Introduction to MATLAB

Numerical Mathematics
Prof. Dr. Ute Bauer-Wersing

Computer Science

Overview

What is MATLAB ?

MATLAB IDE

scalars, vectors and matrices

What is MATLAB ?

- ✓ **MATLAB** = **MAT**rix **LAB**oratory
- ✓ Programming Language / Integrated Development Environment
- ✓ very popular in industry **and** academia
- ✓ Strengths of MATLAB are
  - rich numerical algorithms of all kinds
  - visualization
  - rapid prototyping (data modeling and simulation)
  - analyzing and plotting of all kinds of data
  - available for all major OS (UNIX, Windows)
- ✓ various additional toolboxes http://www.mathworks.com/
- ✓ Commercial, proprietary software product (MathWorks, Inc.)
- ✓ [Public Domain Clone „Octave“]

MATLAB IDE

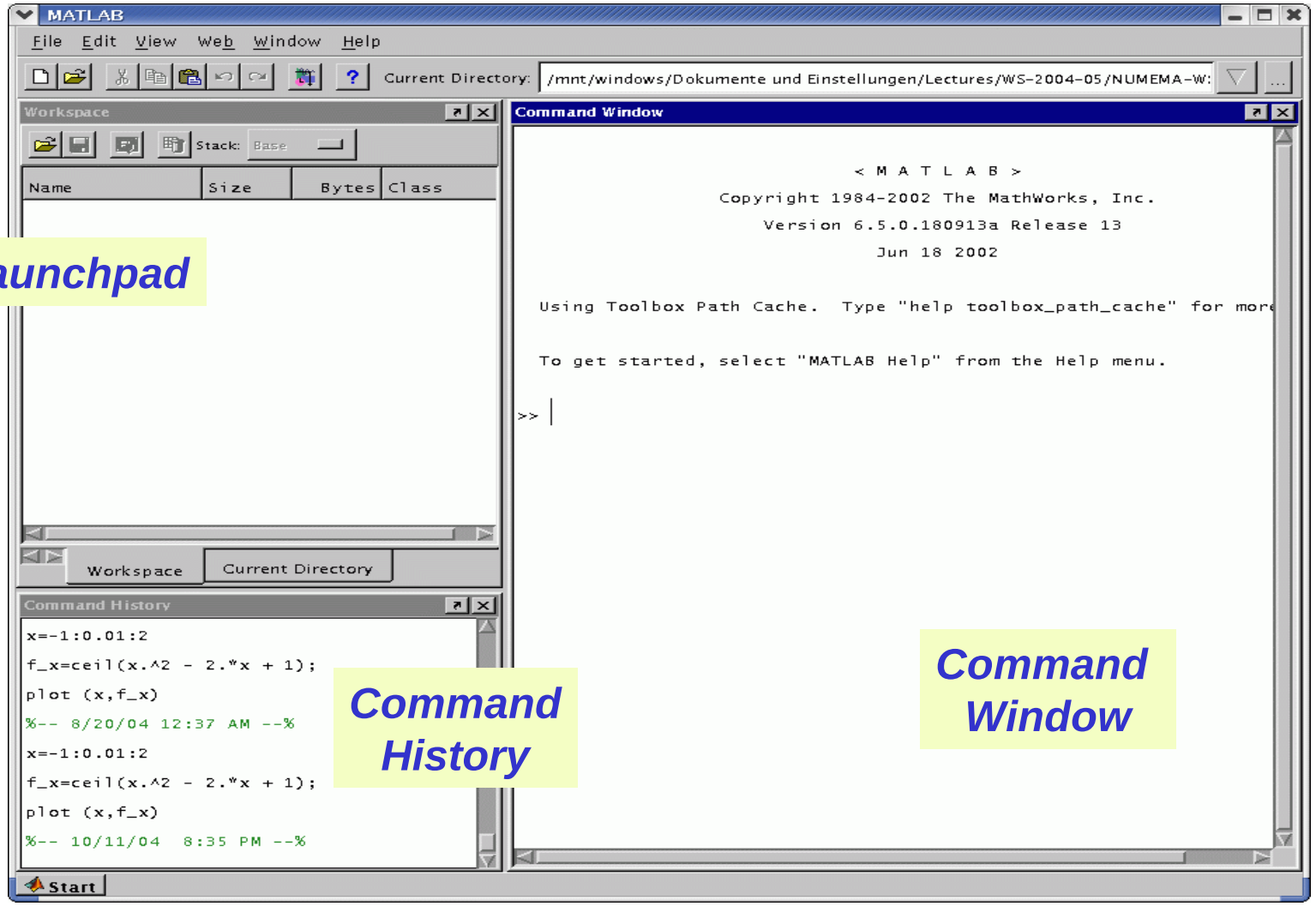
✓ Getting started

... on Microsoft Windows

1. **Start**
   - ✓ Double click on the MATLAB Icon (or XP Start Menu)

2. **Quit**
   - ✓ Menue item File → Exit
   - ✓ Strg Q
   - ✓ Enter „quit“ or „exit“ command

MATLAB IDE


MATLAB
File Edit View Web Window Help
Current Directory: /mnt/windows/Dokumente und Einstellungen/Lectures/WS-2004-05/NUMEMA-W:
Workspace
Stack: Base
Name Size Bytes Class
Workspace
Current Directory
Command History
x=-1:0.01:2
f_x=ceil(x.^2 - 2.*x + 1);
plot (x,f_x)
%-- 8/20/04 12:37 AM --%
x=-1:0.01:2
f_x=ceil(x.^2 - 2.*x + 1);
plot (x,f_x)
%-- 10/11/04 8:35 PM --%
Command Window
< M A T L A B >
Copyright 1984-2002 The MathWorks, Inc.
Version 6.5.0.180913a Release 13
Jun 18 2002
Using Toolbox Path Cache. Type "help toolbox_path_cache" for more
To get started, select "MATLAB Help" from the Help menu.
>>
Start
Launchpad
Command History
Command Window

MATLAB IDE

MATLAB IDE is separated into ...

1. **Launchpad**
   - ✓ **Workspace**: shows variables and data types
   - ✓ **Current Directory**: Directory Browser ... Choose Files

2. **Command History**
   - ✓ History of recent commands
   - ✓ Commands are available in the next session
   - ✓ Copy-and-paste

3. **Command Window**
   - ✓ The area where you can enter your MATLAB commands
   - ✓ MATLAB is an interpreted language (in the first place)
   - ✓ Enter commands and function calls after the prompt >>
   - ✓ Execution of M-files („Ending of MATLAB-Sources .m)

MATLAB IDE

... what else matters?

1. **Help**
   - ✓ Helpdesk: extensive help files (searchable)
   - ✓ to get helpfull explanations for individual functions:

```
help <function|command>
```

2. **M-File Editor**
   - ✓ Edit/save MATLAB Scripts
   - ✓ Open the editor : **File → New → m-file**
   - ✓ Save
   - ✓ Execute **Debug → run**

Skalars, Vectors and Matrices

... Number formats

- ✓ Signed `+/-`
- ✓ exponential format `10e-5`

... Variables

- ✓ Names: max. 31 letters
- ✓ Case-Sensitive
- ✓ Examples `MyVar = 10; my_var=10;`
- ✓ Delete a variable `clear MyVar;`
- ✓ No special data types

... Comments

- ✓ Start with % `MyVar = 10; % comment`

Scalars, Vectors and Matrices

... **THE** data type in MATLAB are **matrices**

- ✓ i.e. (indexed) multi-dimensional arrays
- ✓ typical example: two-dimensional matrices n x m
- ✓ vectors are one-dimensional matrices
  - row vector 1 x n matrix
  - column vector n x 1
- ✓ scalars are 1 x 1 matrices
- ✓ The empty matrix is defined by [ ]

Skalars, Vectors and Matrices

... some Examples

```
a = 3            %  1 x 1 matrix

a = [1,2,3]      %  row vector

a = [1;2;3]      %  column vector

A = [1,1,1;      %  3 x 3 matrix
     2,2,2;
     3,3,3]

A = []           %  empty matrix
```

Skalars, Vectors and Matrices

... special **functions to initialize matrices**

MATLAB	Funktion
`ones(n, m)`	n x m Matrix; elements intitialized to „1s“
`zeros(n, m)`	n x m Matrix, elements intitialized to „0“s
`eye(n,m)`	n x m Identity Matrix, Diagonal elements are intitialized to „1“; Off-diagonals intitialized to „0“
`rand(n,m)`	n x m Matrix; the elements are initialized randomly using uniformly distributed values from the interall [0,1]
`randn(n,m)`	n x m Matrix; the elements are initialized randomly using normally distributed values *N*(0,1)

Skalars, Vectors and Matrices

... Examples

```
A = zeros(2,3)          A = 0 0 0
                            0 0 0

A = eye(3,3)            A = 1 0 0
                            0 1 0
                            0 0 1

A = rand(2)             A = 0.5445 0.2335
                            0.4998 0.8711
```

Skalars, Vectors and Matrices

... important functions

- ✓ Size of a matrix: **`size(A)`**
  Output: first number = `rows`, 2nd number =`columns`

- ✓ Length of a vector : **`length(a)`**

- ✓ Selecting the matrix element a_{ij}
  `A(i,j)`

```
e.g.  A(2,3)          % Element 2. row
                              3. column
```

- ✓ Index starts with 1 ... n

Skalars, Vectors and Matrices

... The <:> Operator is used as increment/decrement operator to construct vectors !

- ✓ Example:
```
n= 1:10    %  result is row vector
           % default increment is 1
n = 1 2 3 4 5 6 7 8 9 10
```

- ✓ Example:
```
k= 1:2:9   % result is row vector
           % increment is 2
k = 1  3  5  7  9
```

- ✓ Loop variable
```
for n=1:10
      % do something
end
```

Skalars, Vectors and Matrices

... the <:> Operator can be used to access vectors and matrices

- ✓ Example: `A(1:10,3)  % select the 3. column of`
  `% rows 1-10 of A`
- ✓ Example: `A(1,: )  % select the 1. row of A`
- ✓ Example: `A(:,:)  % the entire matrix A`

... Preparation

... Log onto your computer (Windows)

... in the E-learning module you'll find the following files

Lab-1

- **Introduction-1**
- **Files-1 (Archive)**
- **Lab-1A**

... Start with the introduction to this lab session