeft het eerste, derde en vierde element van de vector v terug |

Je kan matrix-elementen als volgt selecteren :

| | |
|---|---|
| `A(i,j)` | geeft het element op de i-de rij en de j-de kolom van A. |
| `A(i,:)` | geeft de hele i-de rij van A. |
| `A(:,j)` | geeft de hele j-de kolom van A. |
| `A(i:j,k:l)` | geeft de deelmatrix A(i ... j , k ... l). |
| `A(end,end)` | geeft het laatste element in de matrix. |

Voorbeeld:

```
>> M = A(1:2,2:3)
```

geeft

```
M =

     1     2
     4     5
```

## 2.4 Rekenkundige operatoren

De volgende rekenkundige bewerkingen kunnen zowel voor scalars als voor vectoren en matrices gebruikt worden:

| | |
|---|---|
| `+` | optelling |
| `-` | aftrekking |
| `*` | vermenigvuldiging |
| `/` | rechtse deling (`a/b` betekent a*inv(b) ) |
| `\` | linkse deling (`a\b` betekent inv(a)*b ) |
| `^` | machtsverheffing |

De volgende rekenkundige bewerkingen kunnen zowel voor vectoren als voor matrices gebruikt worden:

| | |
|---|---|
| `.*` | elementsgewijze vermenigvuldiging van matrices |
| `./` | elementsgewijze deling van matrices |
| `.^` | elementsgewijze machtsverheffing matrices |

Voorbeeld:

```
>> N = A(1:2,1:2);
>> P=N * M
P =
     4     5
    19    26
```

berekent het matrixproduct van de matrices `N` en `M`. Uiteraard moeten deze matrices de juiste dimensies hebben. Om de dimensie van de matrix `M` op te vragen kan men het commando

```
>> size(M);
```

gebruiken. Om de elementen van twee matrices of vectoren (van dezelfde dimensie) elementsgewijze te vermenigvuldigen, gebruikt men `.*` Voorbeeld:

```
>> v = [ 9 5 4 ];
>> w = [ 6 7 0 ];
>> u = v .* w

u =

    54    35     0
```

Als je het resultaat van een bewerking niet toekent aan een variabele, wordt hiervoor automatisch de variabele ans gebruikt.

```
>> 2^5

ans =

    32

>> ans - 5

ans =

    27
```

### 2.5 Relationele operatoren

| | |
|---|---|
| `==` | gelijkheid |
| `~=` | ongelijkheid |
| `<` | kleiner dan |
| `<=` | kleiner dan of gelijk |
| `>` | groter dan |
| `>=` | groter dan of gelijk |

Het resultaat van een logische uitdrukking is 1 als ze waar is en 0 als ze onwaar is.
Voorbeeld:

```
>> 8 == ( 2^3 )

ans =

     1

>> 2 > 6

ans =

     0
```

Merk op dat ronde haakjes kunnen worden gebruikt om de volgorde van operatoren op te leggen. Strikt genomen zijn deze in bovenstaand voorbeeld overbodig omdat de machtsverheffing een hogere prioriteit heeft dan de gelijkheid. Meer informatie over deze operatoren en nog andere kan men bekomen met

```
>> doc ops
>> doc arith
>> doc relop
```

Informatie over de volgorde van operatoren kan je vinden door in het help venster te zoeken op 'operator precedence'.

## 3 Functies in MATLAB

### 3.1 Enkele elementaire functies in MATLAB

log, log10, exp, sin, cos, tan, asin, sinh, asinh, ...

Het argument geef je in tussen ronde haakjes. Voorbeeld:

```
>> log(1)
```

geeft als antwoord:

```
ans =

     0
```

## 3.2 Algemene MATLAB-functies

Resultaten van berekeningen van MATLAB worden vaak elders weer opnieuw gebruikt. De waarden van variabelen kunnen bewaard worden in een .mat-bestand. Dit gebeurt met de opdracht

```
>> save filename
```

waardoor alle gebruikte variabelen worden bewaard in het bestand met naam 'filename.mat' De extensie .mat hoeft niet opgegeven te worden. Hoeven er maar een aantal variabelen te worden opgeslagen, dan dienen die gespecificeerd te worden na de bestandsnaam. Het opgeslagen bestand kan in MATLAB worden ingelezen met de opdracht

```
>> load filename
```

waarbij de extensie .mat mag worden weggelaten. De variabelen met hun waarden kunnen nu weer gebruikt worden.

Volgende tabel geeft een overzicht van verschillende algemene MATLAB-functies:

| | |
|---|---|
| who | geeft een lijst van de variabelen die je gebruikt. |
| clear | geeft de gebruikte geheugenruimte terug vrij. |
| save | bewaart de variabelen van de huidige MATLAB-sessie in een .mat-bestand. |
| load | leest de variabelen uit een .mat-bestand terug in zodat deze terug de waarden hebben van net voor de laatste save. |
| pause | laat Matlab wachten op de gebruiker |
| path | zet hierin de padnamen van de directories waar MATLAB naar functies (.m-bestanden) moet zoeken. |
| quit | sluit MATLAB af. |

Voor meer informatie over ingebouwde functies:

```
>> doc elfun
>> help general  (dus hier help ipv. doc)
>> doc <functienaam>
>> lookfor <term>
```

## 3.3 Voorgedefinieerde variabelen

| | |
|---|---|
| `pi` | het getal pi. |
| `Inf` | oneindig |

## 3.4 Functies voor getallen

| | |
|---|---|
| `abs` | geeft de absolute waarde of de modulus van een complex getal. |
| `round` | geeft het dichtstbij gelegen geheel getal. |

### 3.5 Functies voor vectoren en/of matrices

| | |
|---|---|
| `inv` | berekent de inverse matrix (vierkante matrix !). |
| `det` | berekent de determinant (vierkante matrix !). |
| `cond` | berekent het conditiegetal van een matrix. |
| `rank` | bepaalt de rang van een matrix. |
| `length` | geeft het aantal elementen van een vector terug. |
| `size` | bepaalt het aantal rijen en kolommen van een matrix. |
| `norm` | berekent de norm van een matrix of van een vector. |
| `eye(n)` | geeft de eenheidsmatrix terug van dimensie n. |
| `diag(A)` | als A een vector is, geeft dit een diagonaalmatrix met de elementen van A op de diagonaal.<br>als A een matrix is, geeft dit een vector met de diagonaalelementen van A.<br>kan ook gebruikt worden voor onder- en bovendiagonalen, zie help diag. |

| | |
|---|---|
| `lu` | geeft de factoren terug na Gauss-eliminatie (met optimale rij-pivotering). |
| `chol` | geeft de Cholesky-factorizatie. |
| `eig` | berekent eigenwaarden en eigenvectoren. |
| `svd` | berekent de singuliere waarden ontbinding. |
| `rand(m,n)` | geeft een (m x n)-matrix met randomgetallen, uniform verdeeld tussen 0 en 1. |
| `A'` | geeft de getransponeerde van de matrix A. |
| `A(:)` | zet alle kolommen van A onder elkaar in een vector. |
| `ones(m,n)` | geeft een matrix van dimensie m x n met allemaal eentjes. |
| `zeros(m,n)` | geeft een matrix van dimensie m x n met allemaal nullen. |
| `reshape(A,m,n)` | geeft de matrix A dimensie m x n. |
| `fliplr(A)` | draait de matrix A om in de links/rechts richting.<br>(noot: dit is equivalent met `A(:,end:-1:1)`) |
| `flipud(A)` | draait de matrix A om in de onder/boven richting. |
| `tril(A)` | geeft de onderdriehoeksmatrix van A. |
| `triu(A)` | geeft de bovendriehoeksmatrix van A. |

Voorbeelden: Na het intikken van

```
>> [L,U] = lu(A)
```

geeft MATLAB twee matrices L en U terug; L is een benedendriehoeksmatrix met 1-tjes op de diagonaal en U is een bovendriehoeksmatrix. Om een stelsel Ax = b op te lossen, gebruik je de MATLAB-deling

```
>> x = A \ b
```

Hierbij wordt impliciet ook Gauss-eliminatie met optimale rij-pivotering gebruikt. Als A geen vierkante matrix is (dit betekent dat het aantal vergelijkingen verschilt van het aantal onbekenden), dan krijg je de kleinste kwadraten oplossing.

```
>> [V,D] = eig(A)
```

geeft een diagonaal matrix D met de eigenwaarden en een volle matrix V waarvan de kolommen de eigenvectoren zijn zodat A V = V D. (Hier moet A een vierkante matrix zijn!) Voor verdere informatie over de commando's:

```
>> doc <commando>
>> doc matfun  overzicht van de functies voor matrices.
>> doc sparfun  functies voor spaarse (ijle) matrices.
```

### 3.6 Opdrachten

1. Voer de twee opdrachten

   ```
   >> f = 2
   ```

   en

   ```
   >> g = 3 + f
   ```

   uit. Sla de variabelen f en g op in het bestand probeer.mat. Verlaat MATLAB, start MATLAB opnieuw op en laad de file probeer.mat. Kijk of de variabelen f en g bekend zijn. Waar is het bestand probeer.mat terecht gekomen? Verwijder het bestand probeer.mat.

2. Bekijk de helpfunctie van het commando diary. Maak gebruik van dit commando om volgende opdracht uit te voeren: open een tekstbestand 'afschrift' om in- en uitvoer op te slaan. Voer de opdrachten

   ```
   >> f = 2*3
   >> g = f+2
   ```

   uit en beëindig het opslaan van in- en uitvoer. Bekijk de inhoud van 'afschrift'. Waar is het bestand terecht gekomen? Verwijder dit bestand.

3. Evaluatie van de opdracht pause(30) heeft als effect dat MATLAB 30 seconden pauze houdt. Voer deze opdracht uit en pas snel de toetscombinatie Control-C toe. Wat is het effect?

4. Bereken de volgende uitdrukkingen met Matlab:

   - $sin(\frac{\pi}{4})$
   - $e^2$
   - $\frac{tan(x)}{\sqrt{1+x^2}}$, met $x = 4$

5. Maak een kolomvector met als elementen 20,18,16,...,2.

6. Maak een rijvector $v$ met 100 elementen van de vorm: $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$.

7. Maak voor de volgende functies f de vector (f(0), f(0.1), . . . , f(2)).

   (a) $f(x) = x^2 + 2x + 1$.

(b) $f(x) = \dfrac{3^x}{1+3^x}$

(c) $f(x) = \dfrac{x}{1+\sqrt{x}}$

8. Maak twee matrices

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix} \text{ en } H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{pmatrix}$$

   (a) Bereken het verschil tussen $G$ en $H$.

   (b) Vermenigvuldig $G$ met $H$ elementsgewijs.

   (c) Vermenigvuldig $G$ met $H^T$ .

   (d) Bewaar de eerste drie kolommen van de matrix $G$ in $G3$ en bereken $G3^2$.

9. Maak met behulp van `rand` een random 10 bij 10 matrix $R$ en bereken zijn inverse. Controleer of hun product gelijk is aan de eenheidsmatrix.

10. Gebruik het commando `diag` om een diagonaalmatrix met de elementen $10, 20, 30, \ldots 80$ aan te maken. Bereken daarna de determinant van deze matrix en controleer dat deze gelijk is aan het product van de elementen op de diagonaal. De functies `prod` en `det` kunnen hiervoor gebruikt worden.

# 4 Constructies: if, for, while

Deze constructies kunnen zowel interactief als in een .m-bestand worden gebruikt.

## 4.1 if - elseif - else - end

syntaxis: if expression, statements, else, statements, end
Voorbeeld:

```
>> if (i==j)
     i=j+1;
     j=i/2;
   else
     j=2*i;
   end
```

Voor een uitgebreider voorbeeld verwijzen we naar de helpfunctie.

## 4.2 for - end

syntaxis: for variable = vector, statement, ..., statement end
Voorbeeld:

```
>> x=0;
>> for i=1:5,
     x=x+i;
   end
```

### 4.3 while - end

syntaxis: while condition, statement, ..., statement, end
Voorbeeld:

```
>> x = 0;
>> y = 0;
>> while (x <= 5),
     x=x+1;
     y=y+x;
   end
```

## 5 Zelf .m-bestanden schrijven

### 5.1 Functies

In MATLAB kan je zelf ook functies schrijven om complexe berekeningen uit te voeren. Zodoende kan je het pakket naar je eigen behoefte verder uitbouwen. Een dergelijke functie wordt gewoon met een editor geschreven (je gebruikt best het bevel edit) en opgeslagen (in ASCII-formaat) in een bestand met extensie '.m'. De eerste regel van een functie bevat bijvoorbeeld:

```
>> function [x,y] = naam(a,b,c)
```

Als je dit bestand opslaat met de naam naam.m, kan je deze functie in MATLAB oproepen als:

```
>> [x,y] = naam(a,b,c)
```

De parameters a, b, c zijn de gegevens die aan de functie worden doorgegeven. Het resultaat moet binnen de functie worden toegekend aan de variabelen x en y. Uiteraard moeten de namen van de variabelen en parameters binnen het .m-bestand en bij het oproepen van de functie niet hetzelfde zijn. Let op dat de uitvoerparameters tussen vierkante haken staan en de invoervariabelen tussen ronde haken. Als er één of geen enkele uitvoerparameter is, kan je de vierkante haken weglaten. De ronde haken kan je weglaten als er geen invoer nodig is. Na deze functiehoofding voeg je best wat commentaar toe over het doel van de functie, en bijvoorbeeld de volgorde waarin je de argumenten moet geven. Dit kan doen door je regel te laten beginnen met een `%-teken.` Deze eerste commentaarlijnen verschijnen als antwoord als je tikt

```
>> help naam
```

Je kan uiteraard ook op andere plaatsen in je functie commentaar schrijven. Vergeet niet om achter elk commando een ; te schrijven, anders krijg je bij het uitvoeren van de functie alle tussenresultaten op je scherm. Om de functie te kunnen verlaten voor de laatste opdracht is uitgevoerd, bv. bij het slagen van een bepaalde test, gebruik je return.

### 5.2 Scripts

Indien je een functie wenst te schrijven zonder invoer en uitvoerparameters, dan hoef je de eerste regel

```
>> function [x,y] = naam(a,b,c)
```

niet toe te voegen aan je bestand. Het resulterende bestand is nu geen functie maar een script dat je kan uitvoeren door in de editor op de run knop te duwen of door de naam van het bestand in te geven in het Command Window.

### 5.3 Opdrachten

1. De periodieke functie f met periode 2 is gegeven door:

$$f(t) = \begin{cases} t^2 & , \quad 0 \leq t \leq 2 \\ f(t-2) & , \quad 2 \leq t \\ f(t+2) & , \quad t < 0 \end{cases}$$

   Definieer deze functie in MATLAB. Zorg ervoor dat de functie ook voor vectoren werkt. Je kan testen of je functie werkt door de commando's

```
>> t = linspace(-3,3,1000);
>> figure
>> plot(t,f(t))
```

   uit te voeren en te vergelijken met Figuur 1.

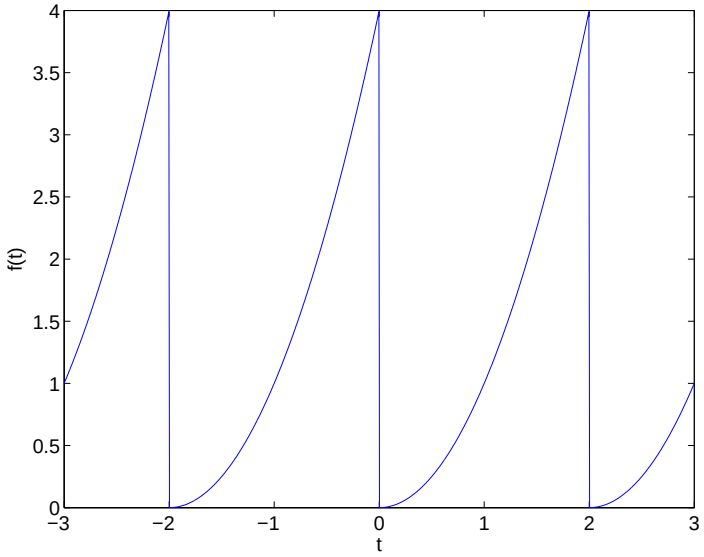
2. Maak een eigen functie met de naam h bij de functie $h(x) = x^2$ die bestand is tegen vector-operaties. Voer de volgende opdrachten in de onderstaande volgorde uit.

   (a) `>> h([1,2,3,4])`
   (b) `>> h = 2.5`
   (c) `>> h([1,2,3,4])`
   (d) `>> h(1)`
   (e) `>> h(1.4)`
   (f) `>> clear h`

   Wordt bij de onderdelen (c) tot en met (e) h als een functie of als een variabele gezien? Waarom volgt er bij onderdeel (d) geen foutmelding? Kent MATLAB de functie h nog?

## 6 Grafieken

Grafieken maken kan met het commando plot. Zie de helpfunctie voor de precieze werking ervan. Onderaan de online-helpdocumentatie bij dit commando zie je een overzicht van een reeks commando's die verband houden met de plot functie. Soms kan het nuttig zijn om de grafiek uit te zetten in een logaritmische schaal. semilogx gebruikt een logaritmische schaal voor de horizontale as, semilogy voor de verticale as en loglog voor de beide assen. Vooral bij




Figuur 1: $f(t)$ van Opdracht 5.3.1

het uitzetten van een foutenkromme kan dit nuttig zijn. Als je een grafiek hebt uitgezet op een figuur en je geeft opnieuw een plotcommando, zal je merken dat de vorige grafiek wordt gewist. Om toch meerdere grafieken op één figuur te zetten, gebruik je het commando hold on. Het commando hold off zet dit terug af. Om de grafieken later nog te kunnen begrijpen kan je de naam van de variabele vermelden die volgens een as is uitgezet (commando xlabel en ylabel). Ook kan je de grafiek een titel geven (commando title). Een legende toevoegen (handig als je meerdere grafieken op één figuur uitzet) kan met het commando legend. Je kan je grafiek afdrukken of opslaan in een bestand met het commando print. Meerdere grafieken openen, kan je met het commando figure. Raadpleeg weerom de helpfunctie voor de precieze werking van deze commando's.

## 6.1 Opdrachten

1. Teken de functie $f(x) = \dfrac{sin(x) + 2x}{1 + x^2}$ op het interval $[0, 2\pi]$

2. Schrijf een script tekening.m waarin de functies sin(t) en cos(t) getekend worden op het interval $0 \leq t \leq 2$ in een figuur. Zorg ervoor dat de sinus in het rood en de cosinus in het groen getekend wordt. Benoem de assen en voorzie een legende.