

TECHNISCHE UNIVERSITEIT DELFT

Faculteit Informatietechnologie en Systemen

Technische Wiskunde en Informatica

Vakgroep Toegepaste Analyse

Groep Numerieke Wiskunde

PRACTICUM : Numerieke Analyse (wi211)  
COMPUTERSYSTEEM : HP Werkstations en Linux PC's  
PROGRAMMEEROMGEVING : Matlab

Door de faculteit der Informatietechnologie en Systemen ter beschikking gestelde software en documentatie mag niet worden gekopieerd, aan derden ter hand gesteld, of anderszins onrechtmatig worden gebruikt.

Oktober 1998

# Inhoud

<b>0</b>	<b>GEBRUIK VAN DE HANDLEIDING</b>	<b>3</b>
0.1	Leeswijzer . . . . .	3
0.2	De opbouw van de handleiding . . . . .	3
<b>1</b>	<b>ALGEMEEN</b>	<b>4</b>
1.1	Practicumadministratie . . . . .	4
1.2	De handleiding . . . . .	4
1.3	Practicumapparatuur . . . . .	4
1.4	Uitsluiting . . . . .	4
<b>2</b>	<b>HET BESTURINGSSYSTEEM</b>	<b>5</b>
2.1	UNIX . . . . .	5
2.1.1	Inloggen . . . . .	5
2.1.2	Uitloggen . . . . .	5
2.1.3	Van directory veranderen: cd . . . . .	6
2.1.4	Files . . . . .	6
2.1.4.1	kopieren: cp . . . . .	6
2.1.4.2	afdrukken: lp . . . . .	6
2.1.4.3	overzicht: ls . . . . .	6
2.1.4.4	-naam veranderen: mv . . . . .	7
2.1.4.5	verwijderen: rm . . . . .	8
2.2	Het X-window systeem . . . . .	8
<b>3</b>	<b>PROGRAMMATUUR</b>	<b>9</b>
3.1	Matlab . . . . .	9
3.1.1	Hoe te werken met Matlab ? . . . . .	9
3.2	De Editor . . . . .	9
3.2.1	Starten van de editor . . . . .	9
3.2.2	Help . . . . .	10
3.2.3	Nieuwe bestanden maken . . . . .	10
3.2.4	Bestanden oproepen en bewaren . . . . .	10
3.2.5	Teksten invoeren . . . . .	11
3.2.5.1	Cursorbesturing . . . . .	11
3.2.5.2	Invoegen en verwijderen van tekst . . . . .	11
3.2.5.3	Het werken met blokken . . . . .	11
<b>4</b>	<b>BASISCURSUS</b>	<b>12</b>
4.1	Matrices en vectoren. Bewerkingen met matrices en vectoren . . . . .	12
4.2	Elementaire wiskundige functies . . . . .	14
4.3	Rationele en logische operatoren . . . . .	15
4.4	Herhalingsopdrachten. Voorwaardelijke opdrachten. . . . .	15

4.5	Script-files. In- en uitvoer . . . . .	17
4.6	Function-files . . . . .	19
4.7	Plotten . . . . .	19
4.8	Het oplossen van een stelsel vergelijkingen . . . . .	21
4.9	Voorbeeldprogramma . . . . .	22
<b>5</b>	<b>REFERENTIE</b>	<b>24</b>

# **0 GEBRUIK VAN DE HANDLEIDING**

## **0.1 Leeswijzer**

In deze practicumhandleiding vindt u de informatie die u nodig heeft voor het gebruik van de bij het practicum behorende programmatuur. Voor de informatie betreffende de inhoudelijke/organisatorische kant van het practicum wordt u verwezen naar het informatiestencil.

Veel succes met het practicum!

## **0.2 De opbouw van de handleiding**

Hoofdstuk 1 bevat een aantal algemene opmerkingen met betrekking tot het practicum.

Een korte omschrijving met betrekking tot het bij dit practicum gebruikte besturingssysteem vindt u in hoofdstuk 2.

In hoofdstuk 3 wordt de bij dit practicum gebruikte software besproken.

Hoofdstuk 4 bevat een korte cursus "Werken met Matlab".

In hoofdstuk 5 vindt u een korte beschrijving van de functies die bij het practicum gebruikt mogen worden.

# **1 ALGEMEEN**

Het practicum wordt verzorgd onder verantwoordelijkheid van de vakgroep Toegepaste Analyse van de faculteit der Informatietechnologie en Systemen. Het practicum wordt ondersteund door de practicumadministratie van de sectie TMV. De begeleiding tijdens het practicum geschiedt door medewerkers van de vakgroep Toegepaste Analyse.

## **1.1 Practicumadministratie**

De practicumadministratie is u bij al uw problemen met en/of vragen over de administratieve kant van het practicum behulpzaam. De practicumadministratie is gehuisvest in kamer 0.200, Gebouw Julianalaan 132, tel. (015-278)4588 en geopend op:

maandag t/m vrijdag van 9:30 tot 12:00 uur en van 13:00 tot 16:30 uur.

## **1.2 De handleiding**

Opmerkingen met betrekking tot de inhoud van de handleiding kunt u, doorgeven aan de verantwoordelijke docent. In de volgende versie van de handleiding zal getracht worden deze opmerkingen zo goed mogelijk te verwerken.

## **1.3 Practicumapparatuur**

Het practicum wordt verricht in practicumzaal HB06.230, Gebouw Mekelweg 4. In de zaal staan werkstations en Linux PC's, opgenomen in een netwerk.

Het netwerk zorgt voor systeemsoftware, gebruikerssoftware, het gebruik van een inkjet-printer, een laserprinter en de harddisks. Nadat u door de practicumadministratie voorzien bent van een practicumnummer en een password kunt u toegang krijgen tot het netwerk.

## **1.4 Uitsluiting**

Indien u de apparatuur of programmatuur niet gebruikt voor het doel waarvoor deze ter beschikking zijn gesteld kan uitsluiting van het practicum het gevolg zijn.

## 2 HET BESTURINGSSYSTEEM

Tijdens het practicum maakt u gebruik van UNIX<sup>1</sup> en het X Window Systeem. In dit hoofdstuk worden beide systemen kort besproken. Wilt u meer weten over deze omgevingen dan wordt u verwezen naar het practicum Computervaardigheden (in252).

### 2.1 UNIX

UNIX is een multi-tasking/multi-user besturingssysteem voor, oorspronkelijk, microcomputers. Tijdens het practicum heeft u slechts een beperkt aantal commando's nodig, deze worden hieronder besproken.

#### 2.1.1 Inloggen

Bij een multi-user omgeving is het noodzakelijk dat u moet inloggen op een systeem, niet iedere gebruiker heeft immers dezelfde rechten op het systeem. In het algemeen zal de machine, een HP-werkstation of een "Linux pc", waarop u werkt aanstaan. Het beeldscherm zal meestal niet aanstaan. Zet het beeldscherm aan, op het scherm staat onder nadere de tekst:

```
login:
```

U dient hier uw loginnaam (= practicumnummer) in te voeren. Na op de [ENTER]-toets gedrukt te hebben wordt er gevraagd een password in te voeren. Sluit het invoeren van het password (zie practicumsticker) af met [ENTER].

Als u één van beide of beide fout heeft ingevoerd wordt u de toegang tot het systeem geweigerd. U dient de procedure te herhalen.

Als u geen fouten heeft gemaakt krijgt u toegang tot het systeem. U komt altijd, onafhankelijk van het werkstation, in dezelfde directory (uw HOME-directory).

Na het inloggen verschijnt op het scherm een window genaamd "console". Hierin kunnen unix opdrachten worden ingetikt. Verder verschijnt er een window genaamd "Matlab Commando Scherm". Hierin kunnen de matlab opdrachten gegeven worden. Onderaan het scherm verschijnt een menubalk met daarop onder andere iconen met `nedit`, `xterm` en `quit`.

#### 2.1.2 Uitloggen

Om uit te loggen gaat u met de (muiscursor) naar Exit en houd u de linker muisknop ingedrukt. Er verschijnt een menu met:

```
Really quit?  
Yes  
No
```

Klik op Yes om uit te loggen.

---

<sup>1</sup>UNIX is een geregistreerd handelsmerk van AT&T

### 2.1.3 Van directory veranderen: cd

Naam: cd = change directory  
Aanroep: cd *dirnaam*  
Omschrijving: Directory *dirnaam* wordt de nieuwe actieve directory voor de gebruiker. De directory *dirnaam* moet wel reeds bestaan.  
Uitbreiding: Als alleen cd wordt gegeven, dus cd zonder *dirnaam*, dan wordt de HOME-directory weer de actieve directory.

### 2.1.4 Files

#### 2.1.4.1 kopiëren: cp

Naam: cp = copy  
Aanroep: cp file1 file2  
Omschrijving: file1 wordt gekopieerd. De kopie krijgt de naam file2.  
Als file2 al bestaat, wordt de inhoud van file2 overschreven door de inhoud van file1.

#### 2.1.4.2 afdrukken: lp

Naam: lp = lineprinter (hier laser printer)  
Aanroep: lp file1  
Omschrijving: file1 wordt in de printerqueue geplaatst en afgedrukt op de laserprinter in de zaal.  
Opmerking: om een file op de matrix printer af te drukken gebruik  
matrix file1

#### 2.1.4.3 overzicht: ls

Naam: ls = list  
Aanroep: ls  
Omschrijving: De namen van alle gewone gebruikersfiles in een directory worden afgedrukt.  
Uitbreiding: Als achter het ls commando nog -s wordt toegevoegd, dan wordt voor iedere filenaam de grootte van de file in blokken weergegeven.

#### **2.1.4.4 -naam veranderen: mv**

Naam: mv = move

Aanroep: mv file1 file2

Omschrijving: file1 wordt hernoemd naar file2: file1 is de oude naam en file2 is de nieuwe naam.

Als file2 al bestaat, wordt de inhoud van file2 overschreven door de inhoud van file1.



### **2.1.4.5 verwijderen: rm**

Naam: rm = remove

Aanroep: rm file1

Omschrijving: file1 wordt verwijderd van schijf.

## **2.2 Het X-window systeem**

U kunt een nieuw window aanmaken door op xterm te klikken. Een window wordt actief door er met de muiscursor op te klikken.

## 3 PROGRAMMATUUR

Tijdens dit practicum zult u werken met Matlab en met Nedit voor het maken van files.

### 3.1 Matlab

De naam Matlab staat voor Matrix Laboratory. Matlab is een interactief programma met als basiselement een matrix waarvan de dimensies niet gedeclareerd hoeven te worden. Hierdoor kunnen op eenvoudige wijze numerieke methoden geïmplementeerd worden.

#### 3.1.1 Hoe te werken met Matlab ?

Na het inloggen start Matlab direct. Er verschijnt een window met de tekst "Matlab Commando Scherm" in het kader.

Matlab werkt interactief. Dit houdt in dat wanneer u een commando geeft, afgesloten met **[RETURN]**<sup>2</sup>, dit direct uitgevoerd wordt door het programma en het resultaat ervan op het beeldscherm verschijnt. U kunt vorige commando's terug halen met de cursortoetsen ↑ en ↓. Met **[Esc]** krijgt u weer een lege commandoregel.

Meer complexe berekeningen kunt u maken door met behulp van de editor een zogenaamde script-file aan te maken, een file bestaande uit een opeenvolging van Matlab-commando's. Hierbij kan gebruik worden gemaakt van de vele in Matlab aanwezige functies. Daarnaast kunt u zelf functies definiëren met behulp van function-files. Variabelen kunnen worden opgeslagen in data-files.

Het uitvoeren van een script-file wordt gestart door het intoetsen van de filenaam.

Met **[Ctrl]+c** kunnen commando's onderbroken worden.

Hoofdstuk 4 (Basiscursus) bevat een korte cursus "Werken met Matlab". Het wordt u aangeraden deze cursus door te werken alvorens u met het programmeren van uw uitwerkingen begint. In hoofdstuk 5 (Referentie) wordt van alle functies die op het practicum gebruikt mogen worden een korte beschrijving gegeven.

### 3.2 De Editor

De editor die gebruikt wordt (Nedit) is zodanig gemaakt dat het gebruik van een handleiding in principe niet nodig is. Om veelvuldig gebruik van de helpfunctie te voorkomen volgt in deze paragraaf een beschrijving van de bij het practicum benodigde commando's.

#### 3.2.1 Starten van de editor

Nedit kan gestart worden door op het Nedit icon te klikken of door Nedit in te tikken. Er verschijnt een window waarin u kunt editten.

---

<sup>2</sup>Ieder commando of -regel in deze handleiding wordt afgesloten met **[RETURN]**, ook als dit niet vermeld staat.

Bovenin het window ziet u de volgende regel:

FILE EDIT SELECT SEARCH PREFERENCES SHELL MACRO WINDOWS HELP

Zodra u met de muiscursor op een van deze opties gaat staan en de linkermuisknop indruk verschijnt een aantal commando's. U kunt een keuze maken door de linkermuisknop ingedrukt te houden en vervolgens met het pijltje het gewenste commando aan te wijzen. Het los laten van de muisknop activeert het commando.

### **3.2.2 Help**

Het Help-commando geeft u de beschikking over een aantal hulp-functies. Normaal is alleen getting started voor u interessant.

### **3.2.3 Nieuwe bestanden maken**

Nadat Nedit is gestart verschijnt er een leeg window. Het is nu mogelijk tekst in te voeren. Zodra u de tekst in een bestand wilt opslaan of zodra u Nedit verlaat wordt er gevraagd het bestand een naam te geven.

U kunt ook een nieuw bestand maken met behulp van het commando NEW in het File-menu.

### **3.2.4 Bestanden oproepen en bewaren**

Het File-commando wordt gebruikt om tekst-bestanden op te roepen, te bewaren. Plaats de muiscursor op het woord File en druk op de linkermuisknop. Er verschijnt nu een lijstje met nieuwe commando's. Hou de muisknop ingedrukt en kies het gewenste commando door dit aan te wijzen. Het loslaten van de muisknop activeert het commando.

#### *Bestaande bestanden oproepen*

Kies uit het File-menu de optie Open. Er verschijnt een window waarin u de bestandsnaam kunt intikken. U kunt kiezen of u een nieuw window wilt openen of dat u in het al geopende window wilt werken.

#### *Bestanden bewaren*

Kies uit het File-menu een van de volgende opties:

Save: bij de eerste keer bewaren wordt u gevraagd een bestandsnaam in te voeren. Daarna krijgt u alleen te zien dat het bestand inderdaad bewaard is.

Save As: u kunt het bestand onder een andere naam bewaren.

Door in het ok-blokje te klikken laat u dit verdwijnen en kunt u verder met het intikken van uw bestand.

### 3.2.5 Teksten invoeren

Bij het maken van bestanden staan er een aantal hulpmiddelen om snel door de tekst te bladeren of met tekst te manipuleren tot uw beschikking.

#### 3.2.5.1 Cursorbesturing

Met behulp van de muis kan de cursor op een willekeurige plek in de tekst worden gezet. Ook met behulp van de scroll-bar aan de rechterzijde van het window kan door de tekst gebladerd worden. Plaats de muiscursor in de balk en klik op een van beide pijltjes of op een willekeurige positie in de balk. U ziet de tekst verschuiven in het window.

#### 3.2.5.2 Invoegen en verwijderen van tekst

Teksten kunt u invoeren door de cursor op een positie neer te zetten en de tekst in te typen. De tekst verwijderen gaat door het te verwijderen deel met de muis grijs te maken (gebruik linker muisknop) en daarna een nieuwe tekst in te typen of op de backspace toets (← toets) te drukken.

#### 3.2.5.3 Het werken met blokken

Blokcommando's hebben betrekking op grotere hoeveelheden tekst. Deze maken het bijvoorbeeld mogelijk delen van een bestand te kopiëren, te verwijderen of te bewaren. Een blok moet eerst gemarkeerd worden voordat er mee gewerkt kan worden. Er kan met één blok te gelijk gewerkt worden.

Een blok wordt gemarkeerd door de cursor op de gewenste plaats te zetten. Druk vervolgens de linkermuisknop in en houd deze ingedrukt. Verplaats vervolgens de cursor naar het einde van het gewenste deel van de tekst. Laat nu de linkermuisknop los, het geselecteerde stuk tekst heeft een andere achtergrondkleur.

U kunt nu de volgende acties uitvoeren:

- verwijderen met het commando `cut` in het Edit-menu of met behulp van de backspace-toets kunt u het blok wissen.
- verplaatsen na het blok verwijderd te hebben kunt u het verplaatsen door de cursor op de gewenste positie te zetten en vervolgens met behulp van het commando `paste` in het Edit-menu de tekst weer op te roepen.
- kopiëren markeer het blok en kies de optie `copy` in het Edit-menu. De tekst blijft nu staan. Zet de cursor op de gewenste positie en met het commando `paste` wordt de tekst geplaatst.

## 4 BASISCURSUS

### 4.1 Matrices en vectoren. Bewerkingen met matrices en vectoren

Het basiselement van Matlab is een matrix. Speciale gevallen hiervan zijn een  $1 \times 1$ -matrix, een scalar, en een matrix bestaande uit slechts één rij of kolom, een vector. De grenzen van een matrix hoeven niet gedeclareerd te worden.

De naam van een variabele mag bestaan uit ten hoogste 19 letters of cijfers, en moet beginnen met een letter. Er wordt onderscheid gemaakt tussen grote en kleine letters.

Wanneer een variabele eenmaal ingevoerd is wordt hij automatisch in het interne geheugen bewaard totdat de Matlabsessie beëindigd wordt.

>> who laat zien welke variabelen in het geheugen zijn opgeslagen,

>> whos: geeft daarbij ook nog de omvang van de variabelen.

Het commando

>> clear <varlijst>: verwijdert de in <varlijst> genoemde variabelen uit het geheugen. Zonder deze lijst worden alle variabelen gewist.

Variabelen kunnen bewaard worden tot een volgende Matlabsessie met het commando save. Met:

>> save *filenaam* <varlijst>: worden de in varlijst opgenoemde variabelen opgeslagen in de file *filenaam.mat*. Zonder <varlijst> worden alle in het geheugen opgeslagen variabelen bewaard. Met

>> load *filenaam*: worden zij weer in Matlab ingelezen.

>> save hulp a b  
{wegschrijven van de variabelen a en b naar de file hulp.mat}

>> clear a b  
{wissen van de variabelen a en b uit het interne geheugen}

>> load hulp  
{inlezen van de variabelen onder de namen a en b}

Het invoeren van variabelen kan op verschillende manieren gebeuren. Ten eerste door opsomming van de afzonderlijke elementen. Hierbij worden rijelementen van elkaar gescheiden door een spatie of komma, kolomelementen door een puntkomma.

```
>> a = [1 2 3 ; 4, 5, 6; 7 8 9]
      {genereren van de matrix a}
a =
     1     2     3
     4     5     6
     7     8     9
```

Een rij kan worden voortgezet op de volgende regel door af te sluiten met drie of meer punten; na [RETURN] begint een volgende rij.

```
>> a = [ 1  2  ...
        3  4
        5  6  7  8]
        {genereren van de matrix a}
a =
    1  2  3  4
    5  6  7  8
```

Ten tweede kunnen variabelen gegenereerd worden met behulp van de dubbele punt :

variabele = beginwaarde : stapgrootte : eindwaarde

```
>> a = 0 : 2 : 10
        {genereren van de rij-vector a}
a =
    0  2  4  6  8  10
```

Voor het genereren van grote matrices staan de volgende functies ter beschikking :

zeros(n,m) - nxm matrix met nullen

ones(n,m) - nxm matrix met enen

eye(n) - nxn eenheidsmatrix

```
>> a = ones(3,3) + 2*eye(3)
        {genereren van de matrix a}
a =
    3  1  1
    1  3  1
    1  1  3
```

Matrices kunnen ook opgebouwd worden uit afzonderlijke variabelen.

```
>> k = ones(3,1); a = [-k 2*k -k]
        {genereren van de matrix a}
a =
   -1  2  -1
   -1  2  -1
   -1  2  -1
```

De puntkomma wordt gebruikt om statements van elkaar te scheiden en uitvoer naar het beeldscherm te onderdrukken.

Matrizelementen kunnen afzonderlijk of in groepen gebruikt worden :

$$\begin{aligned} a(i, j) &= a_{ij} \\ a(:, j) &= j^{\text{e}} \text{ kolom vana} \\ a(i, :) &= i^{\text{e}} \text{ rij vana} \\ v(i : j) &= \text{vectorelementen } v_i, v_{i+1}, \dots, v_j \end{aligned}$$

Combinaties hiervan zijn mogelijk:

$a(r_1 : r_2, k_1 : k_2)$  is het blok bestaande uit de elementen  $k_1$  tot en met  $k_2$  van de rijen  $r_1$  tot en met  $r_2$  van de matrix  $a$ .

Met matrices en vectoren kunnen de volgende bewerkingen gedaan worden :

- + optellen
- aftrekken
- \* vermenigvuldigen
- ' getransponeerde

Bij deze bewerkingen is het belangrijk dat de dimensies van variabelen kloppen, anders volgt onherroepelijk een foutmelding.

Met het accent teken kan bijvoorbeeld het inproduct van twee kolomvectoren worden berekend :

$$x' * y = \langle x, y \rangle$$

Met de `.` als voorvoegsel is het mogelijk om bepaalde bewerkingen elementsgewijs op variabelen te laten werken. Zo is het mogelijk een elementsgewijze vermenigvuldiging van twee kolomvectoren te maken :

$$x .* y = [x(1)y(1), \dots, x(n)y(n)]'$$

In Matlab hoeven grenzen van variabelen niet gedeclareerd te worden. Het kan echter nodig zijn deze te kennen.

`v = size(a)` geeft als antwoord  $v(1) = n$  en  $v(2) = m$  wanneer  $a$  een  $n \times m$  matrix is  
`l = length(v)` geeft het aantal elementen van de vector  $v$ .

Bij ijle matrices is het verstandig om van het sparse commando gebruik te maken. Dit kan veel geheugen en rekentijd besparen.

`a = sparse(a)` bouwt een sparse matrix op, waarbij alleen de niet nul elementen en hun indices opgeslagen worden.

## 4.2 Elementaire wiskundige functies

Enkele bekende wiskundige functies werken op de afzonderlijke elementen van variabelen.

abs - absolute waarde  
sqrt - wortel  
round - afronden op dichtstbijgelegen gehele getal  
rem - rest na deling  
sin - sinus  
cos - cosinus  
tan - tangens  
asin - arcsinus  
acos - arccosinus  
atan - arctangens  
exp - exponentieel grondtal  $e$   
log - natuurlijke logaritme

De constante  $\pi$  heet in Matlab pi.

Op vectoren zijn de volgende functies van toepassing

max - grootste element  
min - kleinste element  
sum - som van de elementen

```
>> max(abs(v))  
    {berekenen van het in absolute waarde grootste element van v}
```

### 4.3 Rationele en logische operatoren

Er zijn zes relationele operatoren beschikbaar voor het vergelijken van variabelen of uitdrukkingen.

< kleiner dan  
<= kleiner dan of gelijk  
> groter dan  
>= groter dan of gelijk  
== gelijk  
= niet gelijk

Daarnaast zijn er de volgende drie logische operatoren :

& en  
| of  
~ niet

### 4.4 Herhalingsopdrachten. Voorwaardelijke opdrachten.

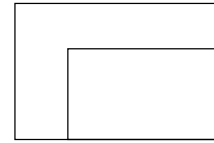
Net als de meeste programmeertalen kent Matlab ook herhalings- en voorwaardelijke opdrachten.



## FOR - opdracht

Syntax :

```
for variabele = beginwaarde : stapgrootte : eindwaarde,  
    opdrachten  
end
```



De stapgrootte kan zowel positief als negatief zijn. Indien niet gedefinieerd wordt hij op één gesteld.

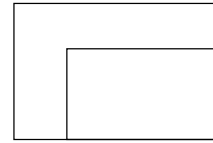
## WHILE - opdracht

Syntax:

```
while voorwaarde  
    opdrachten
```

end

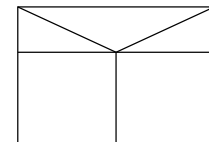
De voorwaarde moet een logische expressie zijn.



## IF - opdracht

Syntax voorwaardelijke opdracht :

```
if voorwaarde  
    opdrachten  
end
```

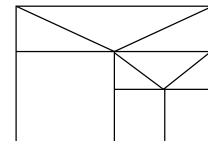


Syntax enkelvoudige keuzeopdracht :

```
if voorwaarde  
    opdrachten  
else  
    opdrachten  
end
```

Syntax meervoudige keuzeopdracht :

```
if voorwaarde  
    opdrachten  
elseif voorwaarde  
    opdrachten  
else  
    opdrachten  
end
```



De voorwaarde moet een logische expressie zijn.

## 4.5 Script-files. In- en uitvoer

Een script-file is een file bestaande uit een opeenvolging van Matlabcommando's : een Matlab-programma. Een dergelijke file wordt aangemaakt met behulp van Nedit. De standaard extensie voor deze files is .m .

Ieder Matlab-commando wordt afgesloten met [RETURN]. Dit geldt ook voor de laatste regel van een script-file.

#### Invoer naar script-file :

Variabelen kunnen op verschillende wijze van buiten het programma van een waarde worden voorzien. Het commando `input('tekst')` schrijft tekst op het beeldscherm en wacht dan op invoer vanaf het toetsenbord. Deze invoer kan daarna aan een variabele worden toegekend.

```
>> h = input('stapgrootte') {invoeren van stapgrootte en toekennen aan de variabele h}
```

Ook met behulp van `load` kan een variabele een waarde worden gegeven.

#### Uitvoer van script-file :

Uitvoer van een programma gaat gewoonlijk naar het beeldscherm. Matlab geeft allereerst iedere opdracht uit een programma weer en daarna de opeenvolgende resultaten ervan. Het echoën van de opdrachten wordt onderdrukt door `echo off`, het echoën van resultaten wordt onderdrukt door commando's af te sluiten met een puntkomma.

Het weergeven van variabelen geschiedt simpelweg door het intoetsen van de naam ervan. Variabelen worden standaard in 15 cijfers plus exponent afgedrukt. Met het commando `format` kan deze instelling verandert worden.

`disp(varnaam)` geeft alleen de waarde van de variabele weer. `disp('tekst')` schrijft *tekst* op het scherm.

Uitvoer van een programma kan worden afgedrukt, nadat zij in een file is gezet. Na

```
>> diary filenaam
```

wordt alles wat op de commandoregel achter de `>>` wordt ingetoetst, en alle resultaten daarvan die op het beeldscherm verschijnen, in de file *filenaam* gezet. Invoer vanaf het toetsenbord voor de functie `input` wordt **niet** in de file gezet.

De file wordt afgesloten met

```
>> diary off
```

De aldus gevormde uitvoer-file kan daarna worden afgedrukt.

```
>> diary progheun.uit {open progheun.uit als uitvoerfile}
```

```
>> progheun {start het programma progheun.m (zie & 2.9)}  
    stapgrootte h = 0.01 {geef h vanaf het toetsenbord}  
    uitvoer {uitvoer van het programma}
```

```
>> diary off {sluit de uitvoerfile progheun.uit}
```

Nota Bene De vetgedrukte tekst wordt in de uitvoerfile progheun.uit gezet.

Met het commando `pause` kan een programma worden onderbroken, bijvoorbeeld om de uitvoer te bekijken. Na het indrukken van een willekeurige toets gaat het programma verder.

Met `clc` wordt het tekstwindow schoongemaakt.

## 4.6 Function-files

Door middel van function-files kunnen er nieuwe functies aan Matlab worden toegevoegd. Een function-file bevat net als een script-file een opvolging van commando's, maar met behulp van in- en uitvoervariabelen is het mogelijk met andere files te communiceren.

De filenaam van een function-file, bestaande uit ten hoogste acht letters en met de standaard extensie `.m`, is de naam waarmee de functie vanuit een andere file wordt aangeropen.

**Het gebruik van functionfiles voor functievoorschriften is verplicht.**

Globaal heeft een function-file de volgende vorm:

```
function uitvoervariabele = funktienaam(invoervariabelen)
    opdrachten
end
```

De eerste regel maakt het onderscheid met een script-file duidelijk door middel van het woord `function`. **Het is verplicht voor *funktienaam* de filenaam te gebruiken.**

De uitvoervariabele kan een vector zijn die opgebouwd is uit meerdere elementen. Een functie moet afgesloten worden met `end`.

De variabelen in een function-file zijn lokaal, en worden dus niet in het Matlab-geheugen bewaard.

Ter verduidelijking een voorbeeld. De functionfile `norm2.m` :

```
function y = norm2(x)
% norm2
% Deze functie berekent de Euclidische- of  $L_2$ -norm van een
% vector  $x$ 
    y = sqrt(sum(abs(x).^ 2));
% Kwadrateer de absolute waarde van de elementen van  $x$ , sommeer
% de elementen, en neem de wortel van de som.
end
```

## 4.7 Plotten

Voor het maken van tekeningen staan in Matlab de volgende commando's ter beschikking:

```
plot
hold
shg
clg
axis
```

Het basiscommando `plot` wordt gebruikt voor het tekenen van lijnen en punten. Na ieder plot-commando wordt het grafische scherm schoongemaakt. De eenvoudigste vorm is

```
>> plot(y)
```

Hierbij wordt de vector `y` uitgezet tegen zijn indices. Wanneer de vectoren `x` en `y` dezelfde lengte hebben zet

```
>> plot(x,y)
```

de vector `y` uit tegen `x`. Het tekenen van een lijnstuk geschiedt dus met

```
plot([xbegin xeind],[ybegin yeind])
```

```
t=0:0.1:2*pi; y = sin(t); plot(t,y)   tekenen van de sinusfunctie tussen 0 en 2π
```

Matlab dimensioneert het grafische scherm automatisch aan de hand van de `x`- en `y`-waarden. Het is mogelijk zelf het assenstelsel in te stellen met

```
axis([xmin xmax ymin ymax])
```

Het plotten van een speciaal teken in `(xpunt,ypunt)` gaat met

```
plot(xpunt,ypunt,'teken')
```

waarbij *teken* één van de volgende tekens is: `.` `+` `*` `o` (kleine letter o) `x` (kleine letter x)

Het commando `hold on` zorgt er voor dat nieuwe plotopdrachten tekenen over de oude heen komen. Het commando moet dus gegeven worden **na** de eerste plotopdracht. Na `hold off` wordt bij ieder plotcommando het window schoongemaakt.

Voor de opmaak van tekeningen kan, **na** de plot-opdrachten gegeven te hebben, gebruik gemaakt worden van :

<code>title('tekst')</code>	- schrijft <i>tekst</i> boven de plot
<code>xlabel('tekst')</code>	- schrijft <i>tekst</i> onder de <i>x</i> -as
<code>ylabel('tekst')</code>	- schrijft <i>tekst</i> naast de <i>y</i> -as
<code>text(x,y,'tekst')</code>	- schrijft <i>tekst</i> in het punt $(x,y)$ van de tekening
<code>numtext(x,y,var)</code>	- schrijft de waarde van <i>var</i> in het punt $(x,y)$ van de tekening
<code>grid</code>	- voorziet de grafiek van een raster
<code>gtext('tekst')</code>	- zet met behulp van de muis <i>tekst</i> op een willkeurige plaats

Het afdrucken van een tekening op de printer kan met het commando `print`. Na enige tijd zal er een afdruk van uw tekening uit de printer verschijnen.

## 4.8 Het oplossen van een stelsel vergelijkingen

Voor het oplossen van het stelsel vergelijkingen  $A\underline{x} = \underline{b}$  kan er gekozen worden uit twee mogelijkheden. Voor beide geldt i dat het commando `a = sparse(a)` veel rekestijd kan besparen als  $A$  een ijle matrix is.

i) Als er één stelsel opgelost moet worde

```
>> x = a \ b ;
```

ii) Als er meerdere stelsels opgelost moeten worden, waarbij de matrix  $A$  hetzelfde is, maar het rechterlid steeds verschillend is. Bepaal eerst de LU decompositie van  $A$ :

```
>> [l,u] = lu(a) ;
```

Los daarna voor elk rechterlid een beneden en boven driehoeksstelsel op

```
>> y = l \ b ;
```

```
>> x = u \ y ;
```

## 4.9 Voorbeeldprogramma

```
% Dit programma berekent de oplossing  $Q(x)$  van de differentiaalvergelijking
%  $Q' = 1 - Q^2$ ,  $Q(0) = 0$ 
% op het interval [0,5] met behulp van de methode van Heun.

% Initialiserende commando's:

% geen commando's op het scherm tonen
echo off;

% uitvoer tot 15 cijfers achter de komma
format long;
format compact;

% Programma

% voer stapgrotte in en bereken het aantal stappen voor de integratie
h = input('stapgrotte h =');
aantstap = round(5 \ h);

% beginvoorwaarde
x = 0;
Q0 = 0;

% creëer twee vectoren voor de grafiek
xgrafiek=[0];
Qgrafiek=[0];

% herhalingsopdracht voor de methode van Heun
% funk is een function-file welke de functiewaarden voor f
% berekent in het punt (x,y)
for i = 1:aantstap,
    Qx = Q0 + h * funk(x, Q0);
    Qx = Q0 + (h/2) * (funk(x, Q0) + funk((x+h), Qx));
    x = x + h;
    xgrafiek=[xgrafiek x];
    Qgrafiek=[Qgrafiek Qx];
    Q0 = Qx;
end;

% Afdrukken van de resultaten
% afdrukken van de kop van de tabel
disp(' x Q(x) ');
disp(' 0 0 ');
For i = 1:(aantstap -5),
    if rem(i, 10) == 0,
        disp([ixgrafiek(i) Qgrafiek(i)]);
    end;
end;
for i = (aantstap-5):aantstap,
    disp([ixgrafiek(i) Qgrafiek(i)]);
end;

pause(5);

% grafiek van de berekende oplossing
axis([0 5 0 1]);
plot(xgrafiek,Qgrafiek);

text(3,0.5,'naam : verplicht !');
title('Afvoer uit waterreservoir');
```

De function-file `funk.m` ziet er als volgt uit:

```
function qacc=funk(x,y)
% Function-file voor het berekenen van de funktiewaarden van
%  $f(x,y) = 1 - y^2$ 
qacc=1 - y^ 2;
end;
```

Het gebruik van commentaar en verklarende variabelenamen zorgt voor snelle assistentie bij eventuele moeilijkheden.



## **5 REFERENTIE**

In het referentiedeelte geldt de volgende notatie:

		axis	ax
			ve
			me
			ax
			vic
<b>n,m</b>	- scalar		
<b>A</b>	- matrix		
<b>v,w</b>	- vector		
<b>x,y</b>	- willekeurig		
[ ]	worden gebruikt bij het creëren van matrices en vectoren	clc	cl
( )	worden gebruikt om :	clear	cl
	- volgorde aan te geven in opdrachten		ve
	- argumenten van functies te omsluiten	clg	cl
	- indices van vectoren en matrices te omsluiten		
.	decimale punt. Drie of meer punten aan het einde van een regel geven aan dat deze op de volgende regel wordt voortgezet	clock	cl
,	scheidingsteken van matrix en vectorelementen. Ook scheidingsteken tussen verschillende opdrachten op één commando-regel	cos	co
;	scheidingsteken tussen verschillende rijen in een matrix. Kan in plaats van de komma als scheidingsteken worden gebruikt om weergave van berekeningen op het scherm te onderdrukken	demo	de
%	alle tekst op een regel na % wordt opgevat als commentaar	diary	di
!	wordt gebruikt voor Unix-commando's die betrekking hebben op files	disp	di
:	wordt gebruikt voor het genereren van variabelen, in for-herhalingsopdrachten en voor het selecteren van matriceelementen : $A(:,n)$ is de $n^{\text{de}}$ kolom van $A$ , $A(m,:)$ de $m^{\text{de}}$ rij	eig	ei
'	transponeert matrices en vectoren	echo	ec
.*	$v.*w$ : elementsgewijze vermenigvuldiging van vectoren	else	zie
abs	$\text{abs}(x)$ is de absolute waarde van (de elementen van) $x$	elseif	zie
ans	variabele welke automatisch de waarde krijgt van een expressie als er geen toekenning wordt gedaan	end	en
acos	$\text{acos}(x)$ is de arccosinus van (de elementen van) $x$	eps	flo
asin	$\text{asin}(x)$ is de arcsinus van (de elementen van) $x$	etime	et
atan	$\text{atan}(x)$ is de arctangens van (de elementen van) $x$		to
		exp	pro
		eye	ex
			ey
			en

for            herhalingsopdracht :  
for    beginwaarde : stapgrootte : eindwaarde,  
         opdrachten  
         end

format        formateren van uitvoer:  
format short : 5 cijfers, vaste komma  
format short e : 5 cijfers, drijvende komma  
format long : 15 cijfers, vaste komma  
format long e : 15 cijfers, drijvende komma  
Standaard instelling is long e  
format compact onderdrukt extra lege regels in uitvoer  
format loose voegt ze toe

full          full(a) converteert een ijle matrix naar een volle matrix.

function     door de gebruiker te creeren functie :  
function   uitvoervar = *funktienaam* (invoervar)  
             opdrachten  
             end  
Function-files moeten de filenaam *funktienaam.m* hebben, waarbij  
*funktienaam* uit ten hoogste 8 letters mag bestaan

grid          voorziet een grafiek van een raster van lijnen

gtext        gtext('tekst') plaatst met behulp van de muis *tekst* in een tekening

help          help toont de functies waarover informatie kan worden gekregen.  
help *funktienaam* toont deze informatie op het scherm

hold          hold on houdt de laatste plot op het scherm en schrijft volgende plot-  
opdrachten eroverheen. hold off schakelt de functie weer uit

if            voorwaardelijke opdrachten :  
if    voorwaarde            if    voorwaarde  
         opdrachten            opdrachten  
else                            elseif    voorwaarde  
         opdrachten                            opdrachten  
end                                else  
    opdrachten  
    end

inf            inf is een variabele welke de machinerepresentatie bevat voor oneindig.  
Wordt onder andere gebruikt als resultaat van de deling 1/0

input	<code>input('tekst')</code> geeft <i>tekst</i> weer op het scherm en wacht op invoer van de gebruiker. Kan worden toegekend aan een variabele
length	<code>length(v)</code> geeft het aantal elementen van de vector <i>v</i>
load	<code>load filenaam</code> laadt de variabelen uit de file <i>filenaam</i> in het geheugen. De file heeft het filetype <code>.mat</code>
log	<code>log(x)</code> is de natuurlijke logaritme van (de elementen van) <i>x</i>
max	<code>max(v)</code> is het grootste element van de vector <i>v</i>
min	<code>min(v)</code> is het kleinste element van de vector <i>v</i>
norm	<code>norm(A)</code> bepaalt de norm van <i>A</i>
numtext	<code>numtext(m,n,var)</code> schrijft de waarde van <i>var</i> in het punt $(m,n)$ van het grafische scherm
ones	<code>ones(n)</code> is een $n \times n$ matrix gevuld met enen, <code>ones(m,n)</code> een $m \times n$ matrix
pack	<code>pack</code> zet de in het geheugen aanwezige variabelen zo efficiënt mogelijk neer
pause	<code>pause</code> stopt het lopende programma en wacht op het indrukken van een willekeurige toets. <code>pause(n)</code> stopt gedurende <i>n</i> seconden
pi	<code>pi</code> is de machine-representatie voor $\pi$
plot	Tekenen van grafieken: <code>plot(v)</code> : zet de vector <i>v</i> uit tegen zijn indices <code>plot(v,w)</code> : zet de vector <i>v</i> uit tegen de vector <i>w</i> <code>plot(m,n,'tekening')</code> : plaatst een tekening in het punt $(m,n)$ . Hierbij kan gekozen worden uit <code>.</code> <code>+</code> <code>*</code> <code>o</code> en <code>x</code>
print	Met behulp van <code>print</code> kunt u uw tekeningen afdrukken op de printer.
quit	Uitloggen vanuit Matlab
rem	<code>rem(m,n)</code> is de rest na heling van <i>m</i> door <i>n</i>
round	<code>round(x)</code> rondt de elementen van <i>x</i> af op de dichtstbijgelegen integers
save	<code>save filenaam x{y}</code> schrijft de variabele <i>x</i> en <i>y</i> weg naar de file <i>filenaam.mat</i>
script	Een script-file is een file bestaande uit een opeenvolging van Matlab-commando's. Na aanroep van de filenaam worden deze uitgevoerd

shg	shg toont het laatste grafische scherm
sin	$\sin(x)$ is de sinus van (de elementen van) $x$
size	$[m, n] = \text{size}(x)$ geeft de dimensies van de variabele $x$ als dit een $m \times n$ matrix is
sparse	sparse(a) transformeert een matrix zodanig, dat alleen de niet-nullen en hun indices opgeslagen worden.
sqrt	sqrt(x) is de vierkantswortel uit (de elementen van) $x$
sum	sum(v) is de som van de elementen van de vector $v$
tan	$\tan(x)$ is de tangens van (de elementen van) $x$
text	text( $m, n, 'tekst'$ ) schrijft <i>tekst</i> in het punt ( $m, n$ ) van het grafische scherm
title	title('tekst') schrijft <i>tekst</i> als titel boven het grafische scherm
type	typefilenaam toont de inhoud van de file <i>filenaam</i> op het commando-scherm. Geldt alleen voor files met de extensie .m
while	voorwaardelijke herhalingsopdracht : while voorwaarde opdrachten end
who	who laat zien welke variabelen in het Matlab-geheugen zitten
whos	whos laat naam en omvang zien van de variabelen die in het Matlab-geheugen zitten
xlabel	xlabel('tekst') schrijft <i>tekst</i> onder de $x$ -as van het grafische scherm
ylabel	ylabel('tekst ') schrijft <i>tekst</i> naast de $y$ -as van het grafische scherm
zeros	zeros( $n$ ) creert een $n \times n$ matrix met nullen, zeros( $m, n$ ) een $m \times n$ matrix

## INDEX

.* .....	15	load .....	13
:	14	log .....	15
;	13	Logische operatoren .....	16
'	15	max .....	16
+	15	min .....	16
-	15	Niet .....	16
*	15	Of .....	16
<	16	ones .....	14
<=	16	pack .....	13
==	16	pi .....	15
>	16	plot .....	19
&	16	print .....	20
	16	Relationele operatoren .....	16
	16	rem .....	15
=	16	round .....	15
abs .....	15	save .....	13
acos .....	15	Script-file .....	9, 17
asin .....	15	shg .....	19
atan .....	15	sin .....	15
axis .....	19	size .....	15
clc .....	18	sqrt .....	15
clear .....	13	sum .....	16
clg .....	19	tan .....	15
cos .....	15	text .....	20
Cursortoetsen .....	9	title .....	20
Demonstratieprogramma .....	9	while .....	17
diary .....	18	who .....	13
disp .....	18	whos .....	13
echo .....	18	xlabel .....	20
elseif .....	17	ylabel .....	20
En .....	16	zeros .....	14
Enter .....	9		
exp .....	15		
eye .....	14		
for .....	16		
function .....	19		
Function-file .....	9, 19		
hold .....	19		
if .....	17		
input .....	18		
length .....	15		