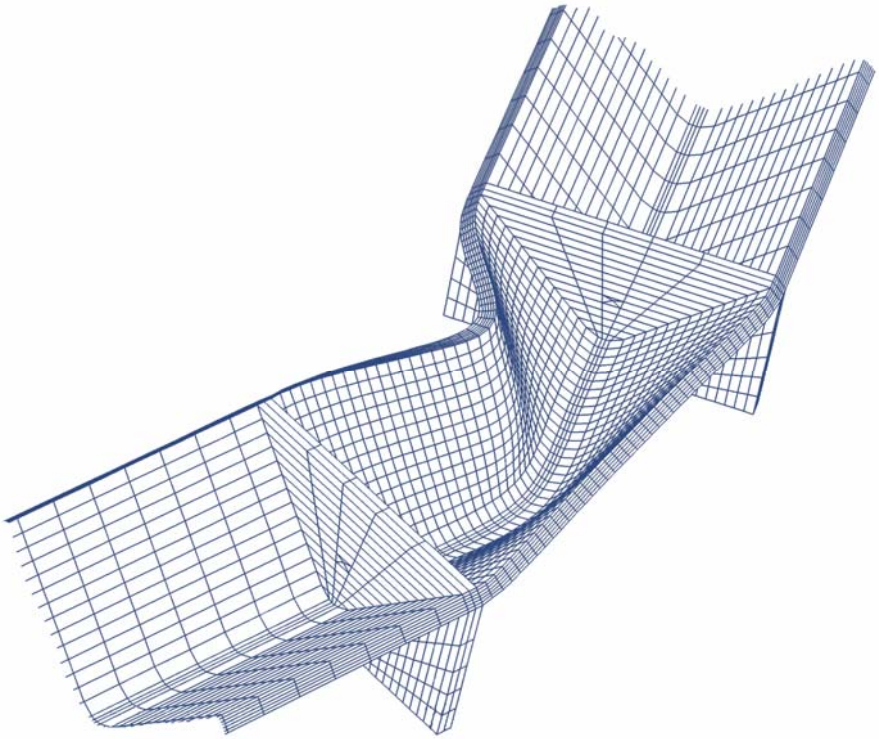
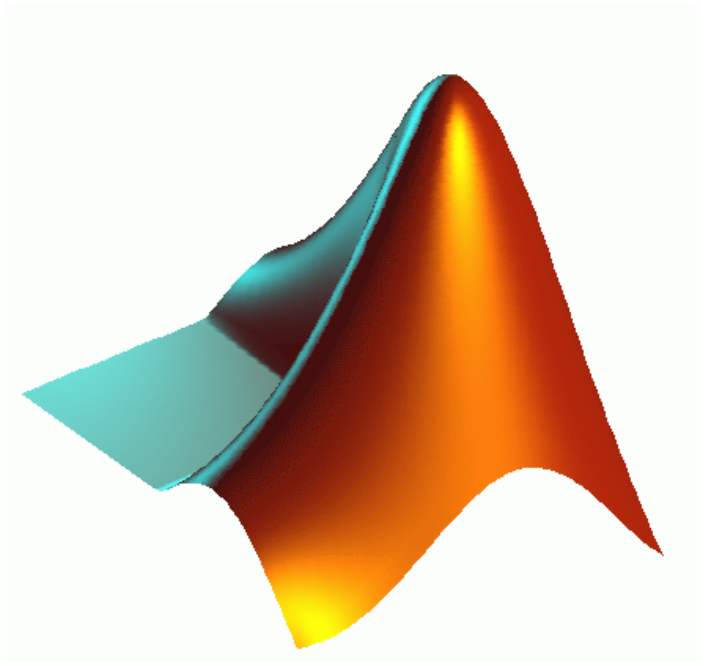
Short introduction to Matlab



Prof. Dr. Michael Havbro Faber
Swiss Federal Institute of Technology
ETH Zurich, Switzerland

Short introduction to Matlab

Overview

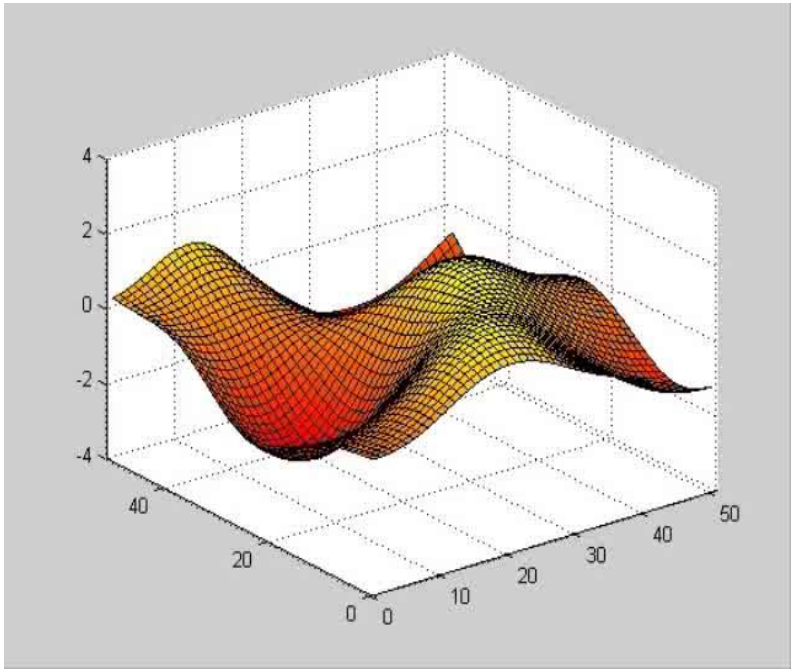


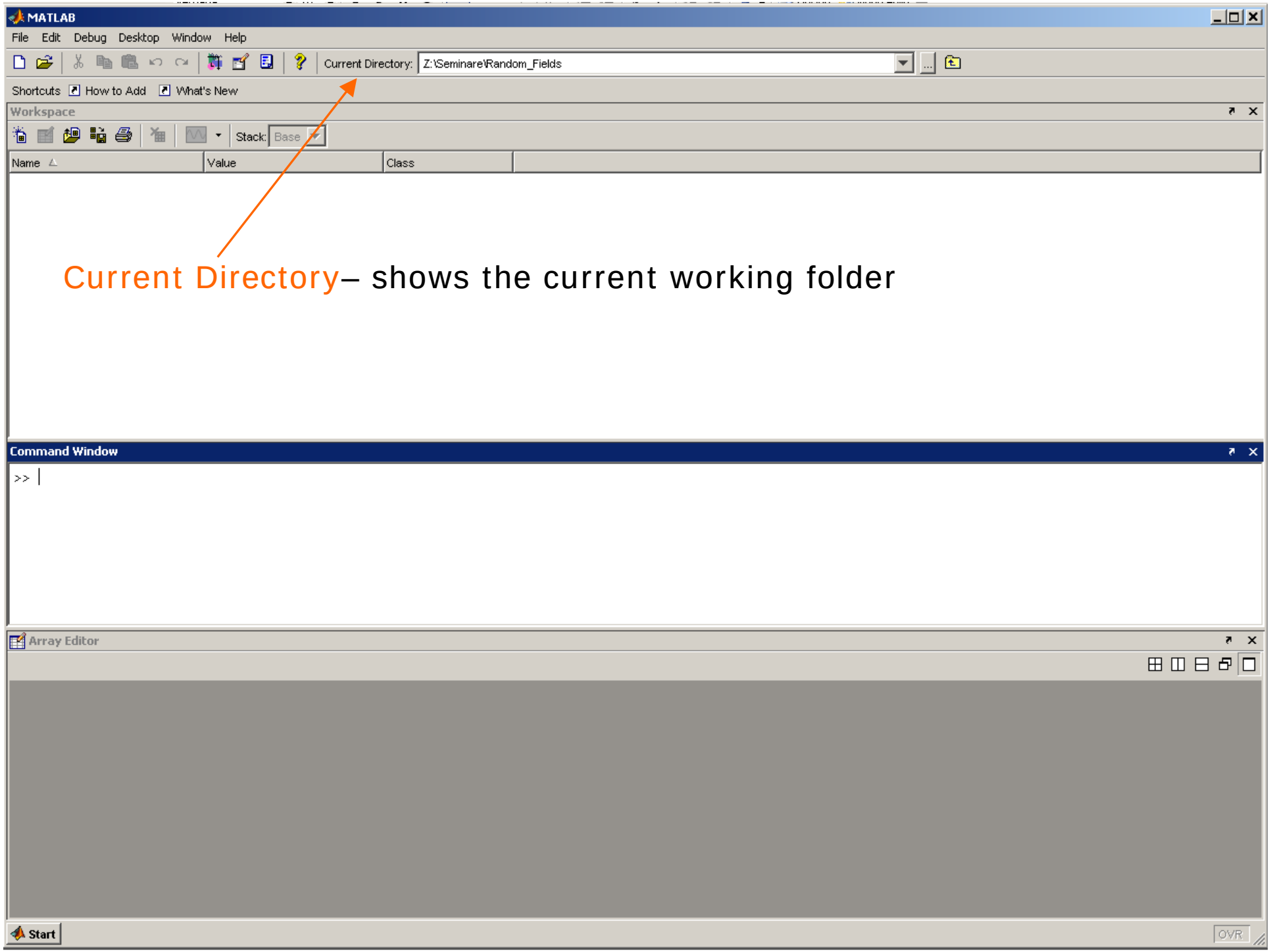
- Introduction
- Matlab user interface
- Creating Variables
- Working with matrices
- The help function
- The m-file
- Plots

Short introduction into Matlab©

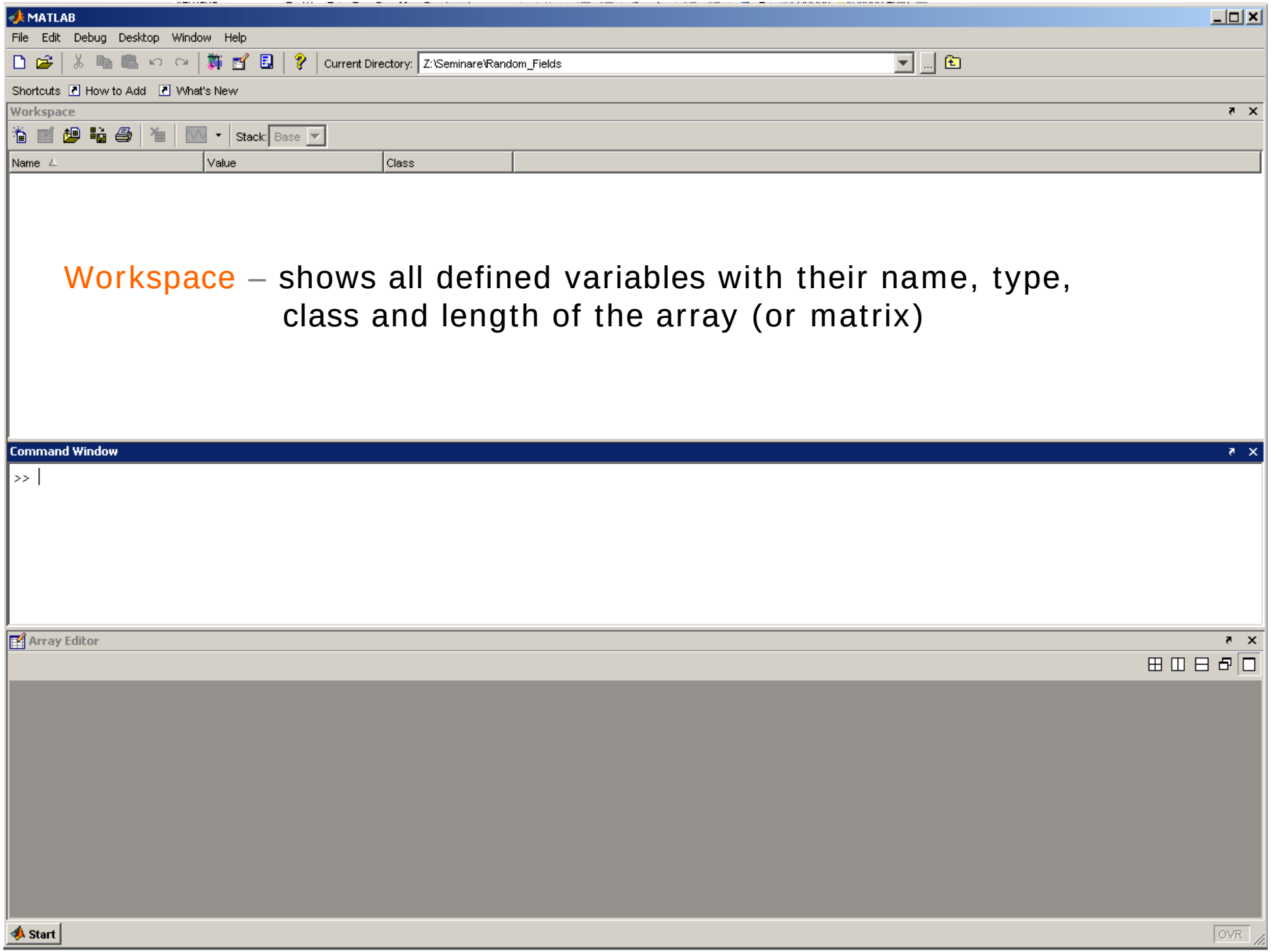
Why will we use Matlab in this course?

- It is useful for numerical calculations
- Symbolic calculations can be performed
- Easy visualization of data
- Interactive use
- Use as a programming language (easy programming)

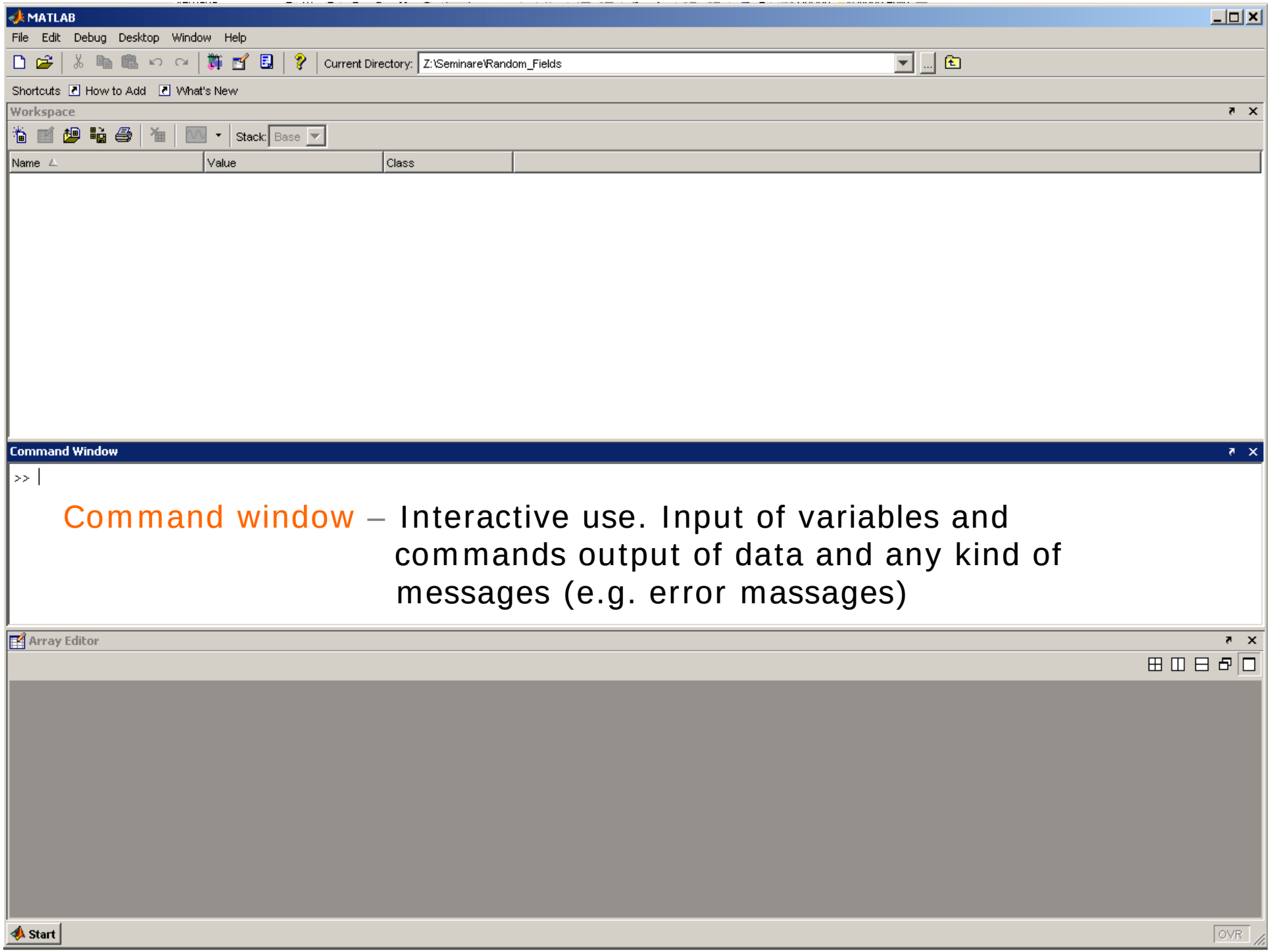




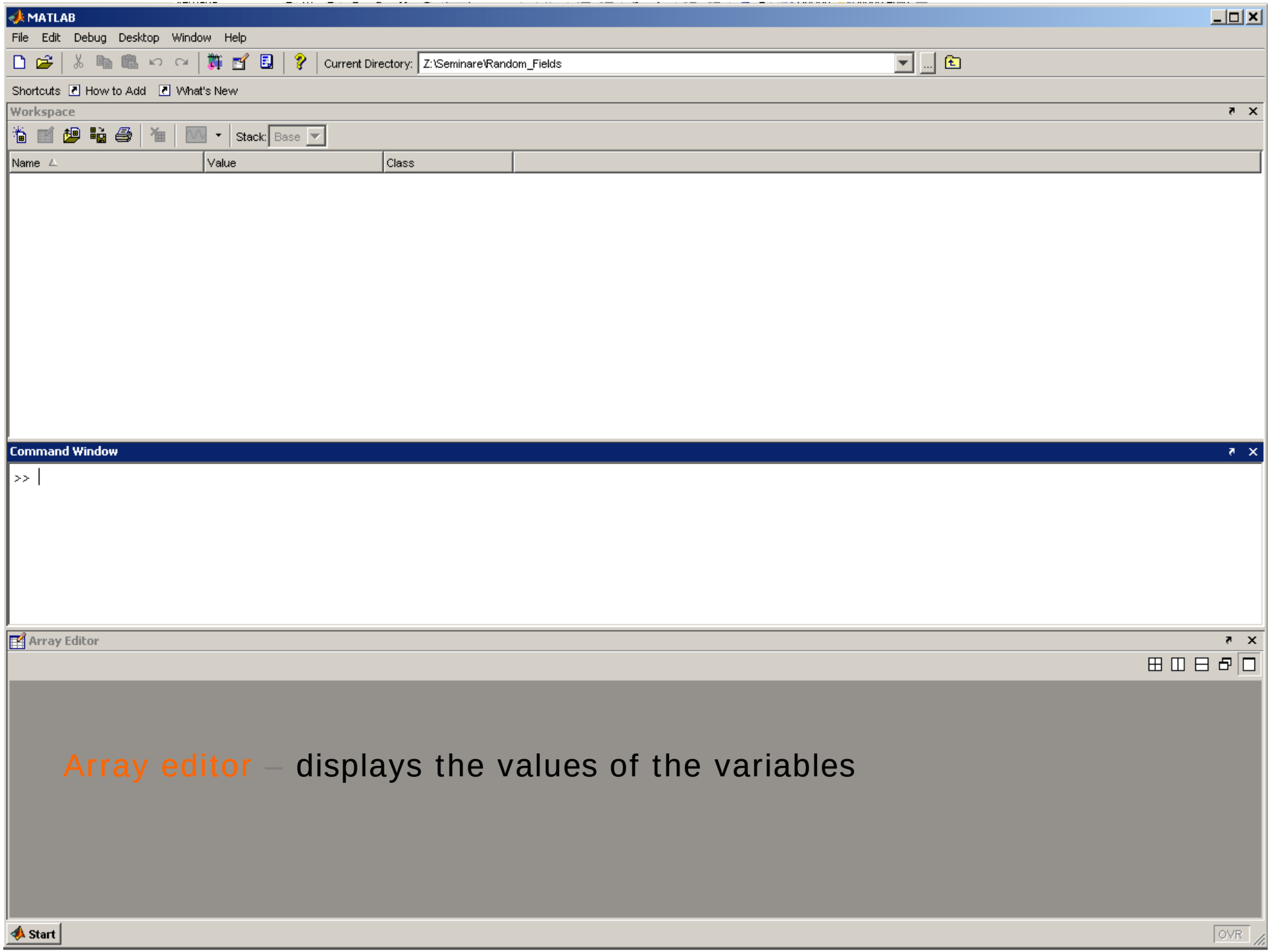
Current Directory– shows the current working folder

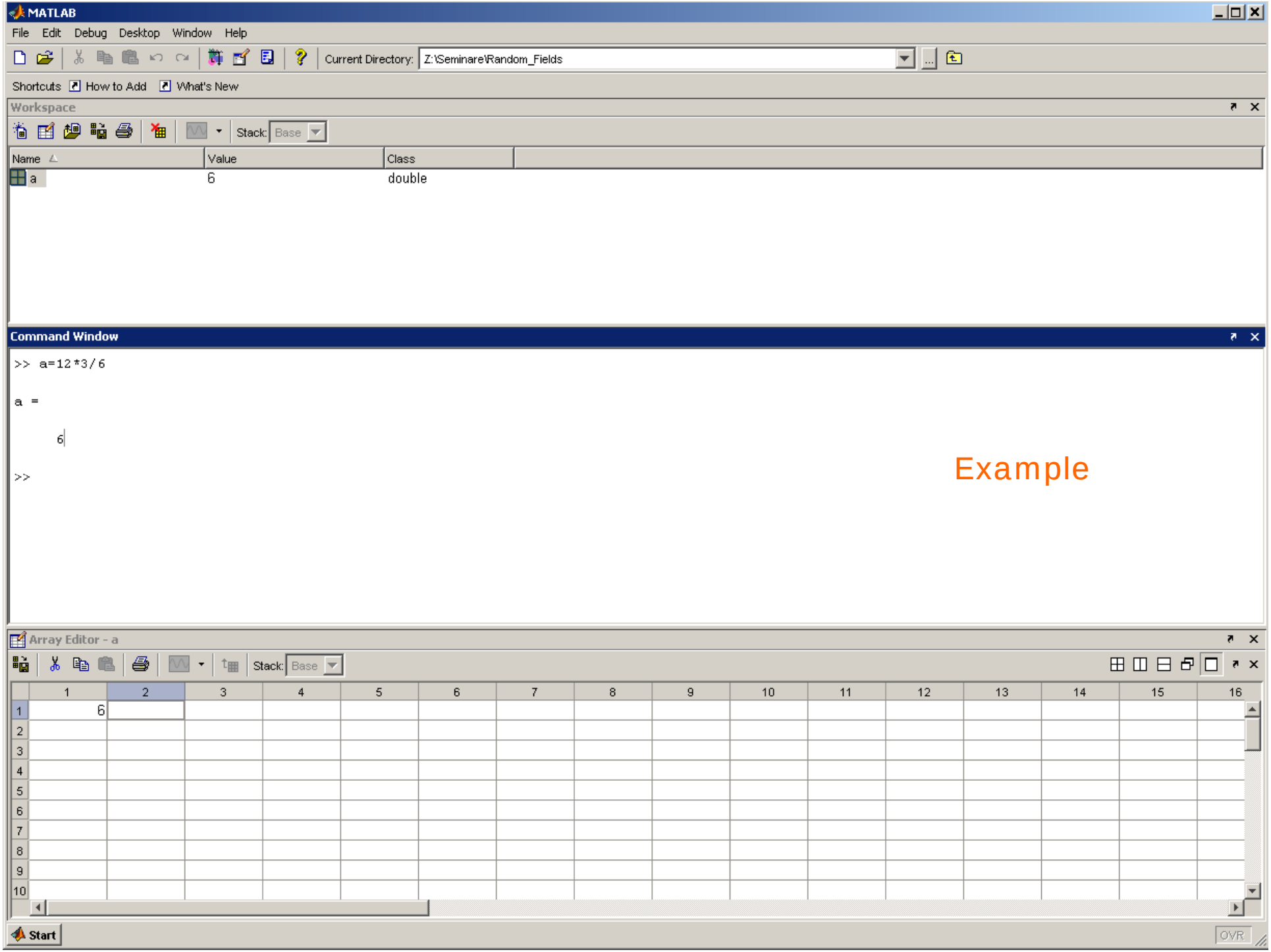


Workspace – shows all defined variables with their name, type, class and length of the array (or matrix)



Command window – Interactive use. Input of variables and commands output of data and any kind of messages (e.g. error massages)


MATLAB
File Edit Debug Desktop Window Help
Current Directory: Z:\Seminare\Random_Fields
Shortcuts How to Add What's New
Workspace
Stack: Base
Name
Value
Class
Command Window
>>
Array Editor
Array editor – displays the values of the variables
Start
OVR

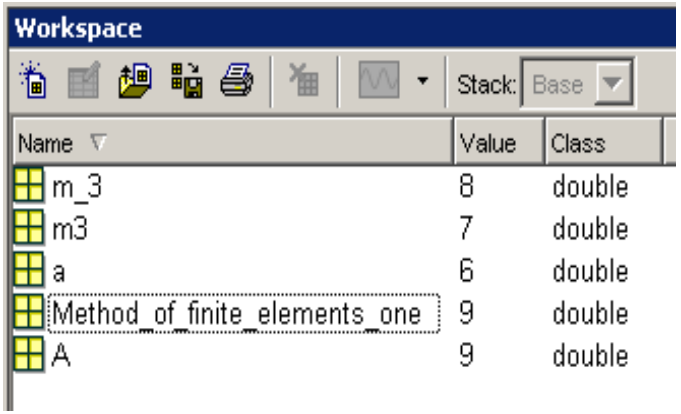

MATLAB
File Edit Debug Desktop Window Help
Current Directory: Z:\Seminare\Random_Fields
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
a 6 double
Command Window
>> a=12*3/6
a =
6
>>
Array Editor - a
Stack: Base
Start
OVR

Example

Variables

Rules for defining variables:

- The name of the variables must not include special characters (%,&,*,...) exception: underscore (_)
- The first character has to be a alphabetic character (m3: ok. – 3m: not ok.)
- Matlab distinguishes between capitals and lower case letters
- With `clear'name'` you can delete variables



Short introduction to Matlab

Defining matrices and vectors

Defining vectors and matrixes:

```
>>x=[0 2 4 6 8]

              x= 0 2 4 6 8
```

x ~ matrix with 1 row and 5 columns

By using a comma the input may be easier to read:

```
>>x=[0,2,4,6,8]

              x= 0 2 4 6 8
```

Short introduction to Matlab

Defining matrices and vectors

Sometimes vectors can be created easier:

```
>>x=0:2:8

            x= 0 2 4 6 8
```

linspace(x,y,n) generates n points between x and y.

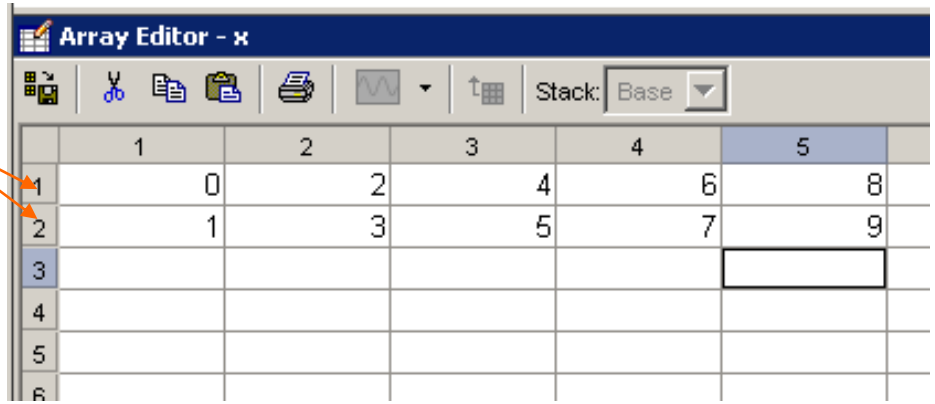
```
>>x=linspace(0,8,5)

            x= 0 2 4 6 8
```

Short introduction to Matlab

Defining matrices and vectors

Matrix

```
>>x=[0 2 4 6 8; 1 3 5 7 9]

x = 0 2 4 6 8
    1 3 5 7 9
```



Or:

```
>>x=[0,2,4,6,8; 1,3,5,7,9]

x = 0 2 4 6 8
    1 3 5 7 9
```

Working with matrices

1) Element wise operations

Multiplication of a matrix with a scalar

```
>>x=[0 2 4 6 8; 1 3 5 7 9]*2
       X=
              0 4 8 12 16
              2 6 10 14 18
```

Element wise multiplication of two matrices

```
>>x=[0 2 4 6 8; 1 3 5 7 9].*[0 2 4 6 8; 1 3 5 7 9]

              X=
                     0 4 16 36 64
                     1 9 25 49 81

>>x=[0 2 4 6 8; 1 3 5 7 9].^2
```

Working with matrices

1) Element wise operations

Element wise division of two matrices

```
>>x=[0 2 4 6 8; 1 3 5 7 9]./[0 2 4 6 8; 1 3 5 7 9]

                X=
                      NaN 1 1 1 1
                      1   1 1 1 1
```

Addition of two matrices

```
>>x=[0 2 4 6 8; 1 3 5 7 9]+[0 2 4 6 8; 1 3 5 7 9]

                X=
                       0 4  8 12 16
                       2 6 10 14 18
```

Short introduction to Matlab

Working with matrices

1) Matrix operations

Transpose a matrix

```
>>x=[3 4 -2; -1 2 8; 2 0 5];
>>y=transpose(x)

        y=      3 -1  2
                4  2  0
               -2  8  5
```

Or:

```
>>y=x'
```

Short introduction to Matlab

Working with matrices

1) Matrix operations

Invert a matrix

```
>>x=[3 4 -2; -1 2 8; 2 0 5];
>>y=inv(x)
```

```
y=      -0.4545    0.9091    1.6364
         0.5000   -0.5000   -1.0000
        -0.1818    0.3636    0.4545
```

Working with matrices

1) Matrix operations

Eigenvalue of a matrix

```
>>x=[1 2 3; 4 5 6; 7 8 9];
>>y=eig(x)

        y=      16.1168
                -1.1168
                -0.0000
```

Sometimes a specific cell of a matrix is needed for the calculation.
If you want to get the 3rd value of the vector, write:

```
>>z=y(3,1)

        z=-1.304e-15
```

Working with matrices

1) Matrix operations

Or you are interested in the first column of matrix x:

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

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

then you can select the whole row or column by using the colon

```
>>x(1,:)

        ans=   1
               4
               7
```

Working with matrices

1) Matrix operations

Other helpful comands

```
det(x)      -  is the  determinant of the matrix x
eye(n,m)    -  creates a n x m identity matrix
zeros(n,m)  -  creates a n x m zero matrix (helpful to allocate memory)
ones(n,m)   -  creates a n x m matrix filled with ones
size(x)     -  gives the dimension of the matrix x
```

If you perform matrix operation,
the dimensions of the matrix must agree

e.g. `y=ones(4,3)` and `x=ones(1,4)`

```
>>z=y*x
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Short introduction to Matlab

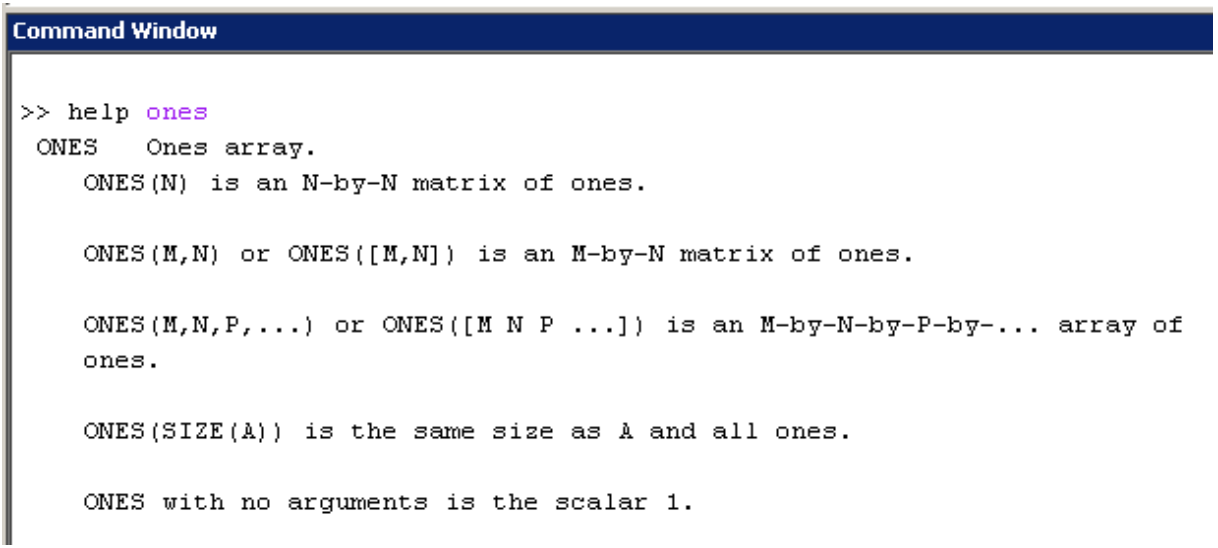
The help function

Here you find everything you need!

```
>>help 'function'
```

Helps you to find the exact notation for all functions, e.g:

```
>>help ones
```



Help

File Edit View Go Favorites Desktop Window Help

Help Navigator

Contents Index Search Demos

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

Search index for: ones

Name	Product
ones	
outputting	Signal Processing...
ones [1] [2]	MATLAB
sparse matrices	
onesided method	Signal Processing...
online algorithms	System Identificati...
online conversions	
addition and subtraction	Simulink Fixed Point
multiplication with zero bias and misma	Simulink Fixed Point
multiplication with zero biases and mat	Simulink Fixed Point
signals	Simulink Fixed Point
online conversions with and fixed-point dat	Stateflow
online documentation, displaying	MATLAB
online help	Communications ...
online help	Embedded Target ...
online help	Link for ModelSim®
online help	Instrument Control...
online help	
Visual Query Builder	Database Toolbox
online help	Data Acquisition T...
online help	Control System To...
online help [1] [2]	MATLAB
viewing	
OPC	
about	OPC Toolbox
OPC Configuration block	OPC Toolbox
OPC Data Access client	OPC Toolbox
OPC Data Access Group	OPC Toolbox
OPC Data Access Item	OPC Toolbox
OPC Foundation Core Components	OPC Toolbox
OPC GUI	
Logging Panel	OPC Toolbox
OPC item properties	OPC Toolbox
OPC Quality Parts block	OPC Toolbox
OPC quality strings	OPC Toolbox
OPC Read block	OPC Toolbox
OPC recommended properties	OPC Toolbox
OPC server ProgId	OPC Toolbox
OPC specific properties	OPC Toolbox
OPC Toolbox	
basic procedure	OPC Toolbox
OPC Toolbox block library	
using	OPC Toolbox
OPC Toolbox objects	OPC Toolbox
configuring	
deleting	
loading	
saving	
server relationship	
OPC Toolbox Objects Toolbar	OPC Toolbox
OPC Write block	OPC Toolbox
opccallback function	OPC Toolbox
read operations	

Title: ones (MATLAB Functions)

MATLAB Function Reference

ones

Create array of all ones

Syntax

```
Y = ones(n)
Y = ones(m,n)
Y = ones([m n])
Y = ones(m,n,p,...)
Y = ones([m n p ...])
Y = ones(size(A))
ones(m, n,...,classname)
ones([m,n,...],classname)
```

Description

`Y = ones(n)` returns an `n`-by-`n` matrix of `1`s. An error message appears if `n` is not a scalar.

`Y = ones(m,n)` or `Y = ones([m n])` returns an `m`-by-`n` matrix of ones.

`Y = ones(m,n,p,...)` or `Y = ones([m n p ...])` returns an `m`-by-`n`-by-`p`-by-`...` array of `1`s.

> **Note** The size inputs `m`, `n`, `p`, `...` should be nonnegative integers. Negative integers are treated as 0.

`Y = ones(size(A))` returns an array of `1`s that is the same size as `A`.

`ones(m, n,...,classname)` or `ones([m,n,...],classname)` is an `m`-by-`n`-by-`...` array of ones of data type `classname`. `classname` is a string specifying the data type of the output. `classname` can have the following values: `'double'`, `'single'`, `'int8'`, `'uint8'`, `'int16'`, `'uint16'`, `'int32'`, `'uint32'`, `'int64'`, or `'uint64'`.

Example

```
x = ones(2,3,'int8');
```

See Also

`eye`, `zeros`, `complex`

odextend open

Programming with matlab – the m.files

Two different types of m.files are distinguished by matlab

- script files: all variables are global

- function files: all variables are local;
input and output parameters are passed

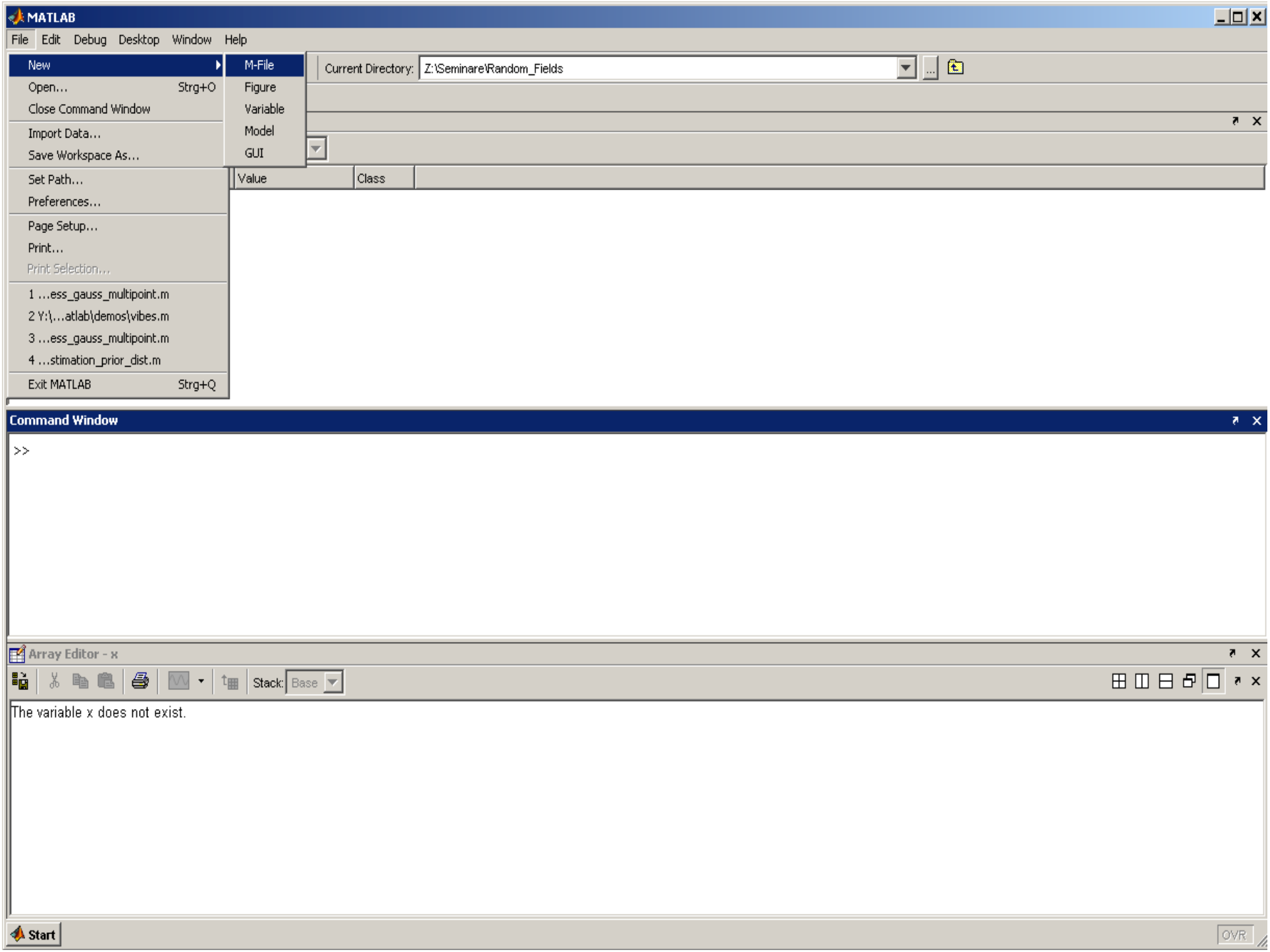
m.files

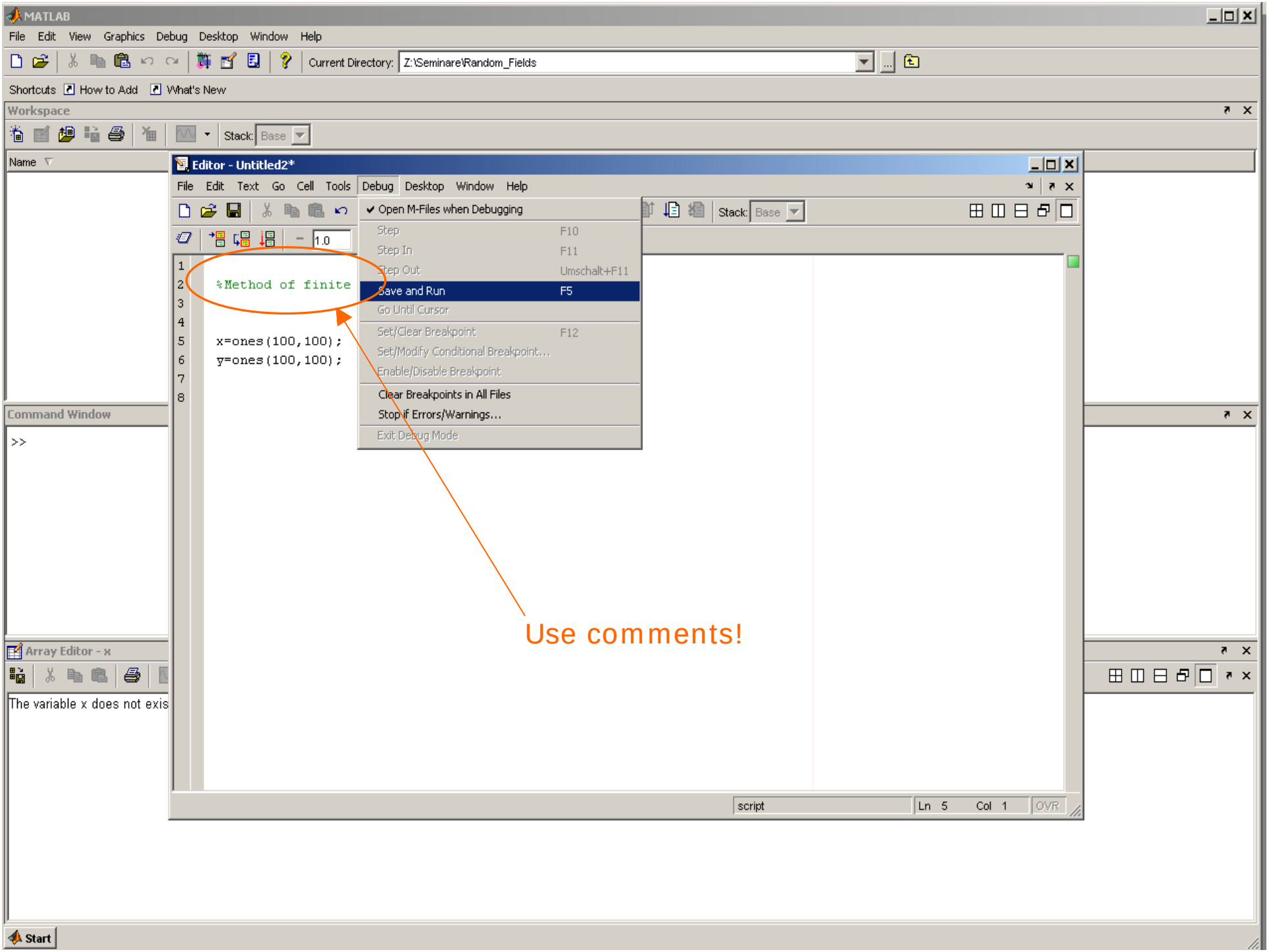
The commands are written in a text editor and saved with the ending .m

The commands are written like in the command line

All commands available in the command line can be used
(*for* loops – *if else*, etc.. + self programmed functions)

All commands are processed line by line

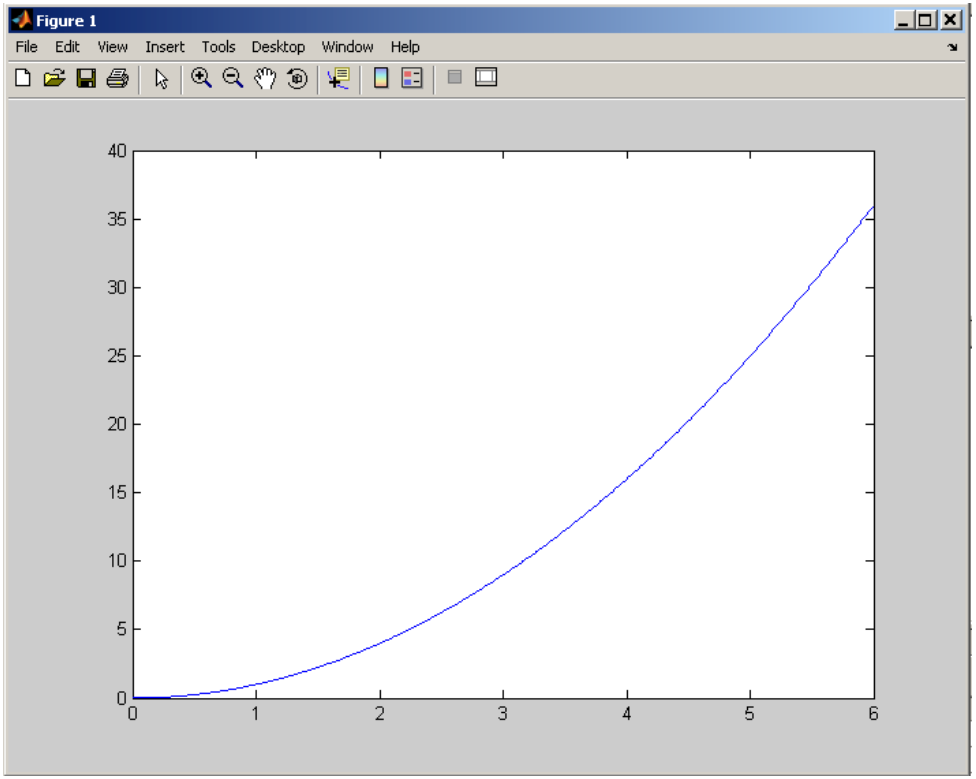

MATLAB
File Edit Debug Desktop Window Help
New
Open... Strg+O
Close Command Window
Import Data...
Save Workspace As...
Set Path...
Preferences...
Page Setup...
Print...
Print Selection...
1 ...ess_gauss_multipoint.m
2 Y:\...atlab\demos\vibes.m
3 ...ess_gauss_multipoint.m
4 ...stimation_prior_dist.m
Exit MATLAB Strg+Q
M-File
Figure
Variable
Model
GUI
Current Directory: Z:\Seminare\Random_Fields
Value
Class
Command Window
>>
Array Editor - x
Stack: Base
The variable x does not exist.
Start
OVR


MATLAB
File Edit View Graphics Debug Desktop Window Help
Current Directory: Z:\Seminare\Random_Fields
Shortcuts How to Add What's New
Workspace
Stack: Base
Name
Editor - Untitled2*
File Edit Text Go Cell Tools Debug Desktop Window Help
Open M-Files when Debugging
Step F10
Step In F11
Step Out Umschalt+F11
Save and Run F5
Go Until Cursor
Set/Clear Breakpoint F12
Set/Modify Conditional Breakpoint...
Enable/Disable Breakpoint
Clear Breakpoints in All Files
Stop if Errors/Warnings...
Exit Debug Mode
1.0
1 %Method of finite
2
3 x=ones(100,100);
4 y=ones(100,100);
5
6
7
8
Use comments!
Command Window
>>
Array Editor - x
The variable x does not exis
script Ln 5 Col 1 OVR
Start

Plotting with Matlab

Easy visualization of data using the manifold plot options

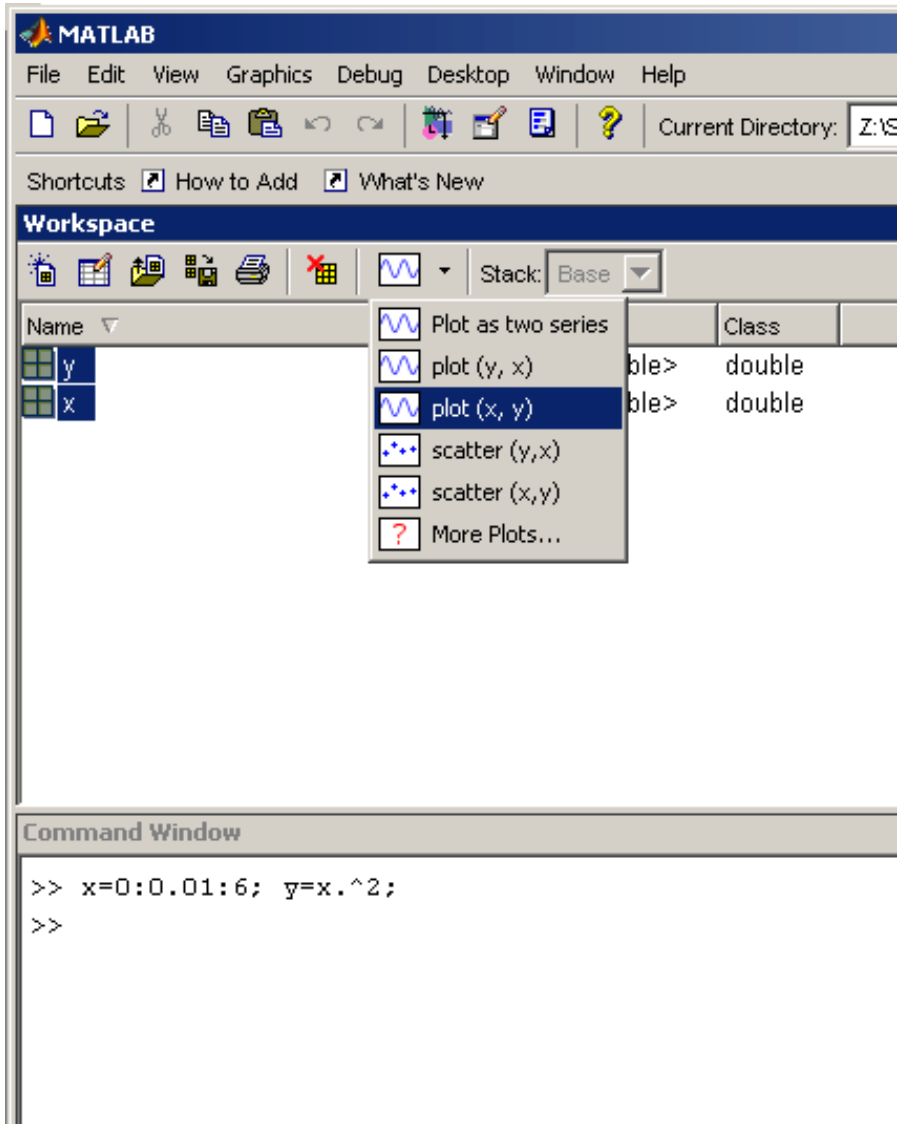
```
>>x=0:0.01:6; y=x.^2;

>>plot (x,y)
```



Short introduction to Matlab

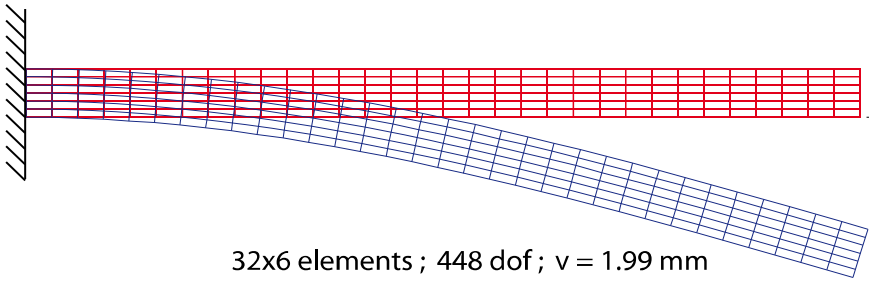
Plotting with Matlab

Or by using the graphical user interface

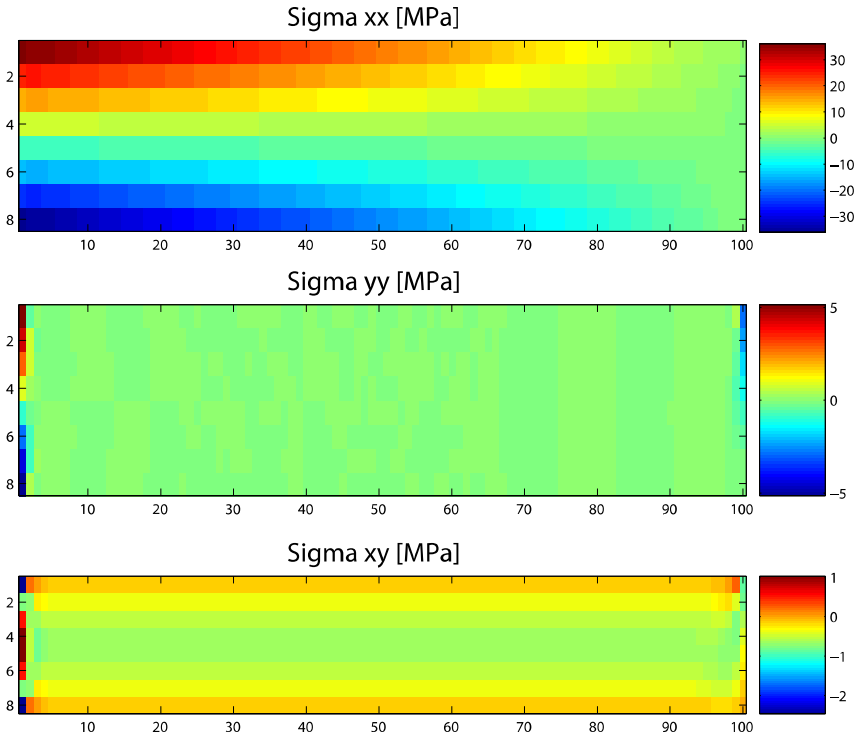


Short introduction to Matlab

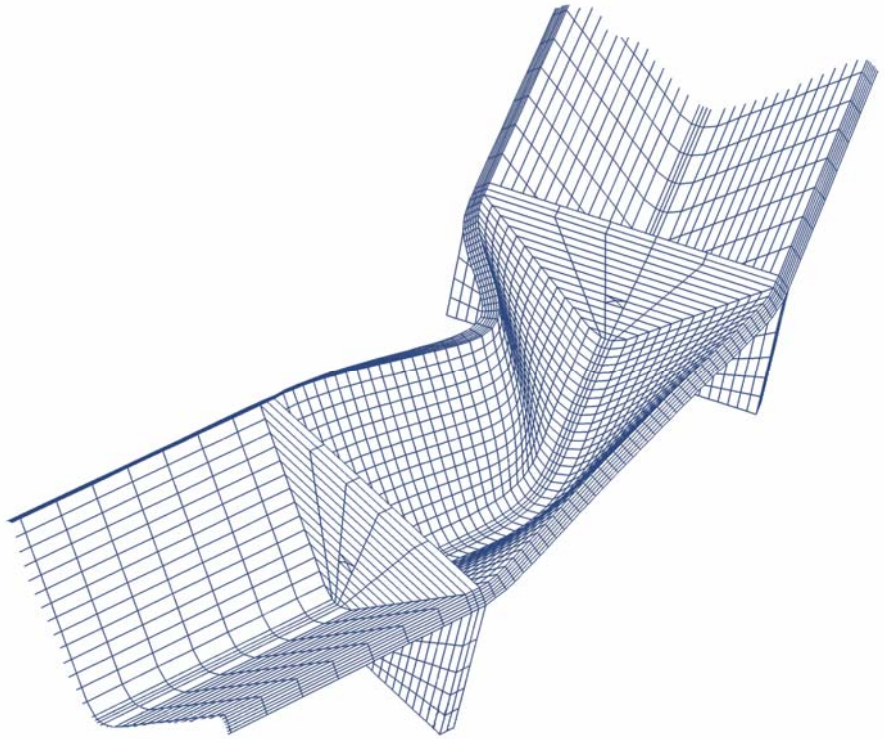
Visualize your FE results by using Matlab



32x6 elements ; 448 dof ; v = 1.99 mm



Short introduction to Matlab



Prof. Dr. Michael Havbro Faber
Swiss Federal Institute of Technology
ETH Zurich, Switzerland