Computational Photonics
Seminar 01, 13 April 2018

What is MATLAB?

• Tool for technical computing
• Integrated development environment for computation, visualization and programming
• At the same time higher level programming language and an interactive tool for numerical work
• Uses an interpreter + just-in-time compilation, many native libraries – High-Performance Computing?
Comparing MATLAB and C

simple example of a matrix multiplication

\[ A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix}, \quad C = AB = \begin{pmatrix} 30 & 24 & 18 \\ 84 & 69 & 54 \\ 138 & 114 & 90 \end{pmatrix} \]
MATLAB

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

>> B = [9 8 7; 6 5 4; 3 2 1]
B =
    9     8     7
    6     5     4
    3     2     1

>> C = A*B
C =
    30    24    18
    84    69    54
   138   114    90
>>
```
```c
#include "stdio.h"

int main(int argc, char* argv[]) {
    double A[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    double B[3][3] = {{9,8,7},{6,5,4},{3,2,1}};
    double C[3][3];

    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            C[i][j] = 0.0;
            for (k=0; k<3; k++) {
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
            }
        }
    }

    printf("C=
");
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            printf("%lf ", C[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```
$ cc -o matmul matmul.c
$ ./matmul
C =
   30.000000  24.000000  18.000000
   84.000000  69.000000  54.000000
  138.000000 114.000000  90.000000
$
Typical applications of MATLAB

• Mathematical and numerical work
• Development of mathematical algorithms
• Simulation
• Data acquisition, -analysis, -visualization
• Development and implementation of (stand-alone) applications with graphical user interfaces
• In industry and academia for research and teaching purpose
A brief history

The name MATLAB stands for matrix laboratory.

**Formerly:**
- Only interface to allow a simplified access on all the tools that have been implemented within the projects LINPACK and EISPACK, their purpose was matrix operations

**Today:**
- Includes modern matrix program libraries LAPACK and BLAS
- TOOLBOXES: extensive collection of MATLAB-functions (M-Files) for special problems and purposes, e.g. signal processing, neural networks, Fuzzy-logic, Wavelets, partial differential equations, image processing, statistics, symbolic operations
The MATLAB system

1. Integrated Development Environment
   Environment for the efficient use of the entire functionality as provided by MATLAB (Command-Window, Editor, Debugger, Help-Browser, ...)

2. Library of mathematical functions
   All kinds of predefined mathematical functions (sum, sine, cosine, ..., matrix inversion, matrix-eigenvalues, Fourier-transformation)

3. Language MATLAB
   Allows to call all the mathematical functions, provides a sufficiently large set of data structures and primitives for controlling the flow of programs (conditions, loops,..)
The MATLAB system

4. Graphics
tools for the visualization of vectors and matrices, 2D, 3D, animation, image processing, graphical user interfaces

5. Application Programming Interface (API)
provides an interface to other programming languages like C and Fortran (dynamic linking, computational engine)
MATLAB help and primers

• The MATLAB help (pressing F1 for any unknown function or typing doc ... or help ...) is extremely helpful and comprehensive (also online at mathworks.com)

• The systems consists of thousands of functions in dozens of toolboxes. For absolute beginners, reading a primer might be helpful. A google search for “MATLAB primer” or a visit in the library should help.
MATLAB open source alternatives

- **GNU Octave:**
  - Core language mostly compatible to Matlab
  - Less comfortable IDE
  - No compatible versions of most Matlab toolboxes (but alternative packages on [Octave Forge](https://octave.sourceforge.net/))

- **Scilab:** Similar to Matlab but not compatible

- **Scientific Python Stack (Python, NumPy, SciPy, Matplotlib, ...):**
  - Completely different programming language but comparable functionality
  - Very large community
  - Specialized packages for almost any scientific domain
  - Common distributions:
    - [Anaconda](https://www.anaconda.com)
    - [Python(x,y)](https://python.org)
    - [WinPython](https://winpython.github.io)
    - [Enthought Canopy](https://www.enthought.com)
Now, you are given a refreshment crash-course in MATLAB in the seminar.
Programming task 1

Plot the sinc-function which is defined as $\frac{\sin(x)}{x}$ within the interval of $[-3\pi, +3\pi]$ using 301 points. Try to avoid the singularity which occurs at $x = 0$ in an appropriate manner!

Necessary functions:

- colon operator: 
- division element by element: ./
- sine: sin
- function for plotting a line graph: plot
Programming task 2

Write down and test a program to compute the sum of all the elements of a matrix.

Header of the program:

```matlab
function sm = supersum(A)
    % Sum over all elements of a matrix
    % Command sm = supersum(A)
    % A: arbitrary 2D-matrix
```

Commands you will need: sum, (size)

Test with: `A = magic(9);`

Compute the sum of all the elements of a magic square.