



Introduction to Matlab

Instructor:

ir drs E.J Boks

Electrical Engineering

Embedded Systems Engineering

phone:(026) 3658173

Room: D2.03

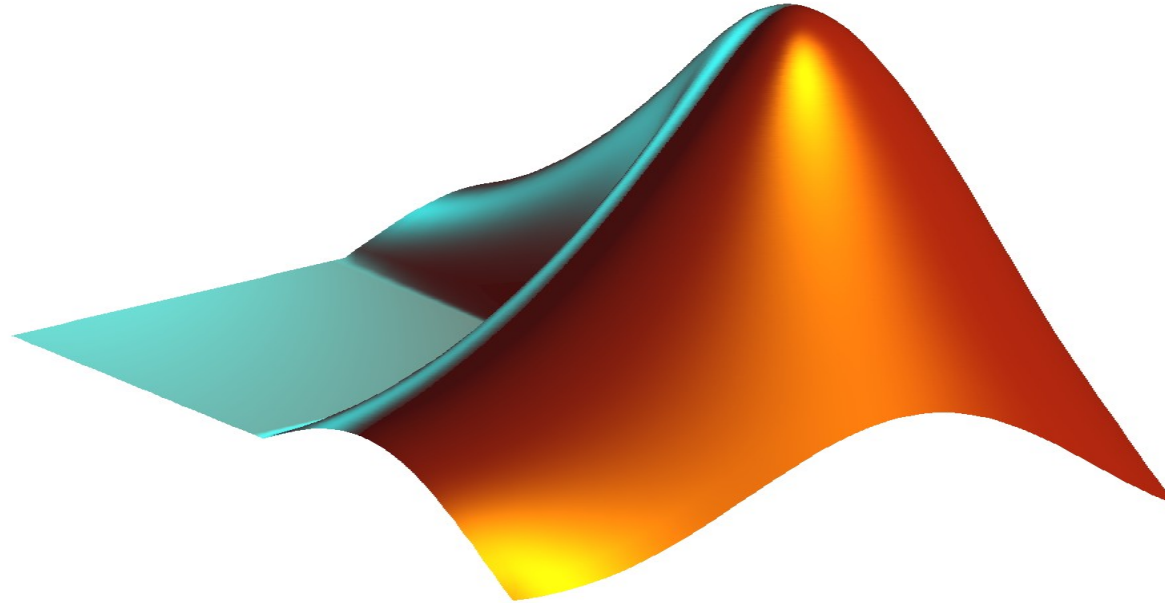
e-mail:ewout.boks@han.nl



Matlab lesson one

- Introduction and history of matlab
- Overview of software, add-ons, options and licenses
- Overview of this course
- Introduction to the desktop and user interface
- Matrices in Matlab
- Files in Matlab
- Linear Algebra and Arrays
- Matlab help
- Matlab exercise – your first attempt

Matlab stands for *Matrix Laboratory*



- Complete mathematical toolset based on matrix manipulating language
- Modular set-up
- Matlab is aimed at scientific, engineering and financial communities
- Core strengths: Modelling, simulation and prototyping.

Matlab history

- Matlab started out of a set of matrix manipulating software libraries: “Linpack” and “Eispack”. These libraries are still usable on most Unix systems.
- The Mathworks was founded 1984 by Cleve Moler and Jack Little to streamline the commercial offering of Matlab.
- Introduction of a complementary simulation tool – Simulink – was accomplished in 1992
- Core packages are extended with an ever increasing proliferation of toolboxes and blocksets – see the Mathworks web-site.

Matlab contact information

The Mathworks inc:

3 Apple Hill Drive
Natick, MA 01760-2098
Phone: 508-647-7000
Fax: 508-647-7001

in the Netherlands:

Postbus 280
2800 AG Gouda
telephone: (0182) 537644

internet: <http://www.mathworks.com>

Matlab fundamentals

At the core of the matter, Matlab is used because it removes the hard edge from evaluating your engineering, financial or scientific problem.

In the past, many of these problems remained theoretical because the mathematics involved demand computations that exceed the level of the desktop calculator. The engineer, financial analyst or scientist could only make estimated guesses of the outcome of his problem, or he could 'trim down' his model to simplify the computations. None of these are recommended but sometimes they were the sole way of getting to a result.

With the advent of computers, it became possible to numerically compute a solution of a problem. This, however, implied that the person involved was proficient enough with computers to write his own software. A numerical computation was therefore only limited to those who were computer-savvy.

Matlab removes this barrier. You don't need software engineering skills in order to numerically evaluate your engineering, financial or scientific problem.

Matlab fundamentals (2)

Matlab is a command line based tool that interprets user commands. These commands are used to define input data, mathematical relations and numerical and graphical outputs.

Matlab uses matrices as the standard data format. So all user data must be presented in matrix or vector form, even though the user does not apply matrices on paper him- or herself.

Matlab uses functions as the standard way to process data. Each function is called by its name, with optional arguments included. The output, ie after executing the function, is presented to the user on the screen in either graphical or textual presentation.

There are hundreds of functions present in the Matlab core package. These are by itself already sufficient to help the user solve many problems.

In addition, extra packages (called toolboxes) are available to supplement the core package. They contain extra functions to cover a specific computational area.

What is Simulink ?

Simulink is a completely autonomous tool (ie not a Matlab toolbox), although it communicates easily with Matlab.

Simulink is a graphical process simulator that provides a designer with the ability to simulate dynamic, non-linear systems. It therefore opens up the design of non-linear systems that otherwise could not be conceived except with real-life prototypes.

Simulink employs an intuitive graphical interface that hardly requires a learning curve – most people will 'get in' and 'drive off' !

Symbols and variables can easily be exchanged with Matlab (which can run concurrently) thus providing an integrated development environment.

As with Matlab, the Simulink core package contains a number of basic ingredients. Additional components (called blocksets) are available to cater for special interests.

Matlab basic set-up

- Matlab runs on :
 - Unix (Solaris, HP-UX, Mac OS X, Linux, FreeBSD)
 - MS Windows
- on all platforms a similar Integrated Development Environment
- Matlab's core is a mathematical function library
- The primary interface is the matrix manipulation language
- Matlab has a powerful graphics system which helps in plotting data effortlessly
- Matlab contains extensive help documentation

Introduction to matrix operations (1)

- Row of data: $\langle \text{name} \rangle = [\langle \text{value 1} \rangle \langle \text{value 2} \rangle \dots \langle \text{value n} \rangle]$
- Vector of data: $\langle \text{name} \rangle = [\langle \text{value 1} \rangle \langle \text{value 2} \rangle \dots \langle \text{value n} \rangle]'$

Example (enter it into the cmd window):

$x = [4 \ 6 \ 7]$ <enter>

$w = [2 \ 4 \ 3]'$ <enter>

- Matrix $\langle \text{name} \rangle = [a_1 \ a_2 \ \dots \ a_n ; b_1 \ b_2 \ \dots \ b_n ; ; ;]$

$W = [1 \ 2 \ 3 ; 7 \ 5 \ 3 ; -34 \ 2.5 \ 2]$ <enter>

$q = [2 \ 4 \ 3];$ <enter>

Notes:

- Matlab is case-sensitive (like all Unix systems) $\rightarrow w \neq W$
- Commands are executed by typing <enter>
- A semicolon hides the output of the last cmd execution

Introduction to matrix operations (2)

- Colon operator : denote ranges with a specified interval by using the colon. Syntax `<name> = [begin:interval:end]`

Example :

`x = [1 : 0.1 : 40]`

- concatenation: create a new matrix out of other matrices.

Example:

`A = [2 -4; 5 6]; B = [1 -2; -22 -21];`

`C = [A B]`

`D = [A ; B]`

- deleting columns: `<name>(:,n) = []` removes column n.
- deleting rows: `<name>(m,:) = []` removes row m.

Introduction to matrix operations (3)

```
A = [ 3 4 5 ]'; % first vector
```

```
B = [12 13 14]'; % second vector
```

```
C = A-B; % subtraction
```

```
D1 = A'*B; % matrix multiplication
```

```
D2 = A.*B; % pointwise multiplication
```

```
E = A./B; % pointwise division
```

special matrices

- Create a matrix with ones: `ones(n)` creates a $n \times n$ dimensional matrix filled with ones.
- Create a matrix with zeros: `zeros(n)` creates a $n \times n$ dimensional matrix filled with zeros.
- Create the identity matrix: `eye(n)` creates a $n \times n$ identity matrix.
- Create a random matrix: `rand(n)` or `randn(n)` creates a $n \times n$ random matrix.

Matlab Files

- File Menu “save” and “load” : preserve and restore the workspace.
- Matlab m-files: ascii files containing matlab scripts or functions.
- m-files are loaded and executed by typing their name. Pay attention the the execution path (your work dir) when executing new m-files.

Linear Algebra

Matlab contains many tools to execute linear algebra functions. Examples are :

- `det(A)` yields the determinant of A
- `eig(A)` yields the eigenvalues of A
- `inv(A)` calculates the inverse of A
- `rref(A)` gives the reduced row echelon form of A

Most Matlab commands have their own help page (similar to Unix man pages. Type `help <command>` to see the help for the command.

Matlab Help

Most Matlab commands have their own help page
(similar to Unix man pages)

Type **help <command>** to see the help for a command.

Also, the development environment contains a help environment in HTML format. This contains the same information, but presented in the HTML help browser.

Knowledge of using the help system efficiently is crucial to employing Matlab effectively.

Exercises (1)

1. Solve the following set of equations:

$$2x + 3y - 2z = 5$$

$$6x + 5y + 3z = -12$$

$$-5x - 3y - 7z = 4$$

a: using matrix inversion

b: with reduced row echelon form (rref)

c: (advanced) with determinants

Exercises (2)

(Advanced) Solve the following set of equations:

2: (λ = eigenvalue vector):

calculate λ

$$A = \begin{bmatrix} 7 & 8 & 9 \\ 4 & -5 & 6 \\ -4 & -3 & 2 \end{bmatrix}$$
$$A * x = \lambda * x$$

3: Calculate $D = C^T A C$

$$C = \begin{bmatrix} 12 & 43 & 3 & -5 \\ -2 & -3 & 5 & 34 \\ 1 & 2 & 3 & 4 \\ 6 & 8 & 5 & 3 \end{bmatrix}$$