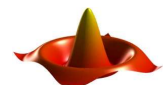


MATLAB

Basics

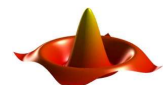


MATLAB Basics

Command Desktop

Integrated development system:

- **Command Window**
- **Editor**
- **Command History**
- **Workspace Browser**
- **Profiler**
- **Current Directory Browser**
- **Shortcut-Bar**



MATLAB Basics

Online–Help

- Online-Help for all commands, functions etc.
- Command Window Help: `help [command]`
- Help-Browser: `helpwin [command]`
`doc [command]`
- Search for *searchstring*: `lookfor searchstring`



MATLAB Basics

Variables

- Names for variables:
 - maximum of 63 characters
 - Letters, underscore “_” und digits
 - First character must be a letter
 - Distinguish small and capital letters
- Assign value to variable: `variable_1 = 25`
- Standard answering variable `ans`
- Variables are globally defined in Workspace



MATLAB Basics

Types of Variables

- Types of Variables:

logical	logical values 0 oder 1
char	character (letters)
single	floating point single precision 32 Bit
double	floating point single precision 64 Bit
intX	integer X = 8, 16, 32, 64
uintX	unsigned integer X = 8, 16, 32, 64
struct	array of C-like structures
cell	array of indexed cells
map data	fast key lookup data structure

- double (default) and char widely used
- Optimised storage with sparse in contrast to full



MATLAB Basics

Output, basic math operations and constants

- Comma or space after expression shows output, semi-colon omits output
- Operations:

+	Addition	-	Subtraction
*	Multiplication	/	Division
^	Power		
- Constants:

pi	Pi
eps	floating point accuracy
inf	Infinity
NaN	Not-a-Number
- Complex numbers: i, j Imaginary Unit $\sqrt{-1}$



MATLAB Basics

Vectors and matrices

- Vectors: Separation of elements by comma or space:

```
vektor = [ 1 2 3 ]
```

- Matrices: Separation of rows by semi-colon:

```
matrix = [ 1 2 3 ; 5 6 7 ]
```

- First element has index 1

- Colon for row/column: `zeile_1 = matrix(1,:)`

- Combining: `matrix = [ vektor ; 5 6 7 ]`

- Last element: `matrix(:,end)`



MATLAB Basics

Specific vectors und matrizes

- Elements with same step width: `fort = (1:2:100)`
- Number *num* of points between *start* and *stop*:
 - linear: `linspace(start,stop,num)`
 - logarithmic: `logspace(start,stop,num)`
- Specific matrizes(*m* rows, *n* columns):
 - Identity matrix: `eye(m)`
 - Matrix of ones: `ones(m,n)`
 - Matrix of zeros: `zeros(m,n)`
 - Random matrix: `rand(m,n)`



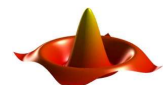
MATLAB Basics

Math functions

- Numerous math & trigonometric functions

<code>sqrt(x)</code>	square root	<code>rem(x,y)</code>	rest after division x/y
<code>exp(x)</code>	exponential function	<code>round(x)</code>	round
<code>log(x)</code>	natural Logarithm	<code>ceil(x)</code>	round to ceil
<code>log10(x)</code>	logarithmu of 10	<code>floor(x)</code>	round to floor
<code>abs(x)</code>	absolut	<code>sum(v)</code>	sum of vector elements
<code>sign(x)</code>	signum	<code>prod(v)</code>	product of vector elements
<code>real(x)</code>	real part	<code>min(v)</code>	smalles vector element
<code>imag(x)</code>	imaginary part	<code>max(v)</code>	largest of vector element
<code>angle(x)</code>	angle of complex number	<code>mean(v)</code>	arithmetic mean
<code>sin(x)</code>	sine	<code>atan(x)</code>	arcus-tangens $\pm 90^\circ$
<code>cos(x)</code>	cosine	<code>atan2(x,y)</code>	arcus-tangens $\pm 180^\circ$
<code>tan(x)</code>	tangens	<code>sinc(x)</code>	sinc-function $\sin(\pi x)/(\pi x)$

- Online-Help by `help elfun` and `help datafun`

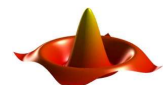


MATLAB Basics

Calculating with vectors und matrizes

- Lots of operations usable with vectors and matrizes
- element-by-element operations with dot-operator “.”
- Specific operations for vectors und matrices:

<code>.’</code>	transpose	<code>det(x)</code>	determinant
<code>’</code>	conjugated trans.	<code>rank(x)</code>	rank
<code>inv(x)</code>	inversion	<code>eig(x)</code>	eigenvalue



MATLAB Basics

Structures and Cell Arrays

- Structures for data of different types: scalars, matrices, strings, etc.

- Fields of structure contain values

str = struct ('name_1',value_1,'name_2',value_2,...)

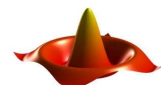
- Access to field values with “.”-operator: *str.name*
- Cell arrays: Multidimensional structure



MATLAB Basics

Administration of variables

- Dimension:
 - of a vector: `length(vektor)`
 - of a matrix: `size(matrix)`
- Anzeigen:
 - `who [variable]`: only variable names
 - `whos [variable]`: name, size, bytes and class
- Deletion:
 - of a variable: `clear variable`
 - of all variables: `clear, clear all`



MATLAB Basics

Relational operators

- Relational operators: `==`, `~=`, `<`, `<=`, `>`, `>=`,
- Test if variable x exists: `exist(x)`
- Relational operators can be used with scalars, vectors and matrices
- Results: Zero (logical false) if operation is not true, otherwise One (logical true)
- Operator precedence:
 1. Mathematical expressions
 2. Relational operators from left to right



MATLAB Basics

Logical operators

- Logical operators: `~` NOT, `&` AND, `|` OR, `xor` XOR
- Logical operators can be used with scalars, vectors and matrices
- Results: Zero is logical false, all other values are logical true. Results are always 0 or 1.
- Operator precedence:
 1. Mathematical expressions
 2. NOT
 3. AND and OR from left to right



MATLAB Basics

Further operators und logical indexing

- “Shortcut“ operators: `&&` (AND) und `||` (OR)

Stops if result is fully determined: `(1 || expression)`
expression is not evaluated

- Any element nonzero: `any(vektor)`
- All elements nonzero: `all(vektor)`
- Indices of nonzero elements: `find(vektor)`
- Logical indexing: `a(a<0) = 1`



MATLAB Basics

Conditional Control

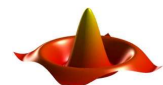
IF–statement

```
if logical expression
    statements
elseif logical expression
    statements
else
    statements
end
```

SWITCH–statement

```
switch expression (scalar or string)
    case value1
        statements
    case {value2 value3 ...}
        statements
    otherwise
        statements
end
```

- Separate statements and commands with “,”, “;”
- Nesting of if and switch possible



MATLAB Basics

Loop Control

- FOR: for *variable* = *expression* for n = 1:1:10 ,
 statements f = n^2,
 end end
- WHILE: while *expression* while w > 1 ,
 statements w = w + 1 ;
 end end
- Loop: – Next iteration: continue
 – Termination: break
 – Termination: return



MATLAB Basics

MATLAB–Scripts

- M–File: Commands in ASCII–File with extension `.m`
- Comments: `%` Comment
`%{ ... %}` Comment in several lines
`%%` Comments as cell-divider
- Continue command in next line: `...`
- Show commands and comments: `echo on`
- Controlled paged output: `more on`
- Display contents of file: `type file`



MATLAB Basics

MATLAB-functions I

- Specific sort of M-files:
 - Input arguments
 - Output of values
 - Local variables
- Definition: `function [var] = functionname (par)`
- Internal indicators: `nargin, nargout`
- Abort function with message: `error('info')`
- Nested functions possible
- Comments in functions: `% Comment`



MATLAB Basics

MATLAB-functions II

- **Local variables** in function are deleted after function call
- **Static variables:** `persistent var`
Keep their value after function call
- **Global variables:** `global var`
Are to be defined as global in function **and** workspace
and are available to all functions declaring it global

Delete with: `clear global`



MATLAB Basics

MATLAB-functions III

- **Inline Functions:** $f = \text{inline}(\text{funktion}, \text{var})$
- **Pseudo-Code:** $\text{pcode}(\text{funktion})$
- **Delete all functions:** clear functions
- **Function Handle:** $f_handle = @\text{funktion}$

Information: $\text{functions}(f_handle)$

Call with: $[y1, \dots, ym] = \text{feval}(f_handle, x1, \dots, xn)$

