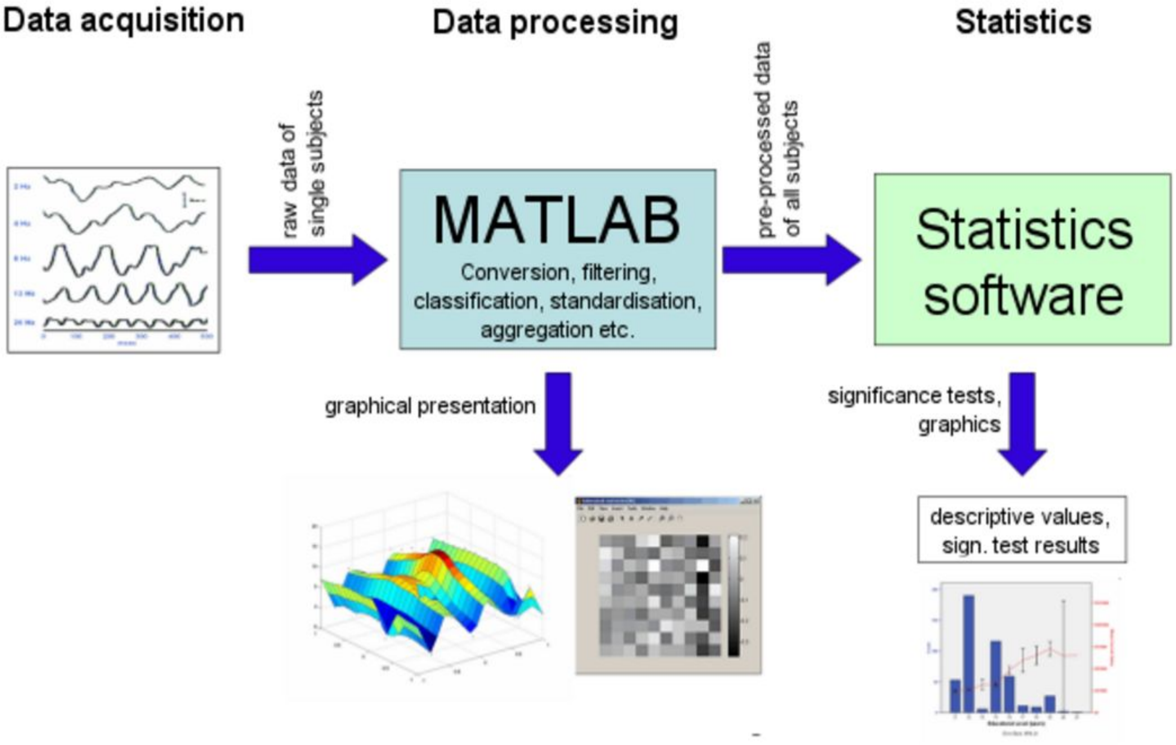
MATLAB®

DSAA Tutorial Session 1

Introduction

- MATLAB, which stands for MATrix LABoratory, is a state-of-the-art mathematical software package, that is used extensively in both academia and industry. It is an interactive program for numerical computation and data visualization, that along with its programming capabilities provides a very useful tool for almost all areas of science and engineering.
- MATLAB deals mainly with matrices. A scalar value is a 1- by-1 matrix and a row vector of length say 5, is a 1-by-5 matrix.
- Documentation at https://in.mathworks.com/help/matlab/



Data acquisition

Data processing

Statistics

raw data of single subjects

MATLAB

Conversion, filtering, classification, standardisation, aggregation etc.

pre-processed data of all subjects

Statistics software

graphical presentation

significance tests, graphics

descriptive values, sign. test results

Exemplary data processing workflow

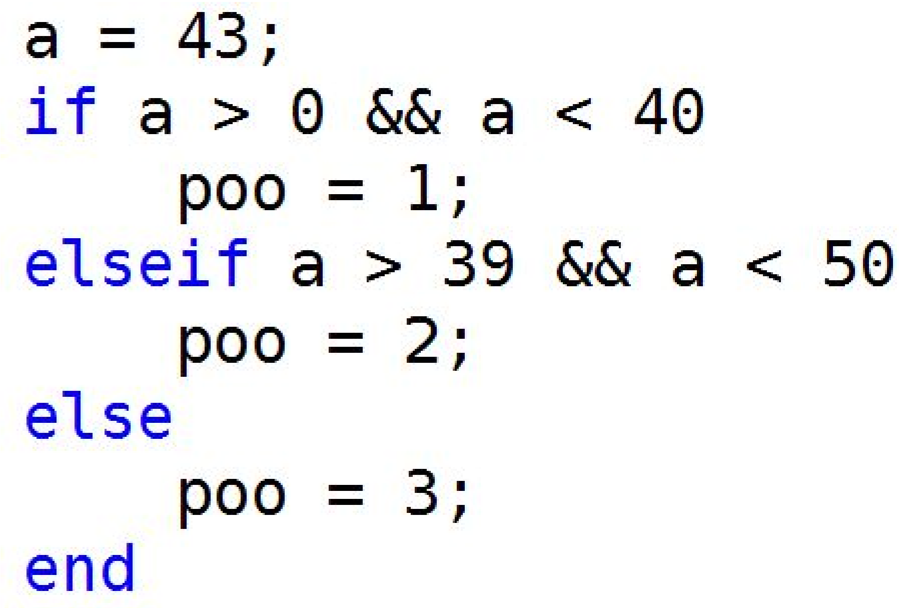
MATLAB Basics

Demo

Operation, function or constant	**MATLAB command**
+ (addition)	`+`
- (subtraction)	`-`
� (multiplication)	`*`
/ (division)	`/`
|x| (absolute value of x)	`abs(x)`
square root of x	`sqrt(x)`
e^x	`exp(x)`
ln x (natural log)	`log(x)`
$\log_{10}$ x (base 10 log)	`log10(x)`
sin x	`sin(x)`
cos x	`cos(x)`
tan x	`tan(x)`
cot x	`cot(x)`
arcsin x	`asin(x)`
arccos x	`acos(x)`
arctan x	`atan(x)`
arccot x	`acot(x)`
n! (n factorial)	`gamma(n+1)`
e (2.71828...)	`exp(1)`
p (3.14159265...)	`pi`
i (imaginary unit, sqrt(-1))	`i`

Programming in MATLAB (Flow control)

IF Statement



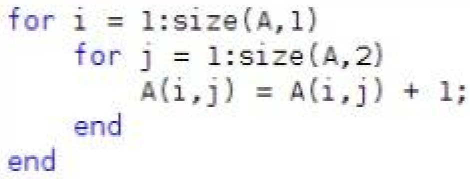
```
a = 43;
if a > 0 && a < 40
    poo = 1;
elseif a > 39 && a < 50
    poo = 2;
else
    poo = 3;
end
```

The FOR loop

```
for iter = 1:100

   disp(iter);

end
```



```
for i = 1:size(A,1)
    for j = 1:size(A,2)
        A(i,j) = A(i,j) + 1;
    end
end
```

A = A + 1;

The FOR loop

```
for iter = 1:100

    disp(iter);

end
```

FOR loops in MATLAB:

- Bad
- MATLAB - interpreted language

The FOR loop

```
for iter = 1:100

    disp(iter);

end
```

FOR loops in MATLAB:

- Bad
- MATLAB - interpreted language

Example - Consider Matrix Multiplication in C++ (Two 2-D arrays).

```
for (int i=0; i<x; i++){
         for (int j=0; j<n; j++){
            c[i][j] = 0;
            for (int k = 0; k < m; k++)
                c[i][j] = c[i][j] + a[i][k] * b[k][j];
         }
      }
```

Complexity is $O(n^3)$. Not good for MATLAB

Interpreted vs Compiled languages

Essentially, in compiled languages (e.g. C++/Fortran) the loop bodies are transformed into machine code only once at compile time. However, in interpreted languages (e.g. Matlab/Python), you must pay the price of interpreting the loop body each time a loop executes. So compiled languages compile loops once and evaluate low-level machine code many times, while interpreted languages must both interpret and evaluate the code many times. This makes looping *many* times slower (often orders of magnitude) in interpreted languages as opposed to compiled languages.

Read about AVX Intel (Vectorised computations) if interested.

Others

- break
- return
- while
- switch - case - otherwise

Some useful commands

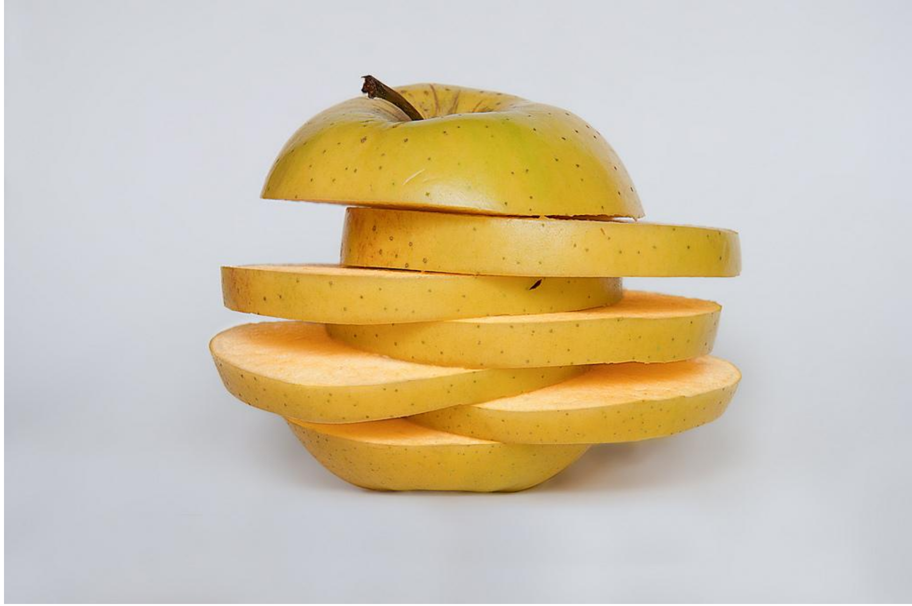
- `clear all; % Clears stored variables`
- `clc; % Clear screen`
- `who/whos; % Information about active variables`
- `help <command/keyword>; % man page`
- `doc <command/keyword>; % Mathwork documentation`
- `disp(variable/string); % Output to STDOUT`
- `ones(...) % Matrix of all ones`
- `zeros(...) % Matrix of all zeroes`
- `size(__anything__) % Returns (1,1) for scalars`

Some useful commands

- Bash (UNIX) commands - ls, cd, pwd etc
- plot, subplot
- imread/imshow
- audioread/sound
- cat
- zeros
- ones
- tic, toc
- hold on
- figure
- linspace
- conv/ conv2

Matlab supports

Slicing!

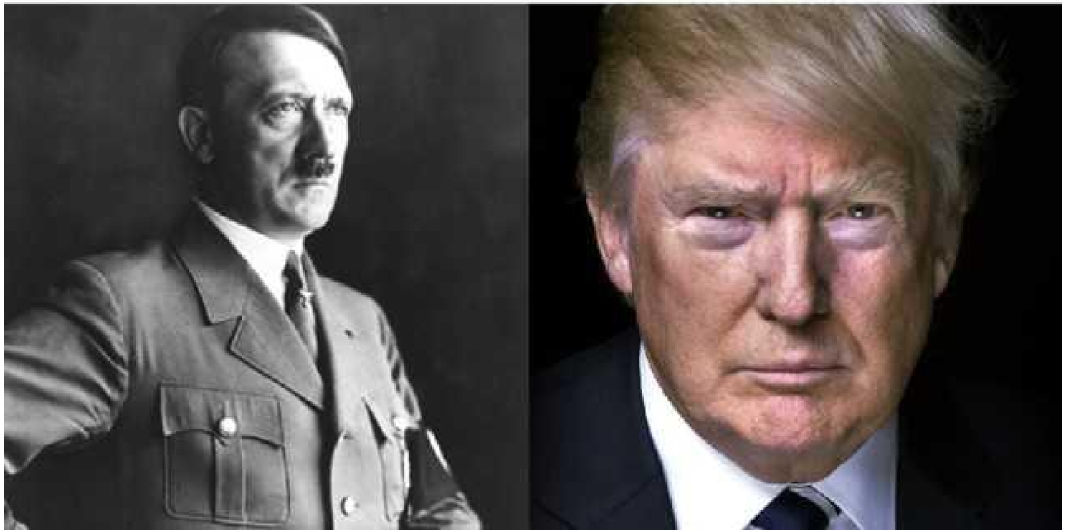


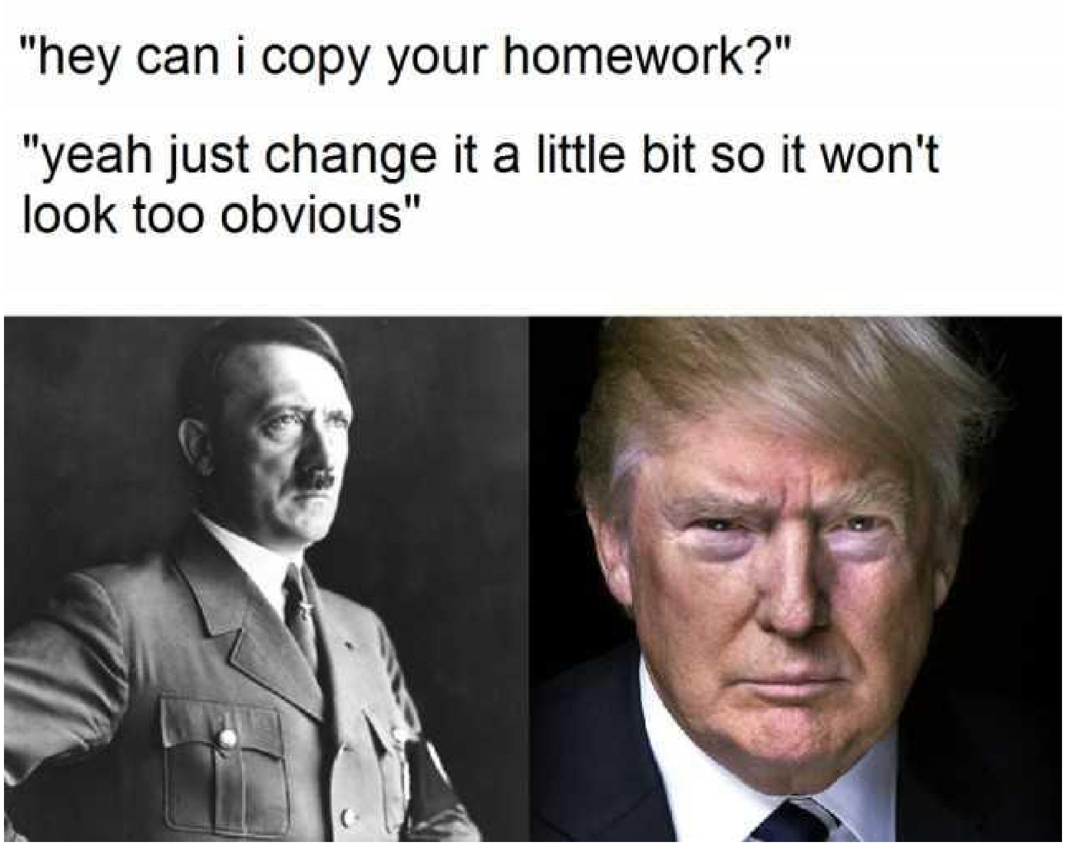
Remember

- Try to avoid usage of FOR loops in MATLAB
- Try to vectorise (matrix representation) data. (Signals, Images, tracks etc)
- Important to understand the notation
  - A 1-D signal in MATLAB is a (1xN) matrix → 1 Row and N samples
  - A grayscale image is a (MxN) matrix → M rows, N columns → M*N pixels
- **MATLAB follows 1-indexing.** Unlike Python/C++ that index arrays from 0
- Explore the various inbuilt functions by referring the documentation
- **Use Functions** for assignments as much as possible
- Practice the basics. Its simple to understand.

"hey can i copy your homework?"

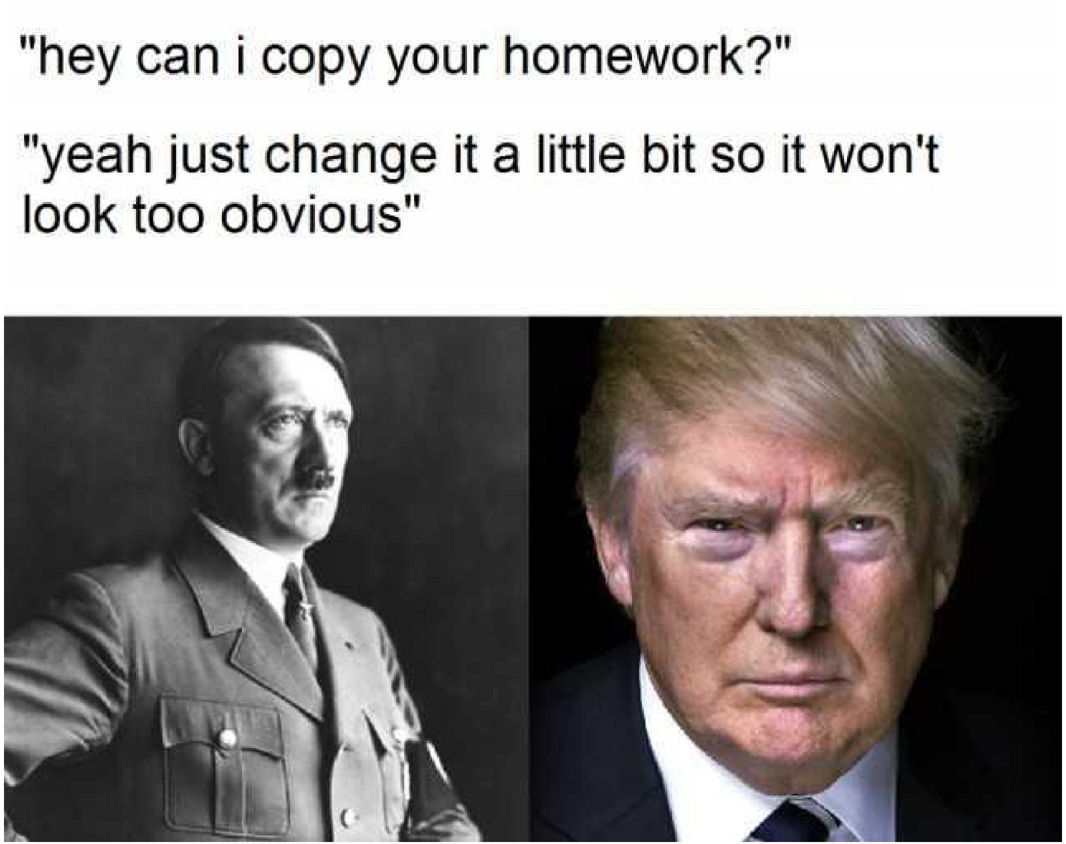
"yeah just change it a little bit so it won't look too obvious"





"hey can i copy your homework?"

"yeah just change it a little bit so it won't look too obvious"



"hey can i copy your homework?"

"yeah just change it a little bit so it won't look too obvious"