A short introduction to MATLAB

jean-marie.furbringer@epfl.ch

Crash course

- First edition: not sure about the time that it will take and how far we will go

- It is a base to be developed by yourself

- So:

  - Ask questions!

  - Try things

  - Read the help

Program

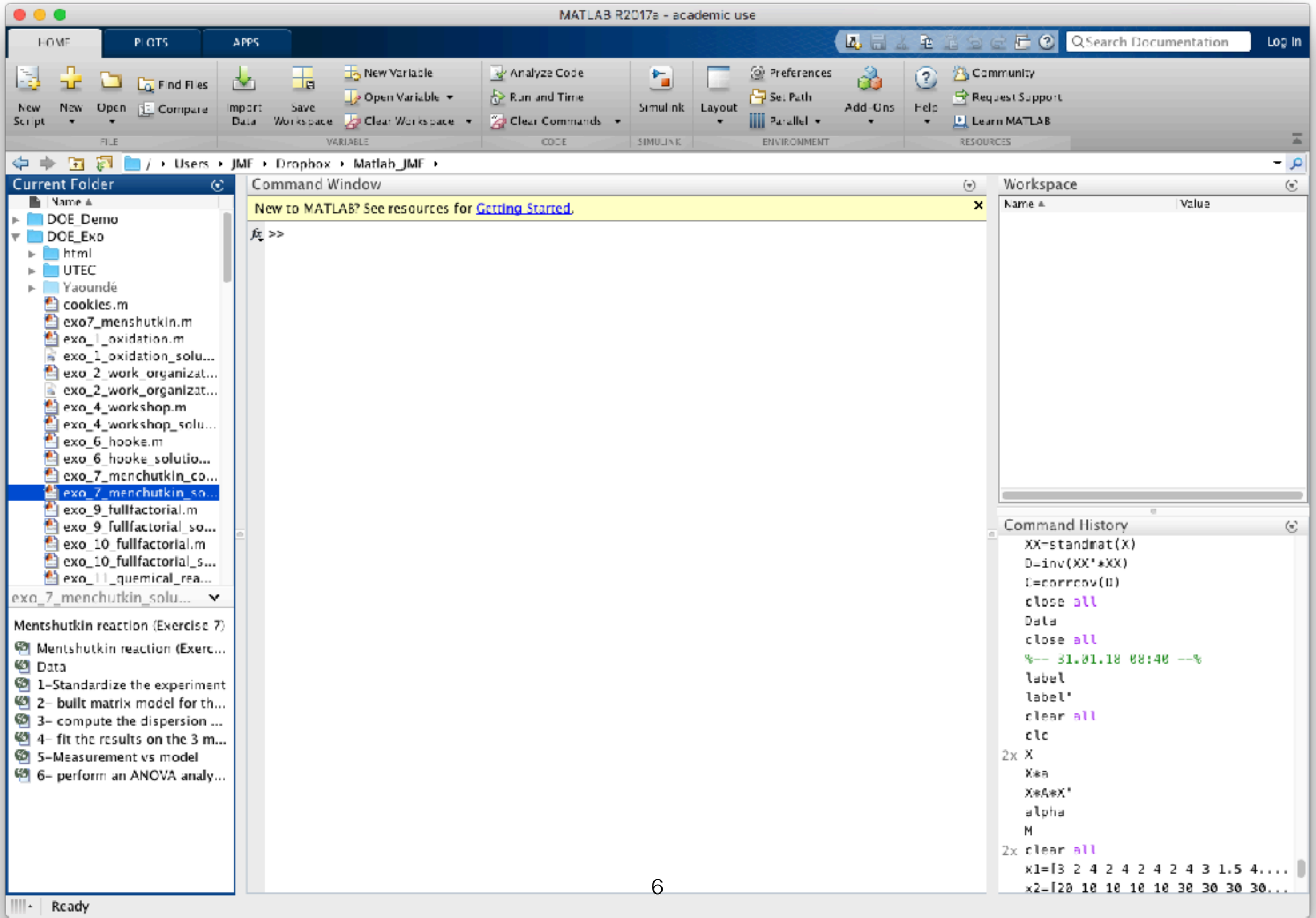
1. **Base**: Monday a.m.
2. **Live editor, batches and functions**: Monday p.m.
3. **Data class**: Monday p.m.
5. **Graphics**: Tuesday a.m.
6. **Symbolic calculation**: Tuesday p.m.

Référence

- **Matlab for Engineers**, Holly Moore, Pearson, 2015, isbn 978-0-133-48597-4
- **Introduction to MATLAB for engineers and scientists**, Sandeep Nagar, Apress, 2017, isbn 978-1-4842-3188-3

- Who are you?
- What is your previous experience with Matlab?

1.1 User interface



1.2 Elements of the interface

- Folders and path
- Command window
- Workspace
- Current folder

1.3 Calculator

- 3+5
- 6-10
- 3.5*4
- 2/3
- format long
- 2/3
- format short
- 2/3

- sin(pi/6)
- exp(3)
- atan(10)/pi
- log10(10)
- log(10)

1.4 Command:`Format style`

Possible styles

- short / long : fix-decimal format 4 or 15 digits
- shortE / longE : scientific notation 4 or 15 digits
- shortEng / longEng : engineering notation 4 or 15 digits
- shortG / longG: fixed-decimal format or scientific notation, whichever is more compact
- bank : two digits after the decimal point
- + : positive/negative format
- hex : hexadecimal
- rat : rational

> Test the different formats for
> A=[1, 4.56565656, 1/3, 1.56e-5]

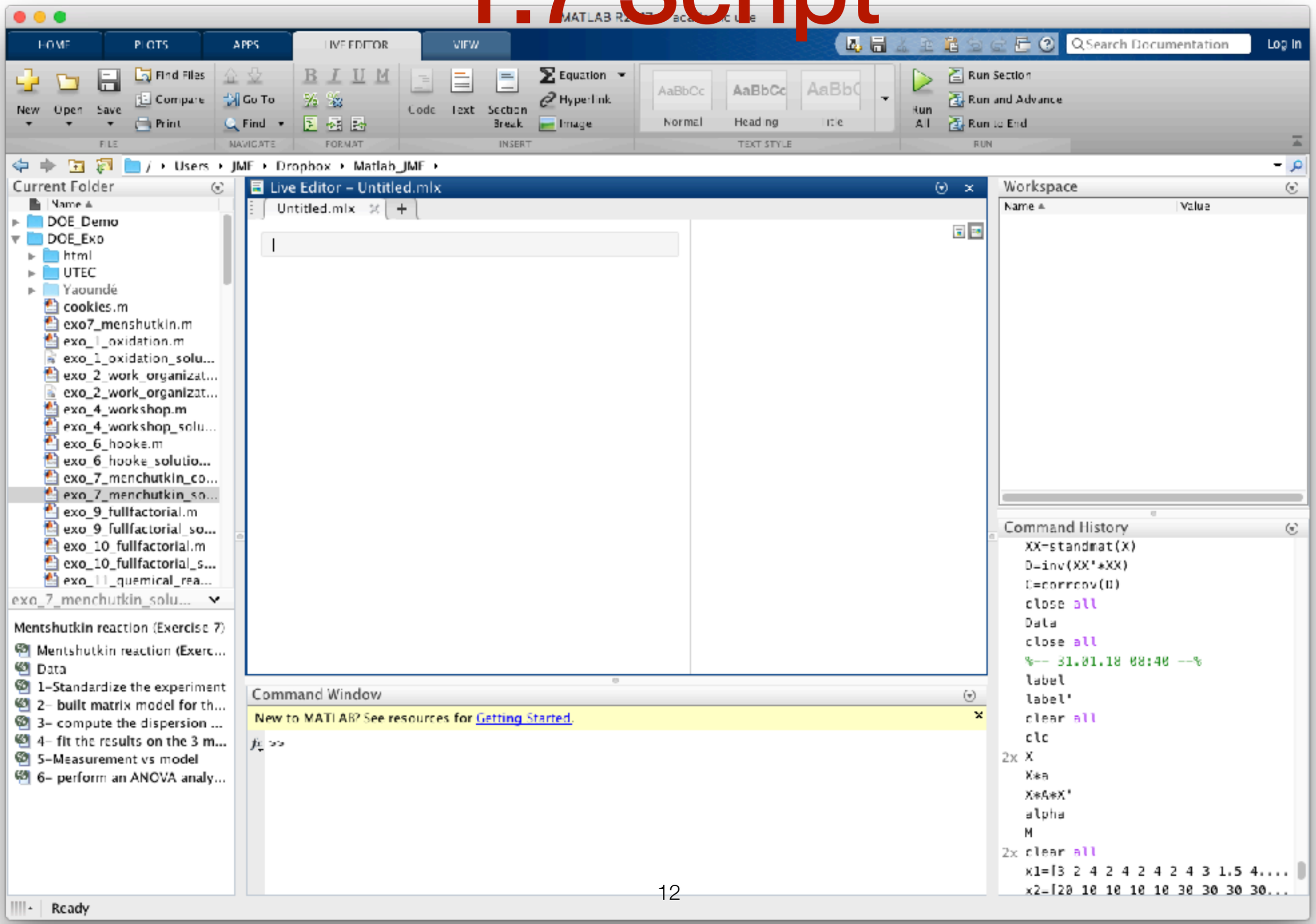
1.5 REPL

- Reads
- Evaluates
- Prints
- Loops back

1.6 Self learning and getting help

- On line help « `help format` »
- Menu: help/documentation

1.7 Script


HOME
PLOTS
APPS
LIVE EDITOR
VIEW
Search Documentation
Log In
New
Open
Save
Find Files
Compare
Print
Go To
Find
Code
Text
Section Break
Equation
Hyperlink
Image
Normal
Run
Run Section
Run and Advance
Run to End
FILE
NAVIGATE
FORMAT
INSERT
TEXT STYLE
RUN
/ ▸ Users ▸ JMF ▸ Dropbox ▸ Matlab_JMF ▸
Current Folder
Name
DCE_Demo
DCE_Exo
html
UTEC
Yaoundé
cookies.m
exo7_menshutkin.m
exo_1_oxidation.m
exo_1_oxidation_solu...
exo_2_work_organizat...
exo_2_work_organizat...
exo_4_workshop.m
exo_4_workshop_solu...
exo_6_hooke.m
exo_6_hooke_solutio...
exo_7_menchutkin_co...
exo_7_menchutkin_so...
exo_9_fullfactorial.m
exo_9_fullfactorial_so...
exo_10_fullfactorial.m
exo_10_fullfactorial_s...
exo_11_quemical_rea...
exo_7_menchutkin_solu...
Mentshutkin reaction (Exercise 7)
Mentshutkin reaction (Exerc...
Data
1-Standardize the experiment
2- built matrix model for th...
3- compute the dispersion ...
4- fit the results on the 3 m...
5-Measurement vs model
6- perform an ANOVA analy...
Live Editor - Untitled.mlx
Untitled.mlx
Workspace
Name
Value
Command Window
New to MATLAB? See resources for Getting Started.
>>
Command History
XX=standmat(X)
D=inv(XX'*XX)
C=corrcov(D)
close all
Data
close all
%-- 31.01.18 08:40 --%
label
label'
clear all
clc
2x X
X*a
X*A*X'
alpha
M
2x clear all
x1=[3 2 4 2 4 2 4 2 4 3 1.5 4....
x2=[20 10 10 10 10 30 30 30 30...
Ready

1.8 Variables

- **Data types:** logical, char, int8, int16, int32, int64, single, double,…

- **Naming**
  - Names should not start with a number; however, numbers can be used anywhere afterward.
  - Variable names are case sensitive.
  - *Keywords* cannot be used as names.
  - Names can include underscores (_).

1.8 Variables

- **Data types:** logical, char, int8, int16, int32, int64, single, double,…

- **Try**

`a=123456789012345` with short and long format

```
b=int8(123456789012345)

c=int16(123456789012345)

d=int32(123456789012345)

f=int16(32768)
```

check attentively what you get back

1.9 Clear - Save

- `clear variable_name`
- `clear all`
- `clc`

- `save(file_name var_name)`
- `save(file_name var_name, fmt)`
- `save(file_name)`

Format:

- `'-mat'`: binary
- `'-acsii','-tabs'`:Tab-delimited text format with 8 digits of precision

1.10 Array based computing

- An horizontal vector: `A=[ 123 45.5 301 ]` or equivalentely `A=[ 123, 45.5, 301 ]`

- A vertical vector: `B=[10;15;20]` equivalent to `B=[10,15,20]'`

- The prime sign « ‘ » means « transpose »

- Regular array `C=0:0.25:5`

- 2D arrays `D=[1 5;5 -2]`

- Useful fonctions `ones(Ni,Nj)`, `zeros(Ni,Nj)`, `repmat(D,2,3)`, `eye(N)`

1.10 Array based computing

- Linear combinations of arrays of the same size and same classs `C=A+B`

- Matrix multiplication `D=A*B`

- Inversion `D=A/B` (eq. to `D=A*inv(B)`) or `D=A\B` (eq. to `D=inv(A) * B`)

- Operations element by element: .*, ./ ; .^

Try these manipulations

- `A=[ones(3),zeros(3,1);ones(1,4)]`
- `B=rand(4)`
- `C=A.*B`
- `D=eye(4)/C`
- `E=D*C`
- `E=int8(E)`
- `Size(E)`

1.11 Built in functions

- find()
- sort(A, ‘descent’)
- sortrows()
- trace(A)
- reshape()

1.12 Using indices

- A=ones(4)
- A(1,1)=0
- A(1:2,1:2)=zeros(2)

1.13 Publish as latex, html or pdf

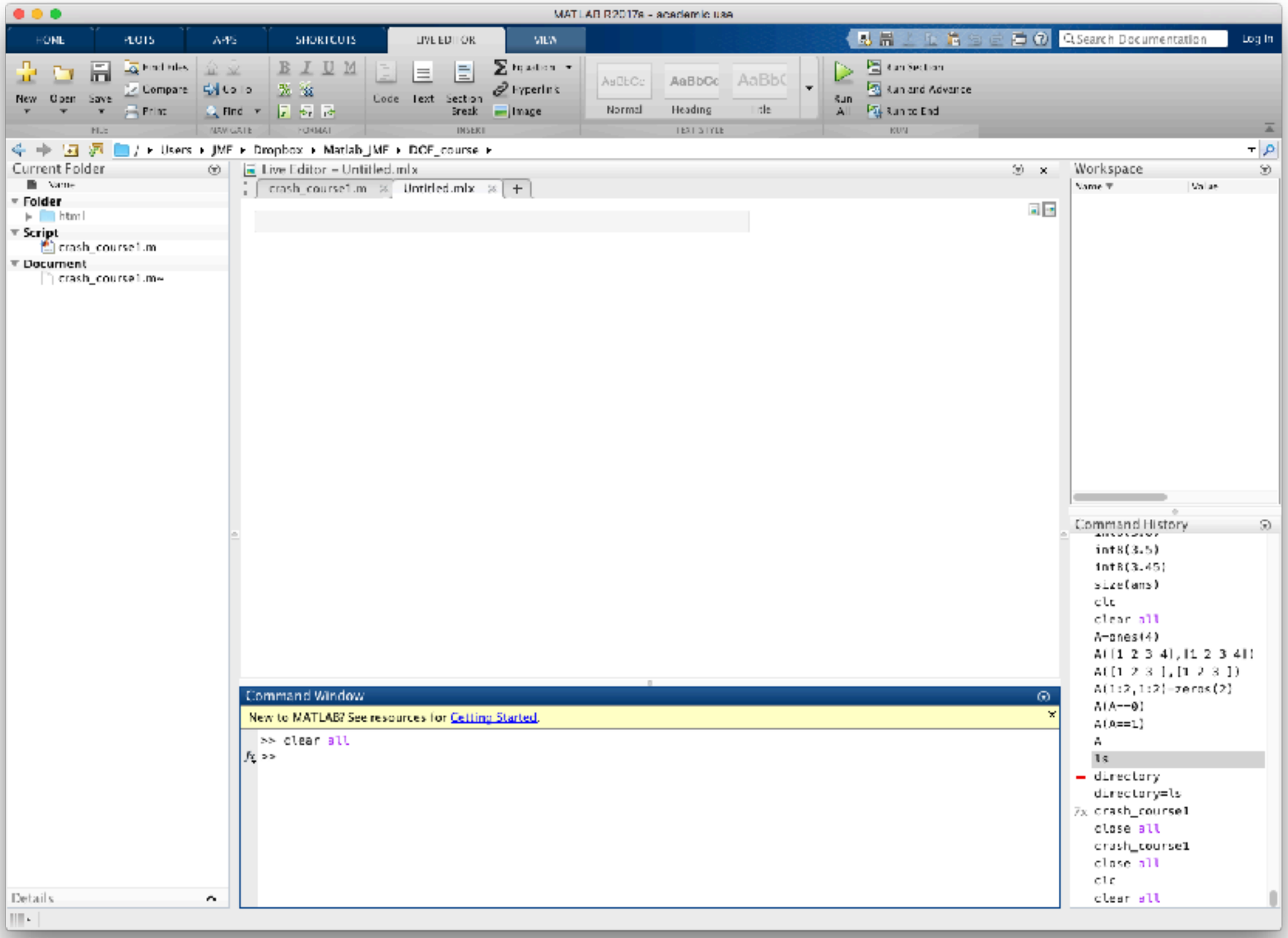
- Menu publish

- Editing publishing options

Program

1. **Base**: Monday a.m.
2. **Live editor, batches and functions**: Monday p.m.
3. **Data class**: Monday p.m.
5. **Graphics**: Tuesday a.m.
6. **Symbolic calculation**: Tuesday p.m.

2.1 Live script



2.2 User built functions

```
function  [out1,out2]=func_name(in1,in2, …)
    commands
end
```

- `nargin`: number of input arguments
- `nargout`: number of output arguments
- by default variables within the function are local variables and then not known outside of the function
- `global x` : define x as a global variable

If several functions all declare a particular variable name as `global`, then they all share a single copy of that variable.

2.3 standard loops

- if condition, execution, end
- if condition, execution, else, execution end
- if condition, execution, else if condition, execution end
- for k=1:10, execution, end
- while k>limit, execution, end
- switch case_list, case 1 execution case 2 execution,… end

Program

1. **Base**: Monday a.m.
2. **Live editor, batches and functions**: Monday p.m.
3. **Data class**: Monday p.m.
4. **Graphics**: Tuesday a.m.
5. **Symbolic calculation**: Tuesday p.m.

3.1 Classes of data

- Array (vectors, matrices)
  - array of double `A=[1 3 5 8], B=eye(4)`
  - array of integer `Ai=int16([1 3 5 8])`
  - array of character `As='abcdef'`
- Cell array
  - `CellArrayName = cell(i,j)`
  - `CellArrayName={data1,data2,data3,…}`
- Structure
  - `StructName = struct(field,value)`
  - `StructName.FieldName=value`
  - access element with `StructName().FieldName()`

3.2 Table

- `T = table(var1,…,varN,Name,Value)`
- `T = readtable(filename,opts,Name,Value)`
- `T = array2table(A,Name,Value)`
- `T = struct2table(S,Name,Value)`
- `summary(MyTable)`
- `T.variable=categorical(T.variable)`
- `grpstats(MyTable,'Color',{'mean','std'})`

Program

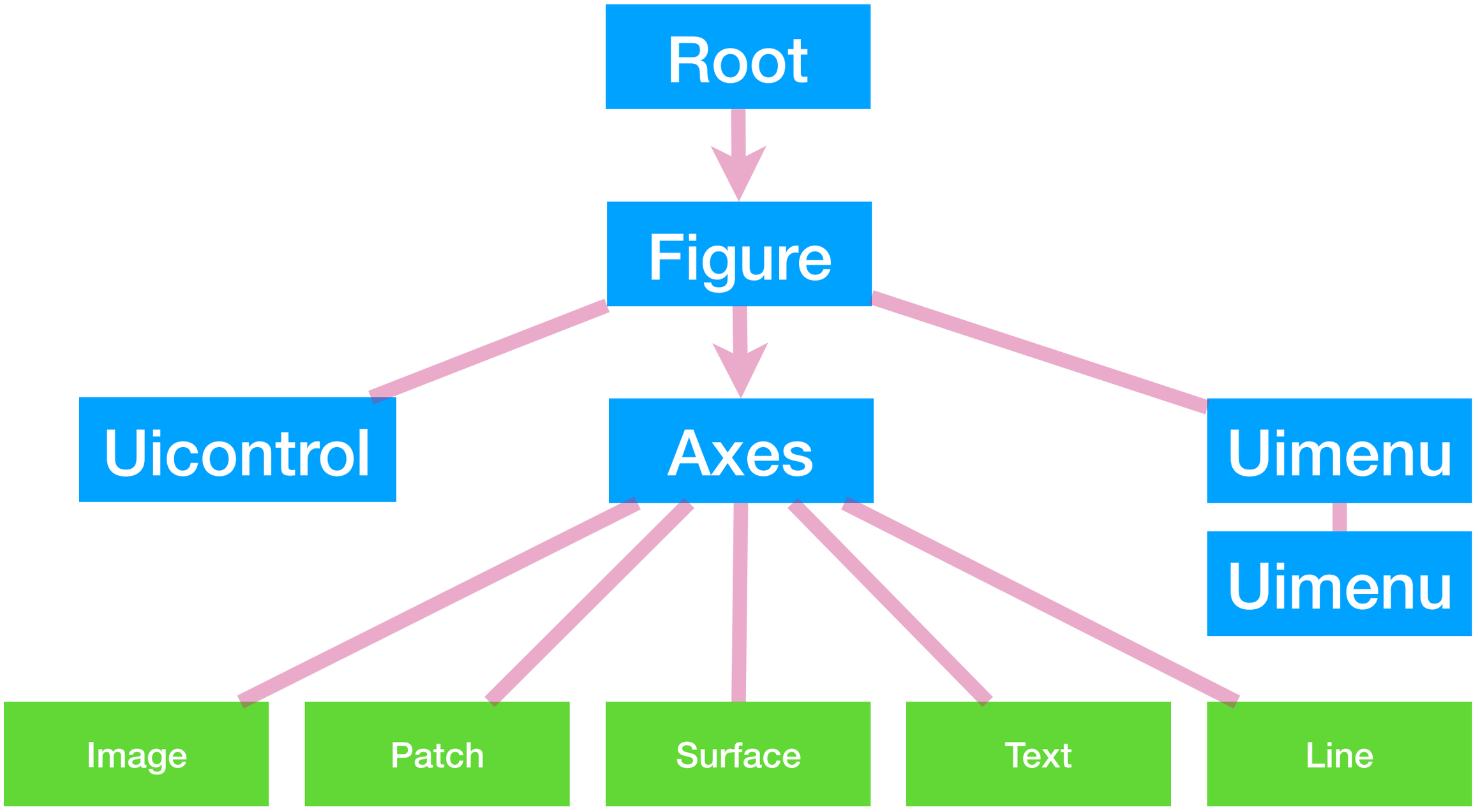
1. **Base**: Monday a.m.
2. **Live editor, batches and functions**: Monday p.m.
3. **Data class**: Monday p.m.
4. **Graphics**: Tuesday a.m.
5. **Symbolic calculation**: Tuesday p.m.

4.1 Figure window

```
figure
figure(1)
figure(Property,value)

Properties
- Name : figure('Name','Measured Data’)
- Number
- Color
- Position [left bottom width height]
 figure('Position',[400 500 1000 750])
- Units: pixels, normalized, centimeters,
points,…
```

4.2 Hierarchical model



4.3 Plot

- plot(x,y,LineSpec)
- plot(x1,y1,…,xn,yn)
- plot(__,Name, Value)

LineSpec

‘-‘ for hard line

‘- -‘ for dashed line

‘:’ for dotted line

o,*,.,+,x,s,d as markers

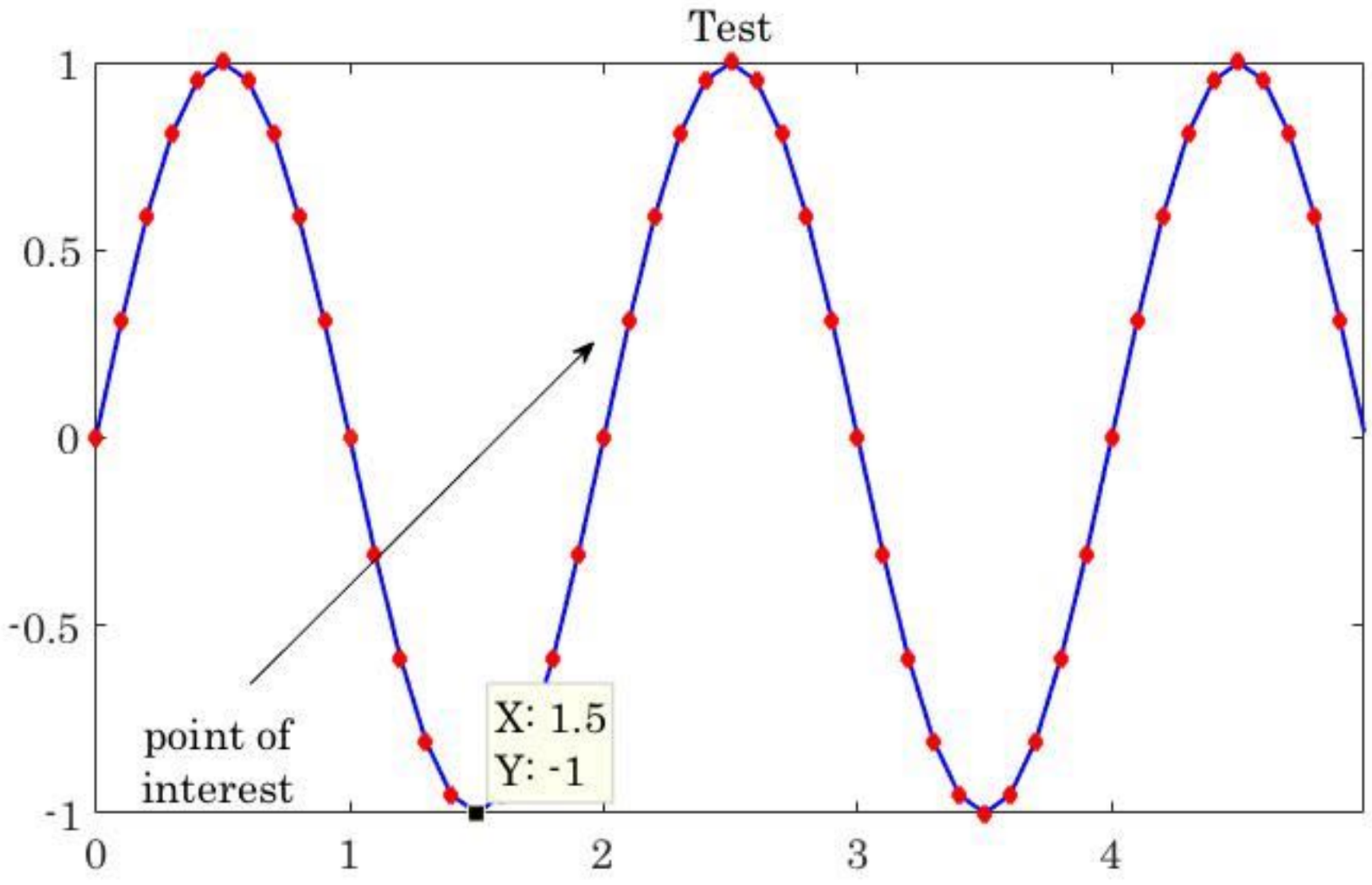
y, m, c, r, g, b, k as colors

combined: ‘r:0’

4.4 A few graphical functions and key words

- `axis([x1,x2,y1,y2,z1,z2])`
- `subplot(a,b,c)`
- `title(str)`
- `xlabel(str)`
- `hold on ….hold off`
- `box on`
- `grid on`
- `get(handle[,property])`
- `set(handle,property,value)`
- `text(x,y,str)`
- `annotation(LineType,x,y)`

4.5 Editing a figure



4.6 Editing the properties of a figure

- `figure`
- `t=0:5/100:5`
- `plot(t,sin(pi*t))`
- `get(gca)`
- `set(gca, property, value)`
- `set(gca,'XTick',1:5,'XTickLabel',{'A' 'B' 'C' 'D' 'E'})`
- `title('Distribution','FontName','Time new roman', 'FontSize', 16)`

4.7 Plot 3D

- plot3(X1,Y1,Z1,LineSpec,…)

4.8 Surf

- surf(Z)
- mesh(Z)
- surfc(Z)
- contour(Z)
- [X,Y]=meshgrid(x,y)
- surf(X,Y,Z,C)
- colorbar

Program

1. **Base**: Monday a.m.
2. **Live editor, batches and functions**: Monday p.m.
3. **Data class**: Monday p.m.
4. **Graphics**: Tuesday a.m.
5. **Symbolic calculation**: Tuesday p.m.