Introduction to MATLAB

EL1150, Lecture 1, HT 2012

What is Matlab?

- A software environment for interactive numerical computations

- Examples:
  - Matrix computations and linear algebra
  - Solving nonlinear equations
  - Numerical solution of differential equations
  - Mathematical optimization
  - Statistics and data analysis
  - Signal processing
  - Modelling of dynamical systems
  - Solving partial differential equations
  - Simulation of engineering systems

What will you learn in EL1150?

- Effective Matlab usage

  - Possibilities and limitations
  - Syntax and interactive computations
  - Matlab programming (using functions and script files)
  - Visualization
  - Optimization of code for efficient computations

How will you learn in EL1150?

- By solving problems by your own
- Practice, practice, practice
- Teaching assistant available for questions

Why should you attend EL1150?

- Matlab used (on a daily basis) in many engineering companies

Why should you attend EL1150?

- Matlab used in many courses at KTH

Numerical analysis
Signal and Systems I
Signals and Systems II
Modeling of Dynamical Systems
Automatic Control, Basic Course
Automatic Control, Advanced Course
Nonlinear Control
Hybrid and Embedded Control Systems

Chemical Process Control
Control Project Course
Signal Theory
Digital Signal Processing
Adaptive Signal Processing
Signal Processing Project
Communication Theory
Advanced Communication Theory

and many, many more...

Today’s Lecture

- Course information
  - Course contents and literature
  - Course web
- The Matlab Programming Environment
  - Background to Matlab
  - Interactive calculations
  - Vectors and matrices
  - Graphical illustrations
  - Matlab programming
  - Tutorials and help
- The exam
  - Bilda

EL1150 – Introduction to Matlab

- Student Handbook:
  - 1.5 credit self study course.

- Objectives:
  - Gain basic knowledge of Matlab programming
  - To prepare for other courses where Matlab is used
  - To give insight into a state-of-the-art tool for technical computation and visualization

Course Literature

- N. Bergman and F. Gustafsson, ”*Matlab for Engineers Explained*”, Springer, 2003
  - Available via student book store (”kårbokhandeln”)
  - Adlibris (1-2 days delivery), 365:-
  - Teaches practical Matlab usage (not a full manual)
  - Basic description of theoretical concepts
  - Based on examples with guided tours of the system
  - Exercises with solutions
  - Applications from engineering courses

- Course covers first 60 pages
- Suggested exercises:

  1-5, 8-17, 21, 23-32, 34, 37, 40-41, 44, 47-48

Additional course material

- Exercise compendium on course webpage (free)

- Matlab help and documentation

```
>> doc
>> help
>> demo
```

- Tutorials on the web

Course webpage

https://www.kth.se/social/course/EL1150/

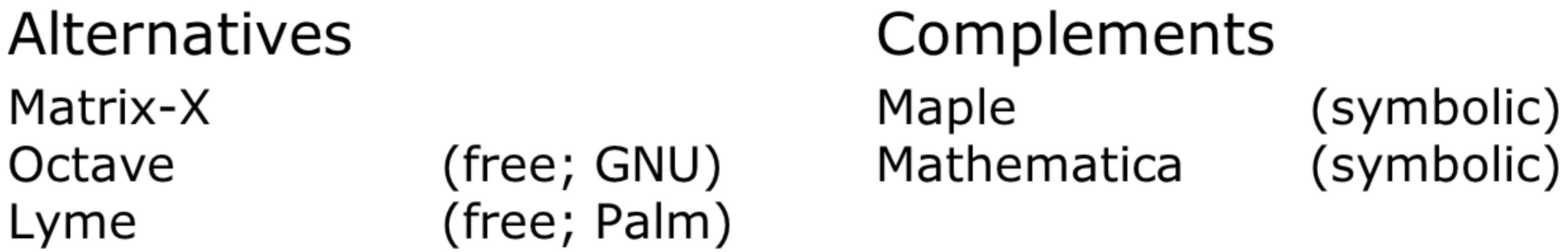
Studies

- Self studies, guided tours
- Supervised computer sessions
- Questions via e-mail <zhenhua.zou@ee.kth.se>
- Examination in Bilda (bilda.kth.se)
  - More info later

Part II – Matlab Basics

Matlab Background

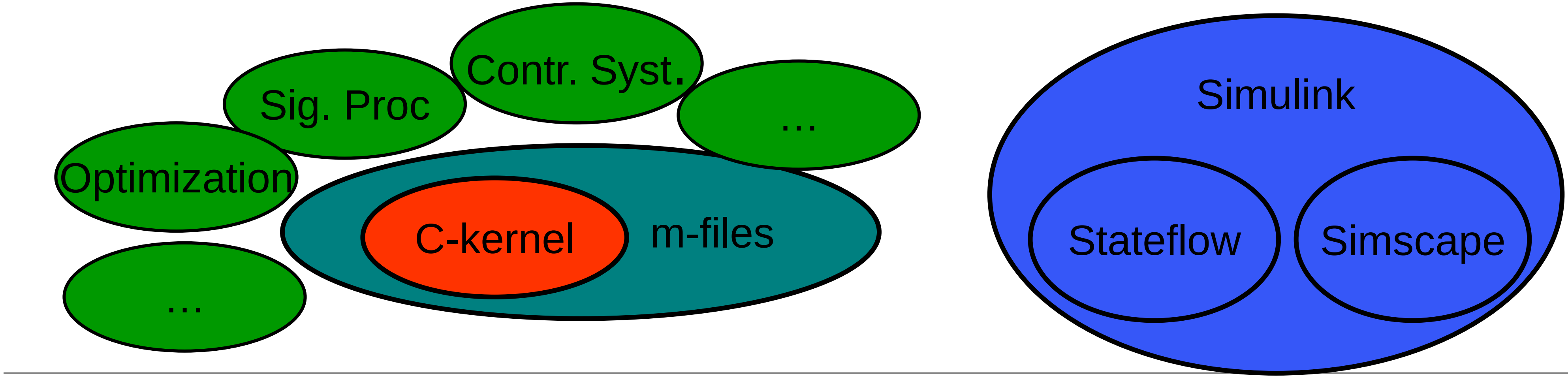
- **Matlab** = **Mat**rix **Lab**oratory
- Originally a user interface for numerical linear algebra routines (Lapak/Linpak)
- Commercialized 1984 by The Mathworks
- Since then heavily extended (defacto-standard)



Alternatives		Complements	
Matrix-X		Maple	(symbolic)
Octave	(free; GNU)	Mathematica	(symbolic)
Lyme	(free; Palm)		

Construction

- Core functionality: compiled C-routines
- Most functionality is given as m-files, grouped into toolboxes
  - m-files contain source code, can be copied and altered
  - m-files are platform independent (PC, Unix/Linux, MAC)
- Simulation of dynamical systems is performed in Simulink



Matlab session

- Interactive calculations
- Variable and memory management
- Vectors and matrices
- Matrix operators
- Matrix functions
- Indexing matrices
- Linear algebra
- Graphics
- Scripts and functions

- **The help/demo system**

Memory pre-allocation

- Programming style has huge influence on program speed! Memory allocation takes a lot of time: Pre-allocate memory!

slow.m

```
clear all
x=-1e4:1e4;
for ii=1:length(x)
  if x(ii)>=0,
    s(ii)=sqrt(x(ii));
  else
    s(ii)=0;
  end;
end;
```

fast.m

```
clear all
x=-1e4:1e4;
s=sqrt(x);
s(x<0)=0;
```

- Use **profile** to find code bottlenecks!

The exam

- The exam file is downloaded from Bilda
- The exam is done on your own computer during 72 hours (24 Sep - 14 Oct)
- The exam is corrected on your computer
- A code is generated that you upload in Bilda
- You get your result direct

- Four problems drawn from different categories

Example of exam problem

```
-----------------------------------------------------------------------
Sort numbers in descending order!
-----------------------------------------------------------------------
Write a function that creates a column vector s, that contains the elements
of a vector x sorted in descending order (from largest to smallest).

Syntax: s = dsort(x)
------------
where s is the output vector and x is the input vector.

Basic functionality:

>> x = [1 5 2 3 9 4]

x =
    1   5   2   3   9   4

>> s=dsort(x)
s =
     9
     5
     4
     3
     2
     1
-----------------------------------------------------------------------
```