

# Een sessie met Matlab

Serge Goossens

14 oktober 1996

## 1 Opstarten

Dit begint uiteraard met het aanzetten van de PC. Normaal verschijnt dan het Windows hoofdmenu. Daaruit kan je rechtstreeks Matlab selecteren door tweemaal snel te klikken met de muis op het ‘ikoon-tje’ van Matlab. Je krijgt dan een nieuw scherm met de Matlab prompt.

```
>>
```

## 2 Elementaire Matlab

Matlab heeft een help commando:

```
>> help
```

Dit commando kan je helpen tijdens een Matlab-sessie.

Je kan ook hulp opvragen bij een specifiek commando

```
>> help exp
```

Toekenningen van variabelen:  
de matrix a

```
>> a = [1 2 3 ; 4 5 6 ; 7 8 9 ]
```

```
a =
```

```
1     2     3
4     5     6
7     8     9
```

de vector x:

```
>> x = [ -1/3 4.5 pi ]
```

```
x =
```

```
-0.3333    4.5000    3.1416
```

De uitvoer komt op het scherm in verschillende formaten:

```
>> format long
```

```
>> x
```

```
x =
```

```
-0.333333333333333    4.500000000000000    3.14159265358979
```

```
>> format short e
```

```
>> x
```

```
x =
```

```
-3.3333e-01    4.5000e+00    3.1416e+00
```

```
>> format short
```

```
>> x
```

```
x =
```

```
-0.3333    4.5000    3.1416
```

Het commando

```
>> help format
```

geeft nog meer mogelijkheden.

Indien je een ; na een commando intikt, wordt het resultaat niet getoond.

```
>> x;
```

```
>>
```

Indien je geen toekenning gebruikt, krijgt de variabele **ans** het resultaat.

```
>> [1 2 3]
```

```
ans =
```

```
1     2     3
```

## 3 Enkele bewerkingen op matrices en vectoren

Een ' betekent een transpose.

```
>> x'
```

```
ans =
```

```
-0.3333
4.5000
3.1416
```

Vermenigvuldigen met een getal:

```
>> y = 2*x;
>> y
```

```
y =
    -0.6667    9.0000    6.2832
```

Bewerkingen met matrices en vectoren of elementen ervan:

```
>> a
```

```
a =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> a(3,3) = a(1,3)+a(3,1)
```

```
a =
```

```
    1    2    3
    4    5    6
    7    8   10
```

## 4 Bewerkingen op matrices

De inverse matrix:

```
>> inv(a)
```

```
ans =
```

```
   -0.6667   -1.3333    1.0000
   -0.6667    3.6667   -2.0000
    1.0000   -2.0000    1.0000
```

```
>> x
```

```
x =
```

```
   -0.3333    4.5000    3.1416
```

```
>> z = x + [ 10 10 10 ]
```

```
z =
```

```
    9.6667   14.5000   13.1416
```

$x$  is een rijvector van lengte 3.

```
>> a*x'
```

```
ans =
```

```
   18.0914
   40.0162
   65.0826
```

```
>> b = x'
```

```
b =
```

```
   -0.3333
    4.5000
    3.1416
```

Stelsel oplossen: gegeven:  $a$  en  $b$ , gevraagd:  $x$  zodat  $ax = b$ .

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

```
x =
```

```
   -2.6362
   10.4390
   -6.1917
```

```
>> a*x - b
```

```
ans =
```

```
   1.0e-014 *
    0.0278
    0.1776
    0.4441
```

De 2-norm van  $a$ :

```
>> norm(a)
```

```
ans =
```

```
   17.4125
```

De oneindignorm van  $a$ :

```
>> norm(a,'inf')
```

```
ans =
```

```
    25
```

Het conditiegetal:

```
>> cond(a)
```

```
ans =
```

```
   88.4483
```

```
>> norm(a)*norm(inv(a))
```

```
ans =
```

```
   88.4483
```

## 5 Genereren van matrices

Genereren van een hilbertmatrix:

```
>> h = hilb(3)

h =

    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
```

Genereren van een inverse hilbertmatrix:

```
>> g = invhilb(3)

g =

     9    -36     30
   -36    192   -180
     30   -180    180
```

```
>> h*g

ans =

    1.0000    0.0000   -0.0000
    0.0000    1.0000    0.0000
    0.0000   -0.0000    1.0000
```

Genereren van een eenheidsmatrix:

```
>> e = eye(2)

e =

     1     0
     0     1
```

Genereren van een  $3 \times 2$  matrix met eentjes op de diagonaal:

```
>> e = eye(3,2)

e =

     1     0
     0     1
     0     0
```

Genereren van matrices met willekeurige elementen:

```
>> a = rand(3)

a =

    0.2113    0.0087    0.4524
    0.0824    0.8096    0.8075
    0.7599    0.8474    0.4832
```

```
>> b = rand(3,1)
```

```
b =

    0.6135
    0.2749
    0.8807
```

Orthogonalisatie van de kolommen van een matrix met willekeurige elementen:

```
>> u = orth(rand(3))

u =

    0.5956   -0.6540   -0.4665
    0.6146    0.7449   -0.2596
    0.5173   -0.1321    0.8456

>> u' * u - eye(3)

ans =

    1.0e-015 *

   -0.1110   -0.0022    0.0384
   -0.0022         0   -0.0169
    0.0384   -0.0169   -0.1110
```

Allemaal eentjes:

```
>> ones(2,3)

ans =

     1     1     1
     1     1     1
```

Allemaal nullen:

```
>> zeros(2,3)

ans =

     0     0     0
     0     0     0
```

Genereren van een diagonaalmatrix:

```
>> sing = [ 0.0001 0.02 1.0 ]

sing =

    0.0001    0.0200    1.0000

>> d = diag(sing)

d =

    0.0001         0         0
         0    0.0200         0
         0         0    1.0000
```

Een matrix maken met gekend conditiegetal:

```
>> v = orth(rand(3));
>> a = u*d*v'

a =
```

```

-0.3190    0.2569   -0.2237
-0.1930    0.1342   -0.1112
 0.5967   -0.4550    0.3898

```

```
>> norm(a)
```

```
ans =
```

```
1.0000
```

```
>> cond(a)
```

```
ans =
```

```
1.0000e+004
```

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

```
x =
```

```
1.0e+003 *
```

```
1.3844
```

```
7.3592
```

```
6.4739
```

Perturbaties aanbrengen op een matrix:

```
>> ap = a + rand(3)*norm(a)*1.e-10
```

```
ap =
```

```
-0.3190    0.2569   -0.2237
```

```
-0.1930    0.1342   -0.1112
```

```
 0.5967   -0.4550    0.3898
```

```
>> xp = ap\b
```

```
xp =
```

```
1.0e+003 *
```

```
1.3844
```

```
7.3592
```

```
6.4739
```

```
>> norm(x-xp)/norm(x)
```

```
ans =
```

```
1.1920e-006
```

## 6 De hoeveelheid rekenwerk bepalen

Matlab telt automatisch hoeveel floating-point bewerkingen uitgevoerd zijn tijdens de Matlab-sessie. Dit aantal wordt opgevraagd door de instructie

`flops`. De teller kan op 0 gezet worden door de instructie `flops(0)`.

Een schatting voor  $1/\text{cond}(a)$  wordt berekend door het commando `rcond(a)`. Dit commando vraagt veel minder rekenwerk dan `cond(a)`.

`rcond(a)` is een schatting voor  $1/\text{cond}(a)$ .

```
>> flops(0)
```

```
>> cond(a)
```

```
ans =
```

```
8.579580886410574e+16
```

```
>> flops
```

```
ans =
```

```
203
```

```
>> flops(0)
```

```
>> 1/rcond(a)
```

```
ans =
```

```
4.863887597560134e+17
```

```
>> flops
```

```
ans =
```

```
71
```

## 7 Manipuleren van matrices

Samenvoegen van matrices:

```
>> a = [ hilb(3) eye(3,2) ; 1 2 3 4 5 ]
```

```
a =
```

```
1.0000    0.5000    0.3333    1.0000         0
```

```
0.5000    0.3333    0.2500         0    1.0000
```

```
0.3333    0.2500    0.2000         0         0
```

```
1.0000    2.0000    3.0000    4.0000    5.0000
```

Selecteren van rijen en kolommen:

```
>> a(2,:) 
```

```
ans =
```

```
0.5000    0.3333    0.2500         0    1.0000
```

```
>> a(:,4)
```

```
ans =
```

```
1
```

```
0
```

```
0
4
```

Verwisselen van kolommen:

```
>> a( : , [ 1 4 ]) = a( : , [ 4 1 ])
```

```
a =
```

```
1.0000    0.5000    0.3333    1.0000         0
         0    0.3333    0.2500    0.5000    1.0000
         0    0.2500    0.2000    0.3333         0
4.0000    2.0000    3.0000    1.0000    5.0000
```

Gebruik van lussen:

```
>> for i= 1:3,
    b(i) = i^3
end
```

```
b =
```

```
1
```

```
b =
```

```
1      8
```

```
b =
```

```
1      8     27
```

Vermits na het bevel in de **for**-lus geen ‘;’ stond,  
werd **b** telkens gedrukt.

```
>> for i = 1:3,
    b(i) = i^3;
end
>> b
```

```
b =
```

```
1      8     27
```

Je kan ook gebruik maken van **while**:

```
>> i=10;
>> while(i>0),
a(i)=i;
i=i-1;
end
```

## 8 Berekening van een Kleinste kwadratenbenadering

Gegeven:  $a \in \mathbb{R}^{m \times n}$ ,  $d \in \mathbb{R}^m$  en gevraagd:  $x \in \mathbb{R}^n$ .

```
>> a = rand(6,4); b = rand(6,1);
>> x = a\b;
>> res = b - a*x;
>> norm(res)

ans =

    0.5463

>> res'*a

ans =

    1.0e-015 *
   -0.1101   -0.1570   -0.2668   -0.1648
```

Het residu staat duidelijk orthogonaal op de kolom-ruimte van  $a$ .

## 9 Werken met bestanden

Wil je een afdruk van (het vervolg van) je sessie op het bestand met naam `filename`, druk dan `diary filename`. Bemerkt dat wat je op het scherm krijgt verloren gaat. De enige manier om dit bij te houden, is dit commando te gebruiken. Het commando `save filename` bewaart al je variabelen in het bestand `filename.mat`. Deze kan je in een later terug opladen door het commando `load filename`.

Het commando `who` geeft alle bestaande variabelen. Met `clear` vernietigt je alle variabelen.

```
>> a = rand(50); b = rand(10);
>> who
Your variables are:  eps, pi, Inf, NaN, ...
a                  b
>> save data
>> clear
>> who
Your variables are:  eps, pi, Inf, NaN, ...

>> load data
>> who
Your variables are:  eps, pi, Inf, NaN, ...
a                  b
>> clear a
>> who
Your variables are:  eps, pi, Inf, NaN, ...
b
```

## 10 Programmeren in Matlab

Soms kan het handig zijn een programma in Matlab te maken zodat je niet iedere keer alle commando's opnieuw moet intikken. Dit kan je doen met de zogenaamde *m-files*. Dit zijn bestanden van het formaat *\*.m* die een aantal Matlab commando's bevatten. Onder Matlab kan je de editor oproepen om zo een bestand aan te maken.

Een voorbeeld.

We hebben het bestand *vb.m* aangemaakt met de volgende inhoud.

```
% een voorbeeld voor m-file
%
a = diag([1:n]); % n is op voorhand ingegeven
p = rand(n);
a = p*a*inv(p);
eig(a)
```

Alle karakters achter een % worden geïnterpreteerd als commentaar. Wat doet dit programmaatje eigenlijk? Indien we eerste de variabele `n` definiëren en daarna de naam van het bestand intikken, wordt het programma uitgevoerd.

```
>> n=5;
>> vb

ans =
```

```
    1.0000
    4.0000
    3.0000
    5.0000
    2.0000
```

## 11 Functies in Matlab

Matlab beschikt over een aantal ingebouwde functies, waaronder de meeste elementaire wiskundige functies, zoals bv. de cosinusfunctie.

```
>> for i=1:11,
a(i)=(i-1)*pi/10;
end
>> a
```

a =

Columns 1 through 7

0 0.3142 0.6283 0.9425 1.2566 1.5708 1.8849

Columns 8 through 11

2.1991 2.5133 2.8274 3.1416

```
>> cos(a)
```

Columns 1 through 7

Columns 8 through 11

7

```

    x = x + alpha * p;                % update approximation vector

    r = r - alpha*q;                  % compute residual
    error = norm( r ) / bnorm2;        % check convergence
    if ( error <= tol ), break, end

    rho_1 = rho;

end

if ( error > tol ) flag = 1; end      % no convergence

% END cg.m

```

De lijnen commentaar (dit zijn de lijnen die beginnen met %) verschijnen wanneer je uitleg vraagt over de functie met het help commando (het is dus nuttig je functies op deze manier te documenteren).

```
>> help cg
```

```

[x, error, iter, flag] = cg(A, x, b, M, max_it, tol)

cg.m solves the symmetric positive definite linear system Ax=b
using the Conjugate Gradient method with preconditioning.

input   A          REAL symmetric positive definite matrix
        x          REAL initial guess vector
        b          REAL right hand side vector
        M          REAL preconditioner matrix
        max_it     INTEGER maximum number of iterations
        tol        REAL error tolerance

output  x          REAL solution vector
        error      REAL error norm
        iter       INTEGER number of iterations performed
        flag       INTEGER: 0 = solution found to tolerance
                   1 = no convergence given max_it

```

Zoals aangegeven in de hoofding van de functie

```
function [x, error, iter, flag] = cg(A, x, b, M, max_it, tol)
```

zijn de invoerparameters A, x, b, M, max\_it en tol en de uitvoer bestaat uit x, error, iter, flag. Een voorbeeld:

```
>> A=makefish(3)
```

```
A =
```

```

    4    -1     0    -1     0     0     0     0     0
   -1     4    -1     0    -1     0     0     0     0
    0    -1     4     0     0    -1     0     0     0
   -1     0     0     4    -1     0    -1     0     0
    0    -1     0    -1     4    -1     0    -1     0
    0     0    -1     0    -1     4     0     0    -1
    0     0     0    -1     0     0     4    -1     0
    0     0     0     0    -1     0    -1     4    -1
    0     0     0     0     0    -1     0    -1     4

```



```

>> exact=rand(9,1)

exact =

    0.95441490269937
    0.85126966836456
    0.28931620311426
    0.53742574133790
    0.51443466614673
    0.10343392803494
    0.41402848317010
    0.57671663983572
    0.87656571896587

>> b=A*exact

b =

    2.42896420109503
    1.64691290149787
    0.20256121605754
    0.26682491333542
   -0.01110731298621
   -1.26658087608711
    0.54197155150677
    0.50183769106019
    2.82611230799282

>> x=zeros(9,1)

x =

    0
    0
    0
    0
    0
    0
    0
    0
    0
    0

>> M = eye(9)

M =

    1     0     0     0     0     0     0     0     0
    0     1     0     0     0     0     0     0     0
    0     0     1     0     0     0     0     0     0
    0     0     0     1     0     0     0     0     0
    0     0     0     0     1     0     0     0     0
    0     0     0     0     0     1     0     0     0
    0     0     0     0     0     0     1     0     0
    0     0     0     0     0     0     0     1     0
    0     0     0     0     0     0     0     0     1

>> [x, error, iter, flag] = cg(A, x, b, M, 9, 1.0e-15)

```

```

x =

    0.95441490269937
    0.85126966836456
    0.28931620311426
    0.53742574133790
    0.51443466614673
    0.10343392803494
    0.41402848317010
    0.57671663983572
    0.87656571896587

error =

    6.012734133162594e-17

```

```

iter =

```

```

    5

```

```

flag =

```

```

    0

```

```

>> x-exact

```

```

ans =

```

```

    1.0e-15 *

```

```

    0.11102230246252
    0.33306690738755
    0.11102230246252
    0.22204460492503
    0.11102230246252
    0.15265566588596
    0.16653345369377
    0.22204460492503
    0.11102230246252

```

```

>> A\b-x

```

```

ans =

```

```

    1.0e-15 *

```

```

   -0.22204460492503
   -0.44408920985006
   -0.11102230246252
   -0.33306690738755
   -0.11102230246252
   -0.11102230246252
   -0.11102230246252
   -0.22204460492503
    0.11102230246252

```

```

>> A\b-exact

```

```

ans =

```

```

    1.0e-15 *

```

```

   -0.11102230246252
   -0.11102230246252
    0
   -0.11102230246252
    0
    0.04163336342344
    0.05551115123126
    0

```

```

    0.22204460492503

```

## 12 Stoppen

Met het commando:

```

>> quit

```

verlaat je het programma. Alle gegevens behalve de bestanden gaan verloren.