MATLAB Lecture 1. Introduction to MATLAB

1.1 The MATLAB environment

MATLAB is a software program that allows you to compute *interactively* with matrices. If you want to know for instance the product of a matrix a with a matrix b, then you just type in $a * b$ and the result appears immediately on screen. MATLAB has a wide range of functions to compute the determinant, inverse, LU-decomposition, eigenvalues, etc.

While working with MATLAB you get on your screen always a *prompt*. This is >>. This means that MATLAB is ready to accept commands. If you press the **enter** key, the command gets executed.

Unless you finish your command with a semicolon (;), MATLAB displays the result on screen. To copy the output also to a file, you can invoke the command **diary**. For example,

```
>> diary('h:\session.txt')
```

will cause MATLAB to copy all user input and output to the file `h:\session.txt`. The **diary off** command suspends the output to the file, whereas **diary on** turns it back on.

The command **who** displays the list all variables currently in use. With **clear** you deallocate memory space for those variables. If you want to use some or all of the variables with their values in a later MATLAB session you can save the workspace variables with the command **save('h:\myvars')** which creates the binary MAT-file named **myvars.mat**. With **load('h:\myvars')**, the workspace variables are loaded from the file **myvars.mat**.

Make sure you understand the difference between the file produced by diary and the file made by save! With diary we create a human readable text file, with save we make a binary mat-file.

MATLAB records the commands you typed in. You can scroll through the list of executed commands with the arrow keys ↑ and ↓. The arrows ← and → enable to go to the location in the command where edits are necessary. This is convenient to correct type errors in a command, or if you need to execute lots of slightly different versions of the same instruction.

MATLAB has online help facilities. The command **help** displays information in the current window, **helpwin** opens a new window, **helpdesk** launches your internet browser.

1.2 A short introduction to MATLAB

MATLAB follows a C like syntax. For example, the equality symbol '=' is used for assignment while '==' must be used to test equality. The matrix is the fundamental data structure in MATLAB.

1.2.1 Entering variables

NUMBERS < name-of-the-variable > = < value >

e.g.:

```
>> x = 0.65
```

makes that MATLAB assigns the value 0.65 to the variable `x`.
MATLAB confirms :

```
x =
     0.6500
```

To suppress this confirmation, end a command with ; (a semicolon).

MATRICES These are defined row-wise. A square bracket [means to MATLAB that you wish to enter a matrix. The input of a matrix is terminated by a]. You can enter a matrix in two ways.

For instance, you want

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

then you either type

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

or

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

In both cases, MATLAB answers :

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

1.2.2 The output

The content of a variable can be gotten just by typing in the name of the variable, followed by pressing the **enter** key. For example,

```
>> x
```

MATLAB answers:

```
x =
      0.6500
```

This is the standard output format of MATLAB. To obtain the scientific notation, you switch formats as follows

```
>> format short e
```

After typing

```
>> x
```

MATLAB answers then with:

```
x =
      6.500e-001
```

If you replace in the **format**–commands **short** by **long**, you see more decimal places.
To return to the standard output format, you type

```
>> format short
```

1.2.3 Selecting en transposing

You can select elements from a matrix :

$a(i,j)$ returns the element on the i-th row and j-th column.

$a(i,:)$ returns the entire i-th row of a.

$a(:,j)$ returns the entire j-th column of a.

$a(i:j,k:l)$ returns the matrix $a(i\ldots j,k\ldots l)$. e.g.: `>> m = a(1:2,2:3)`
returns

```
m =
     1     2
     4     5
```

To transpose a matrix, you type an accent after the matrix. For example `>> b = a'` returns

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

1.2.4 Calculations in MATLAB

Operators you can use are the following:

$+$	addition	
$-$	subtraction	
$*$	multiplication	
$/$	right division	(a/b means $a*\text{inv}(b)$)
$\backslash$	left division	($a\backslash b$ means $\text{inv}(a)*b$)

You can apply these operators on numbers as well as on matrices.

If you want to compute something in MATLAB, then you type the expression after the prompt, e.g.:

```
>> x+2
```

MATLAB answers:

```
ans =
       2.6500
```

The result is by default assigned to the variable ans (= answer).

You can also assign the result to a variable, e.g.:

```
>> y = ans
```

Then MATLAB answers:

```
y =
       2.6500
```

1.2.5 Some elementary functions in MATLAB

log, exp, sin, cos, tan,...

The argument has to be enclosed in round brackets, e.g.: `log(1)`.

abs returns the absolute value

round returns the nearest integer number

1.2.6 Matrix operations

inv(a) computes the inverse of a

det(a) computes the determinant of a

cond(a) computes the condition number of a

rank(a) computes the rank of a

size(a) returns number of rows and columns of a

norm(a) computes the norm of a vector or a matrix

eye(n) gives the n-dimensional unit matrix

diag(d) returns a diagonal matrix with elements $d(i)$ on its diagonal

rand(n, m) generates a random n-by-m matrix

orth(a) returns an orthonormal basis for the range of a

lu(a) returns the factors of the LU-decomposition of a

After typing

$$>> [l, u] = \mathrm{lu}(a)$$

MATLAB returns two matrices l and u: l is a (eventually permuted) lower triangular matrix with ones on the diagonal and u is an upper triangular matrix.

To solve a linear system $ax = b$, you can use the MATLAB division

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

eig(a) computes eigenvalues and eigenvectors.

After typing **eig**(a) MATLAB gives you a vector with all eigenvalues of a.

To obtain the eigenvectors as well, you type

```
>> [v,d]=eig(a)
```

The two matrices v and d on return have the following meaning: d is a diagonal matrix with the eigenvalues of a on its diagonal, and v contains in its columns the eigenvectors of a. Without roundoff, `a*v == v*d`.

1.2.7 Programs

You can collect a sequence of MATLAB commands in one file, which should have the '.m' extension. This sequence of commands is then executed when you type in the name of that file (without the extension). To ensure that MATLAB will find your program, you may have to adjust MATLAB search path with the command **path** Typing **path** displays MATLAB's current search path. To append a directory to this path, type **path(path,'c:\temp\my_files')** for instance.

1.3 Assignments

1. Type **tour** in a MATLAB session, follow the `Intro to MATLAB` link and browse through the various aspects that interest you.

2. Visit `http://www.mathworks.com/matlabcentral/fileexchange`, this is the site of the MATLAB Central File Exchange which contains user contributions. Download one file, and answer the following two questions: (1) Why did you choose this type of application? (2) What did you learn from it?

3. Type in `u = [2 3 1 4]; v = [4 3 2 1]; w = [1 2 3 4];`

   Give MATLAB commands to create a matrix **a** that has as rows **u**, **v**, and **w**.

   The matrix **a** defines the augmented matrix of a 3x3 linear system with right hand size vector in its fourth column. Compute its reduced row echelon form with **rref**.

   What is the solution of the corresponding linear system?
   Verify that the solution vector `x` makes `b - a*x` (almost) zero.

4. The linear system $Ax = b$ is defined by $A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$.

   Give the MATLAB commands to enter A and b, to find x, the solution to $Ax = b$, and finally to compute the norm of $b - Ax$.

5. Consider the linear system $\begin{cases} 2x_1 + 3x_2 = 1 \\ -2x_1 + x_2 = 3. \end{cases}$

   Give all the MATLAB commands to define this linear system in the format $Ax = b$, with $x = [x_1 \ x_2]^T$, i.e.: what are the commands to define A and b? What is the solution of this system?

6. Type $a = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$, followed by $[l, u] = lu(a)$. Test whether $l * u$ equals a. Notice that u is upper triangular. Explain why l is not lower triangular. (*hint*: type *help lu*.)

7. Generate a random 6×6 matrix a and compute the LU-decomposition of a, storing the factors of the LU-decomposition in the matrices l and u.

   (a) Test whether $l * u$ equals a by computing $r = a - l * u$. Is r the zero matrix?
   Interpret the results and explain what went wrong.

   (b) The determinant of a product of two square matrices is the product of the determinants of the factors in the product. Test whether $det(a) = det(l) * det(u)$. What is the most efficient way to compute the determinant of a matrix, given its LU-decomposition?

8. Create a random 3-by-3 matrix a and compute its eigenvectors and eigenvalues via $[v, d] = \mathbf{eig}(a)$.

   (a) Verify the accuracy of the computed results via the norm of the matrix $a * v - v * d$.
   Interpret the result.

   (b) The product of a with the kth column of v is $d(k, k)$ times the k column of v. Give the MATLAB commands and the output to verify this property of eigenvectors and eigenvalues.

9. A simplex is the three dimensional analogue to a triangle, spanned by four points:

   ```
   a = [2 3 -1]; b = [2 4 9]; c = [3 -1 8]; d = [4 3 1];
   ```

   Start by typing in this line in a MATLAB session.

   To compute the volume of this simplex, we proceed as follows:
   (1) define the matrix whose rows are `b - a`, `c - a`, and `d - a`;
   (2) compute the absolute value of the determinant of this matrix;
   (3) divide this value by six.

   What is the volume of this simplex? Give all relevant MATLAB commands.