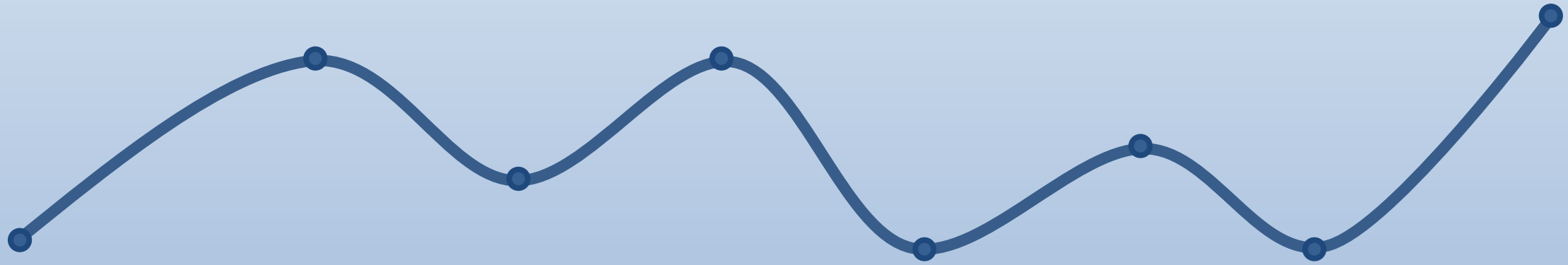
Quick Start Tutorial

Introduction to MATLAB

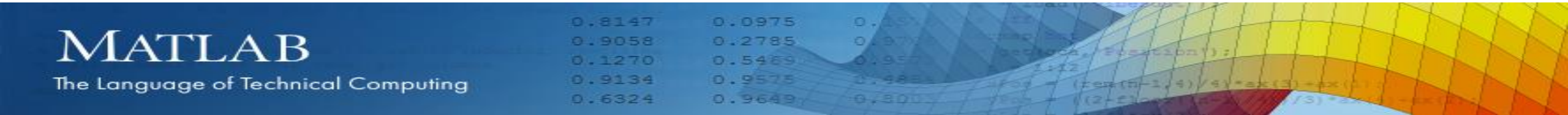
Hans-Petter Halvorsen



https://www.halvorsen.blog

What is MATLAB?

- MATLAB is a tool for technical computing, computation and visualization in an integrated environment.
- MATLAB is an abbreviation for MATrix LABoratory
- It is well suited for Matrix manipulation and problem solving related to Linear Algebra, Modelling, Simulation and Control Applications
- Popular in Universities, Teaching and Research



MATLAB Syntax - Example

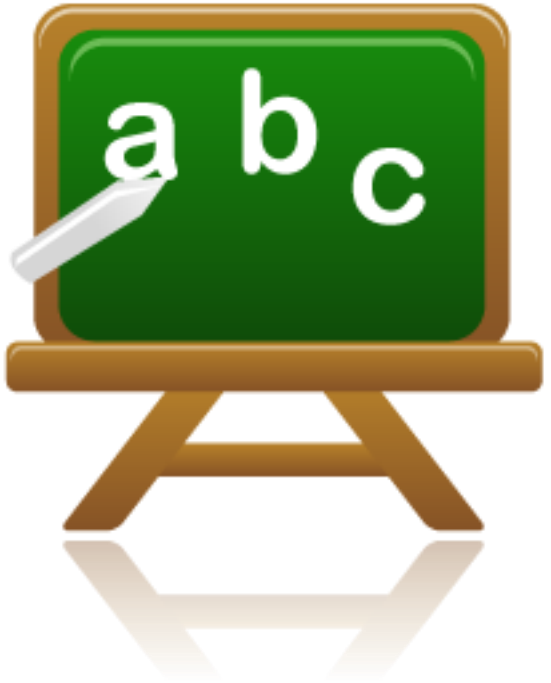
Defining Vectors

For Loop

Built-in Functions

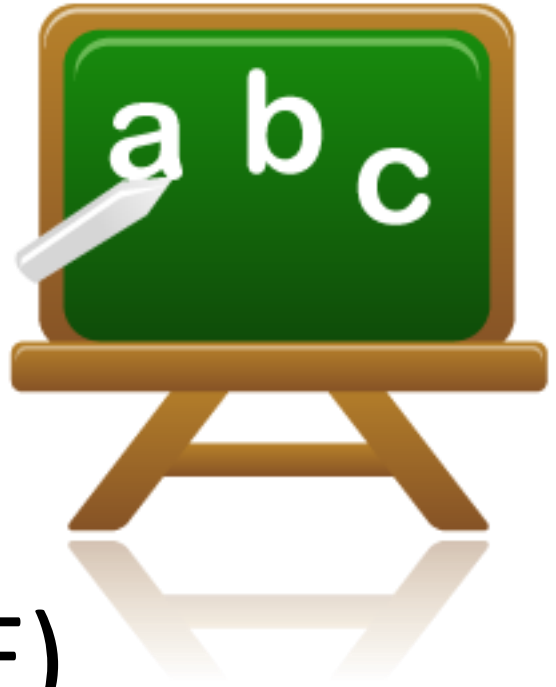
```
clear
clc
close all

x=[0, 1, 2, 3, 4 ,5];
y=[15, 10, 9, 6, 2 ,0];

for n=1:6 % n = model order

    p = polyfit(x,y,n)

    ymodel = polyval(p,x);

    subplot(3,2,n)
    plot(x,y,'o',x,ymodel)
    title(sprintf('Model order %d', n));

end
```

Lessons



1. The MATLAB Environment (IDE)
2. MATLAB Basics
3. Vectors and Matrices
4. Plotting
5. Scripts (m-files)
6. User-defined Functions
7. Flow Control (if...elseif...else, while, switch...case)

Lesson 1



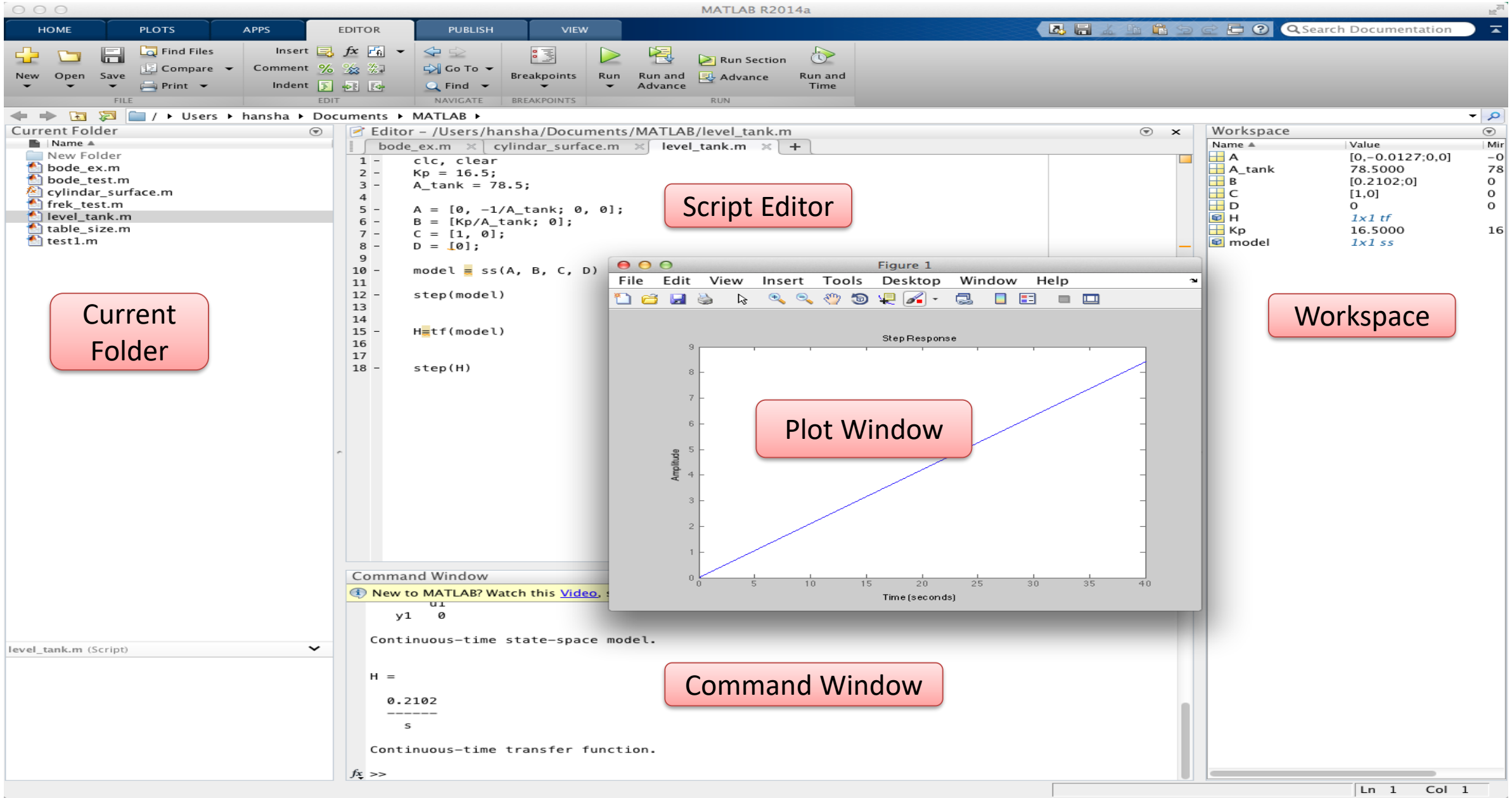
- The MATLAB Environment (IDE)
- MATLAB Basics

The MATLAB Environment (IDE)

Working in the Development Environment

http://www.mathworks.com/videos/working-in-the-development-environment-69021.html

The MATLAB Environment (IDE)



MATLAB Basics

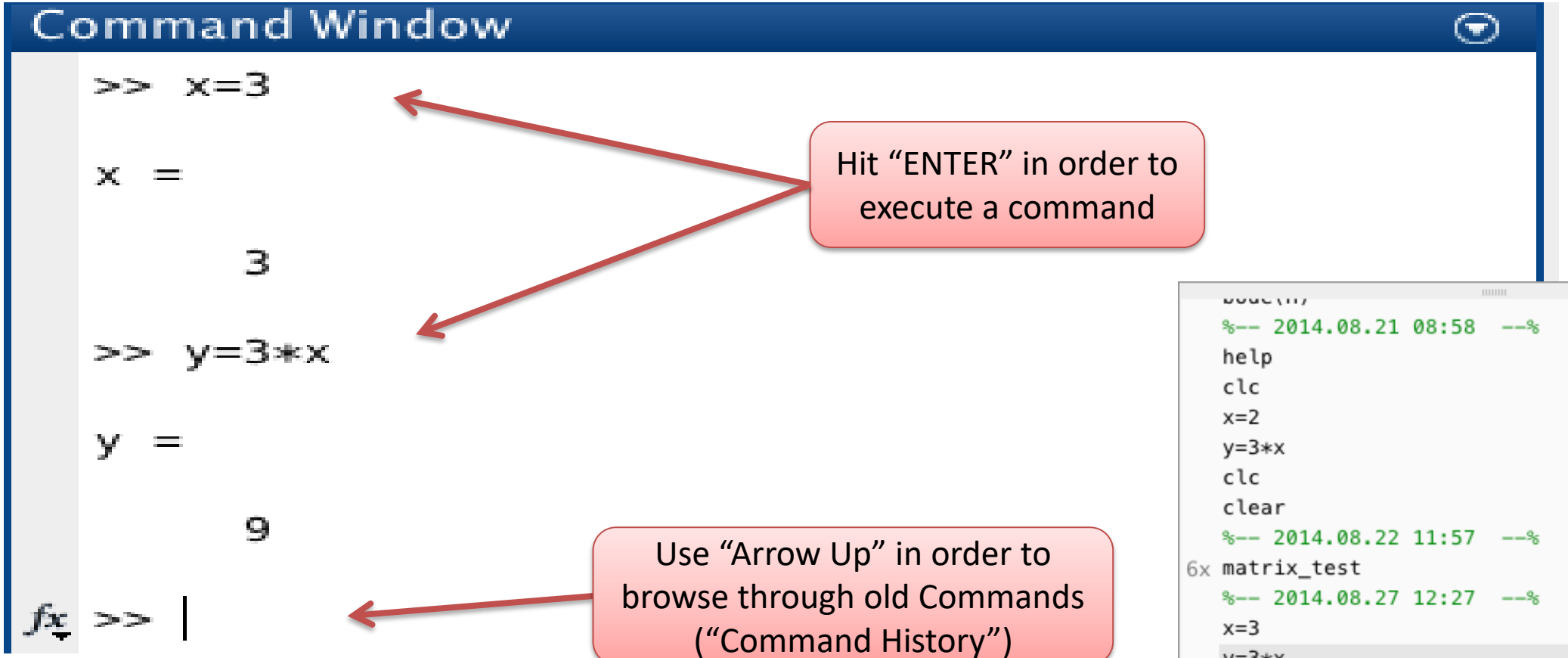
Getting Started with MATLAB

http://www.mathworks.com/videos/getting-started-with-matlab-68985.html

MATLAB Basics

Command Window

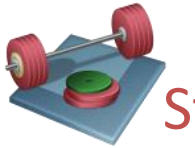
The Command Window is the main window in MATLAB. Use the Command Window to enter variables and to run functions and M-files scripts (more about m-files later). Its like an advanced calculator!



MATLAB Basics

MATLAB is **case sensitive**! The variables *x* and *X* are not the same.

```
>> x=5;
>> X=6;
>> x+X

ans =
     11
```



Students: Try these examples

```
>> x=3
x =
     3

>> y=4;
>>
```

Unlike many other languages, where the semicolon is used to terminate commands, in MATLAB the semicolon serves to suppress the output of the line that it concludes.

MATLAB Basics

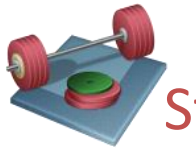
```
>> clear
>> clc
```

The “clear” command deletes all existing variables” from the memory

The “clc” command removes everything from the Command Window
clc – Clear Command Window

```
>> clear x
```

Only clear the variable “x”



Students: Try these commands

MATLAB Basics

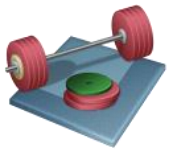
Built-in constants

Name	Description
i, j	Used for complex numbers, e.g., z=2+4i
pi	π
inf	∞, Infinity
NaN	Not A Number. If you, e.g., divide by zero, you get NaN

```
>> r=5;
>> A=pi*r^2

A =
   78.5398
```

```
>> z1=3+3i;
>> z2=3+5i;
>> z = z1+z2
z =
   6.0000 + 8.0000i
```

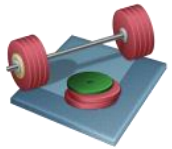
```
>> a=2;
>> b=0;
>> a/b
```



Students: Try these examples

MATLAB Basics

Mathematical Expressions



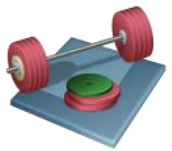
Students: Try this example

$$y(x) = \frac{3x + 2}{2} \qquad y(2) = ?$$

```
>> x=2;
>> y=3*x+2/2
y =
     7
>> y=(3*x+2)/2
y =
     4
```

Which are correct?

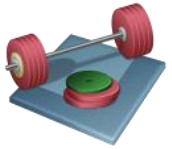
	MATLAB
$\ln(x)$	`log(x)`
$\log_{10}(x)$	`log10(x)`
$\sqrt{x}$	`sqrt(x)`
e^x	`exp(x)`
x^2	`x^2`



Students: Calculate this expression, try with different values for x and y

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

MATLAB Basics



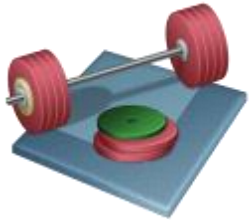
Students: Calculate this expression, try with different values for x and y

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

Solutions:

```
>> x=2;, y=2
>> z = 3*x^2 + sqrt(x^2 + y^2) + exp(log(x))

ans =
    16.8284
...
```

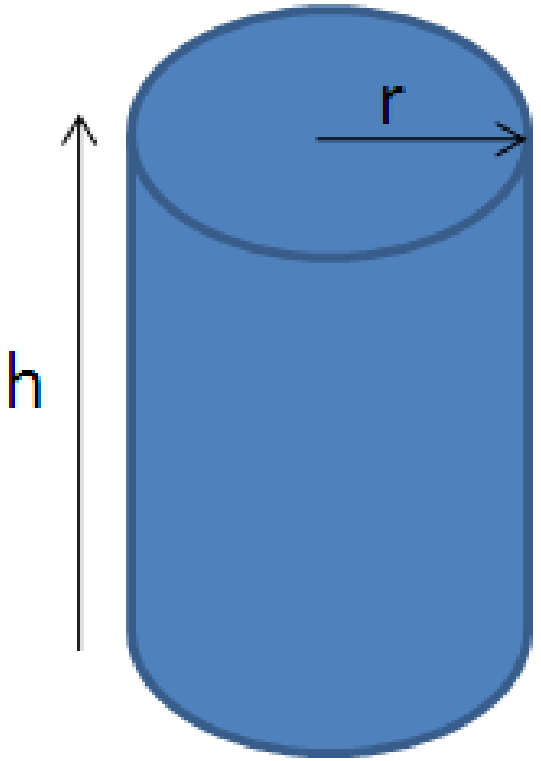
MATLAB Basics



Students: Use MATLAB in order to find the surface area (A) of a cylinder based on the height (h) and the radius (r) of the cylinder

$r = 3$

$h = 8$

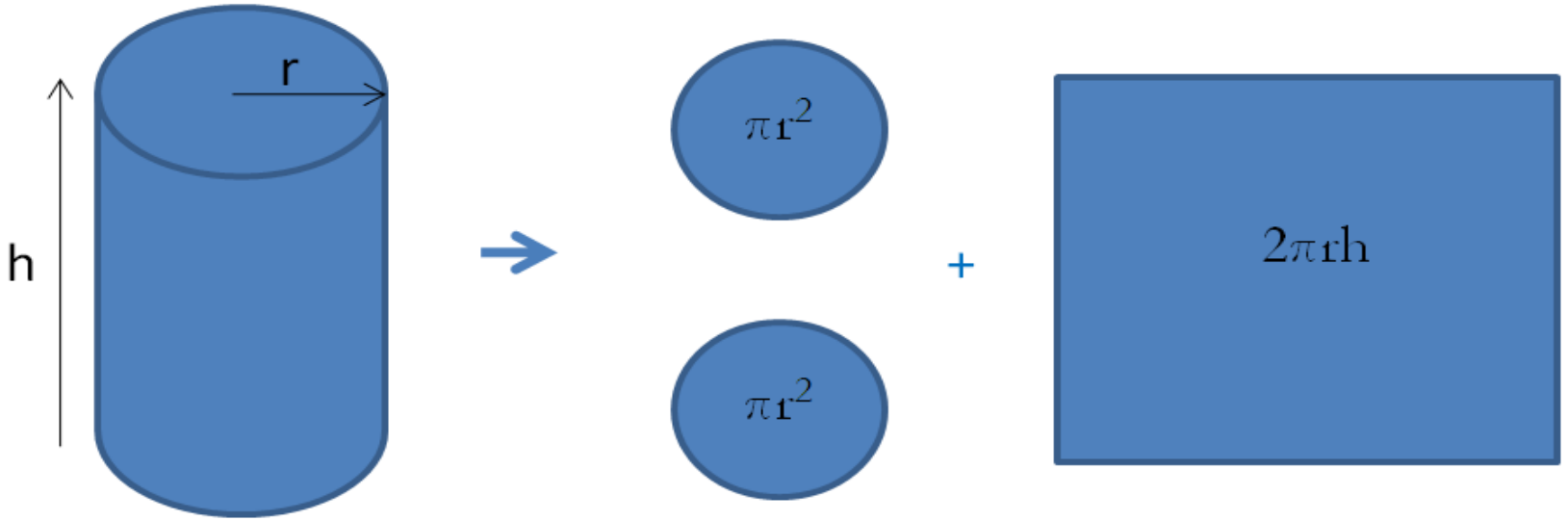


$A = ?$

MATLAB Basics

Students: Find the surface area of a cylinder based on the height (h) and the radius (r) of the cylinder

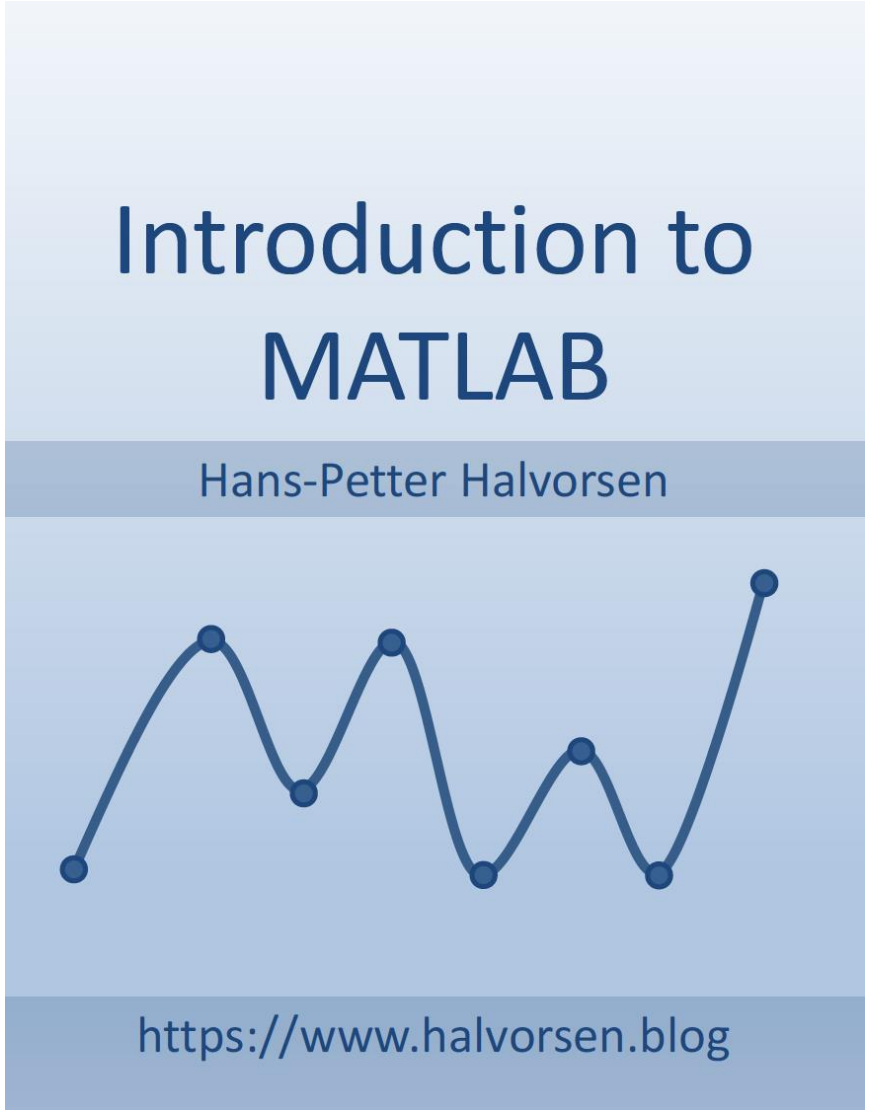
Solutions:




```
>> h=8
>> r=3
>> A = 2*pi*r^2 +2*pi*r*h;
A =
   207.3451
```

Whats next?

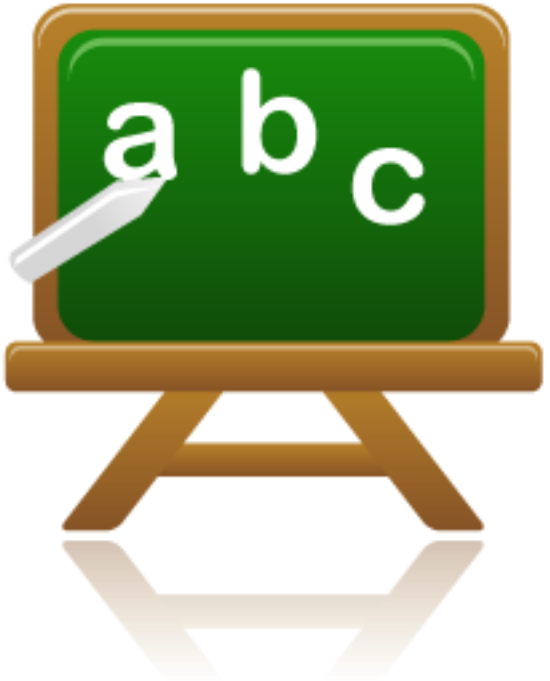
Learning by Doing!



Self-paced Tutorials with lots of Exercises and Video resources

Do as many Exercises as possible! The only way to learn MATLAB is by doing Exercises and hands-on Coding!!!

Lesson 2



- Vectors & Matrices
- Plotting

Vectors & Matrices

Working with Arrays

http://www.mathworks.com/videos/working-with-arrays-in-matlab-69022.html

Vectors & Matrices

- Matrices and vectors (Linear Algebra) are the basic elements in MATLAB and also the basic elements in control design theory, etc.
- All variables in MATLAB is a matrix (but with different dimensions)
- So it is important you know how to handle vectors and matrices in MATLAB and in general

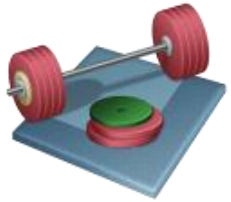
$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \in R^{nxm} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^n$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Vectors

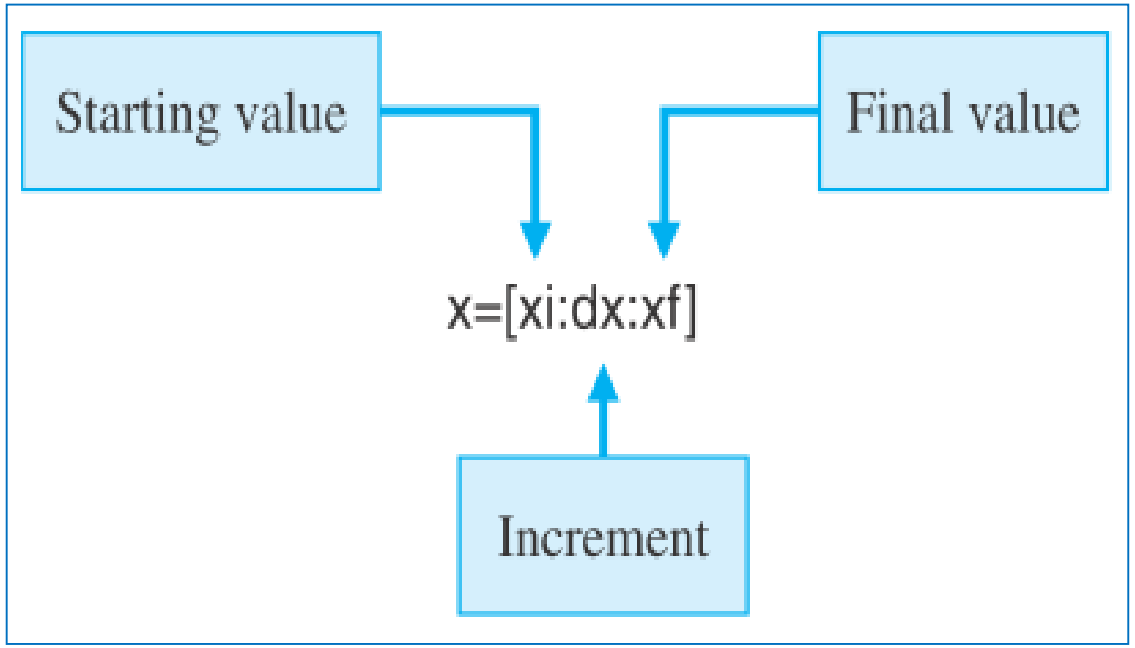
Examples of different Rows and Columns vectors

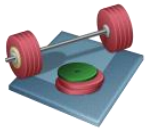
```
>> x = [1, 2, 3]

>> y = [4; 5; 6]

>> z = [8, 9, 10]'
```



Students: Define these vectors in MATLAB. Try also to multiply the different vectors like this:

```
>> x*y
>> y*x
>> x*z
>> y*z
...
```





Students: Try these examples

```
>> a = [1:10]

>> b = [1:2:10]

>> b = [1:0.5:4]
```

Vectors

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

We define the Data in MATLAB like this:

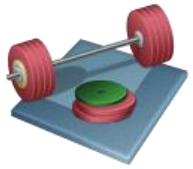
```
>> x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
```

If we are only interested in the Rain Amount on Monday:

```
>> x(1)
ans =    2.1000
```

Rain Amount on Friday:

```
>> x(5)
ans =    2.5000
```



Students: Try these examples

Etc.

Vectors

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

We define the Data in MATLAB like this:

```
>> x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
```

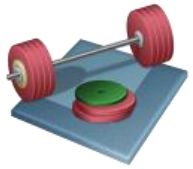
What is the Average Rain Amount this Week?

In MATLAB we can use the ”mean” function:

```
>> mean(x)
ans =    5.5714
```

We can define a variable, e.g.:

```
>> mean_value_week = mean(x)
mean_value_week =    5.5714
```



Students: Try these examples

Vectors

Given the following function:

$$y(x) = 2x^2 + 3x + 1$$

where: $-10 \leq x \leq 10$

```
>> x=-10:10
>> y=2.*x.^2 + 3.*x + 1
y =
   171   136   105    78
55    36    21    10     3
0     1     6    15    28
45    66    91   120   153
190   231
```

Note how we have used .* and .^

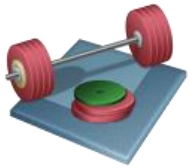
.* each element-wise Multiplication

.^ each element-wise Power

What is $y(3) =$?

```
>> y(14)
ans =     28
```

Index	x	y(x)
1	-10	171
2	-9	136
3	-8	105
4	-7	78
5	-6	55
6	-5	36
7	-4	21
8	-3	10
9	-2	3
10	-1	0
11	0	1
12	1	6
13	2	15
14	3	28
15	4	45
16	5	66
17	6	91
18	7	120
19	8	153
20	9	190
21	10	231



Students: Try these examples

We can also do like this:

```
>> x = 3;
>> y = 2*x^2 + 3*x + 1
y =     28
```

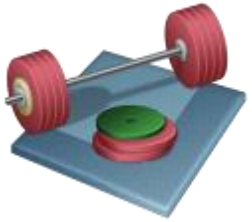
Matrices

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
>> A = [1 2; 3 4]

A =   1     2
      3     4
```

or:

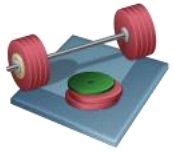
```
>> A = [1, 2; 3, 4]

A =   1     2
      3     4
```



Students: Define the following matrices in MATLAB

$$B = \begin{bmatrix} 4 & 3 & 0 \\ 1 & -7 & 2 \\ 8 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 3 & 0 \\ 4 & 7 & -2 \\ 2 & 0 & 9 \end{bmatrix}$$



Try these examples

```
>> B+C
>> B-C
>> B/C
>> B*C
>> B.*C
>> B'*C
...
```

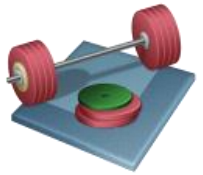
Matrices

$$n\left[\overset{m}{A}\right] m\left[\overset{p}{B}\right] = n\left[\overset{p}{C}\right]$$

Given the following matrices:

$$A = \begin{bmatrix} 1 & 3 & 0 \\ 1 & -2 & 2 \\ 3 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$



Define the matrices and try these examples

```
>> A*B
>> B*A
>> A+B
>> B'
>> B'*C
>> A*B'
>> A'*B'
>> A.*B
...
```

```
>> A*(B*C)
>>  (A*B)*C
>>  (A+B)*C
>> A*C + C*B
>>  (A+inv(B))*C

...
```

```
>> rank(A)
>> det(A)
>> inv(A)
>> inv(B)
>> eig(A)
>> inv(A)
>> inv(B)
>> diag(A)
>> inv(A)*A
>> A*inv(A)

...
```

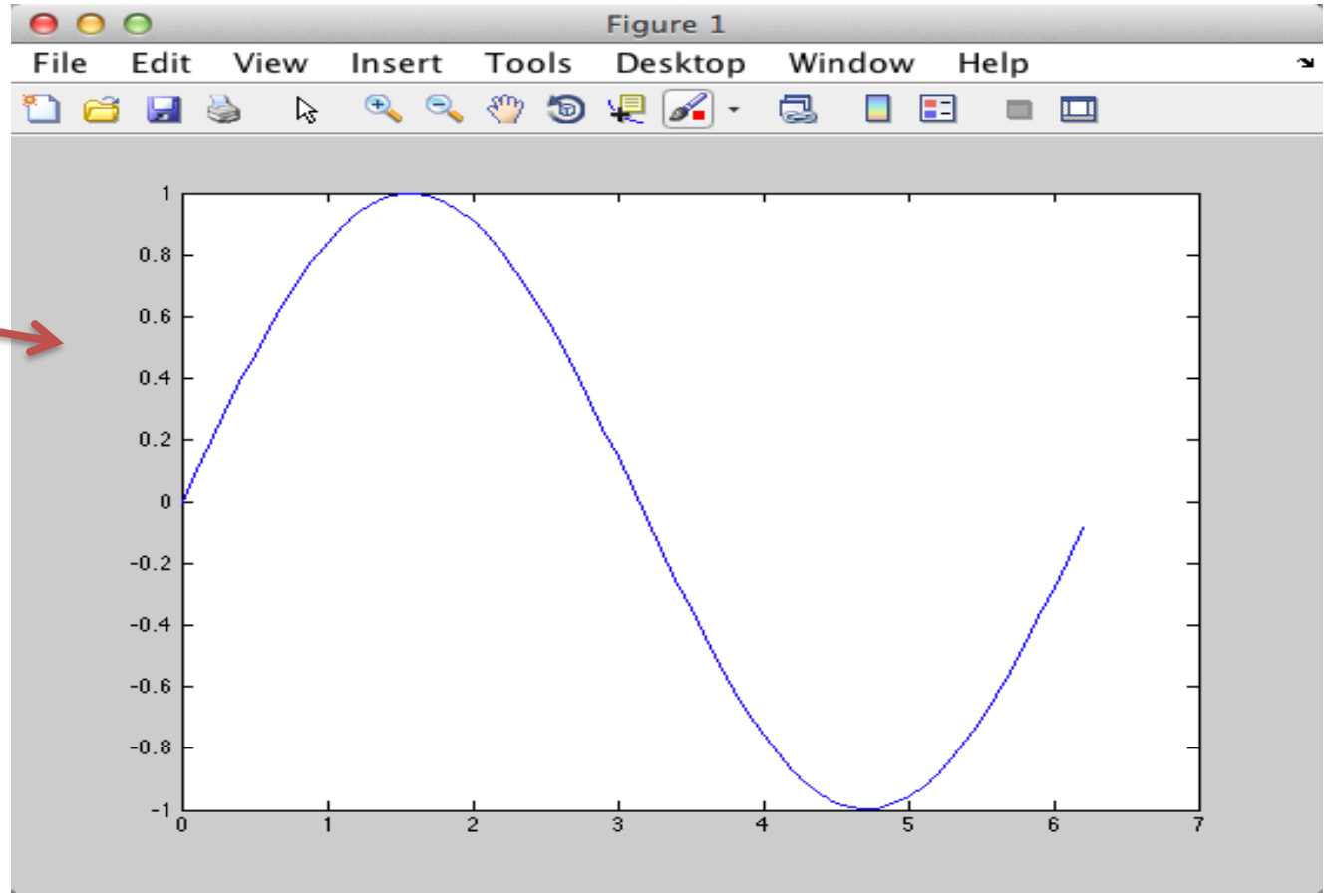
Plotting

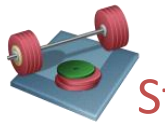
Using Basic Plotting Functions

http://www.mathworks.com/videos/using-basic-plotting-functions-69018.html

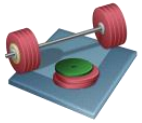
Plotting

```
>> x = 0:0.1:2*pi;
>> y = sin(x);
>> plot(x,y)
```





Students: Try this example



Students: Try also these examples:

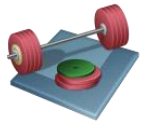
```
>> x = 0:0.1:2*pi;
>> y = sin(x);
>> y2 = cos(x);
>> plot(x,y, x,y2)
```

```
...
>> plot(x,y,'r*', x,y2,'g+')
```

Plotting

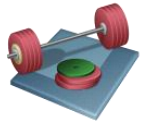
Plotting functions:

Name	Description
plot	Create a Plot
figure	Define a new Figure/Plot window
grid on/off	Create Grid lines in a plot
title	Add Title to current plot
xlabel	Add a Label on the x-axis
ylabel	Add a Label on the x-axis
axis	Set xmin,xmax,ymin,ymax
hold on/off	Add several plots in the same Figure
legend	Create a legend in the corner (or at a specified position) of the plot
subplot	Divide a Figure into several Subplots



Students: Try this example

```
>> x=0:0.1:2*pi;
>> y=sin(x);
>> plot(x,y)
>> title('Plot Example')
>> xlabel('x')
>> ylabel('y=sin(x)')
>> grid on
>> axis([0,2*pi,-1,1])
>> legend('Temperature')
```

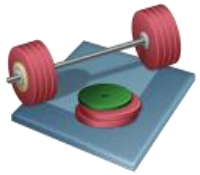


Students: Try also to change some of the commands and see what happens with the plot

Plotting

Given the following Rain Data for a given Week (Monday to Sunday):

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm



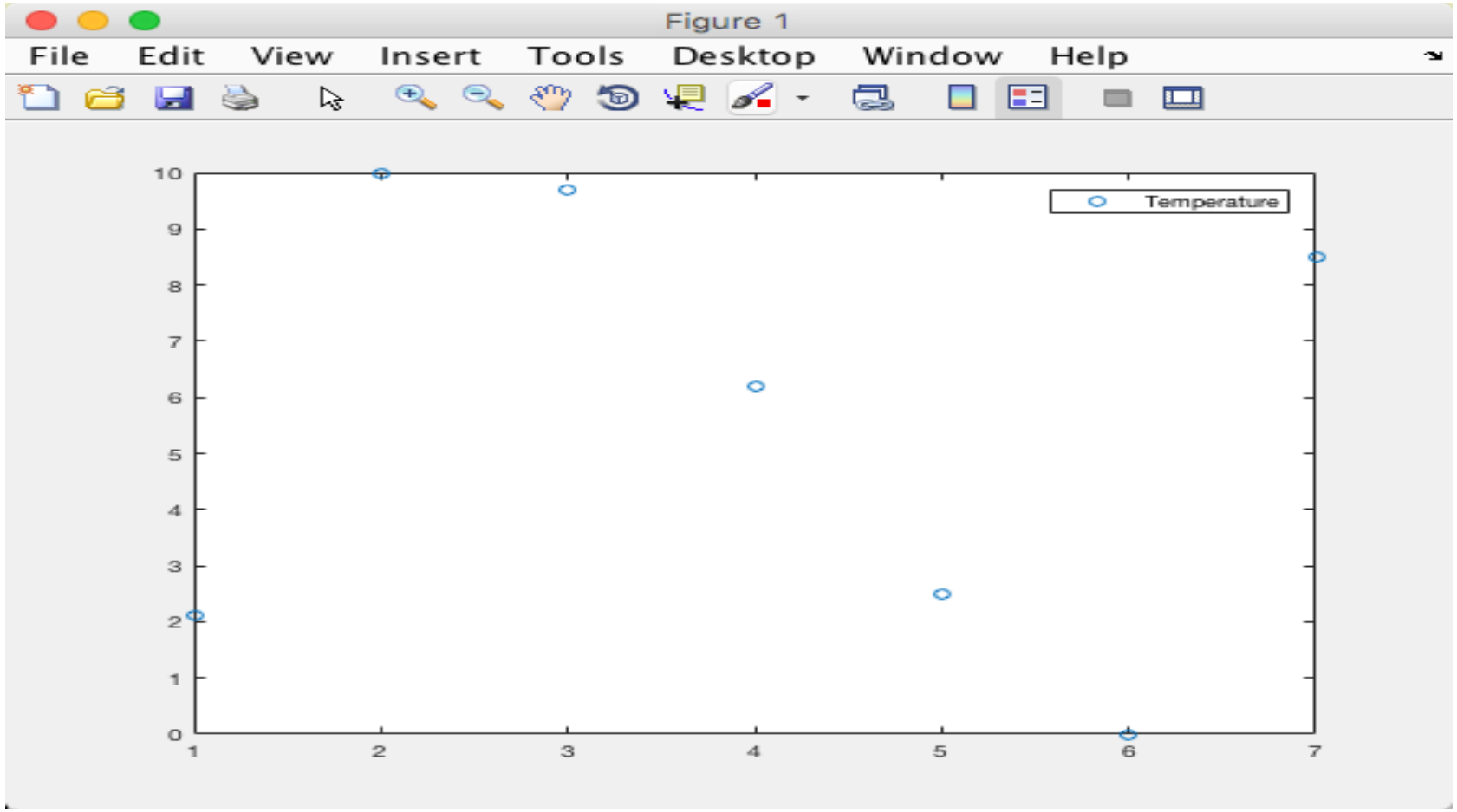
Students: Plot these Values

Solution

Plotting

Day	Rain Amount
Monday	2,1 mm
Tuesday	10 mm
Wednesday	9,7 mm
Thursday	6,2 mm
Friday	2,5 mm
Saturday	0 mm
Sunday	8,3 mm

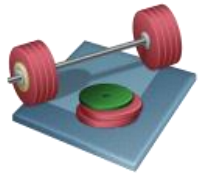
```
x = [2.1, 10, 9.7, 6.2, 2.5, 0, 8.5]
>> plot(x, 'o')
```



Plotting

Given the following function ($-10 \leq x \leq 10$):

$$y(x) = 2x^2 + 3x + 1$$

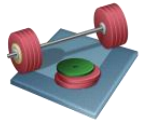


Students: Plot this function

Use the Plot to find out:

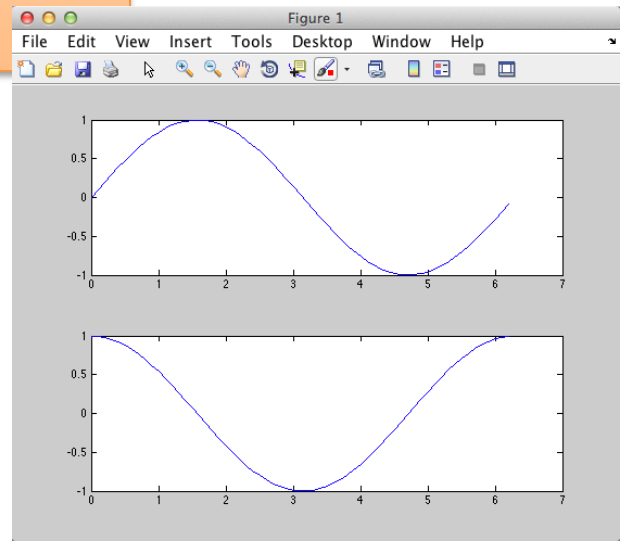
- For which value of x is $f(x) = 0$?
- What is $f(5) =$?

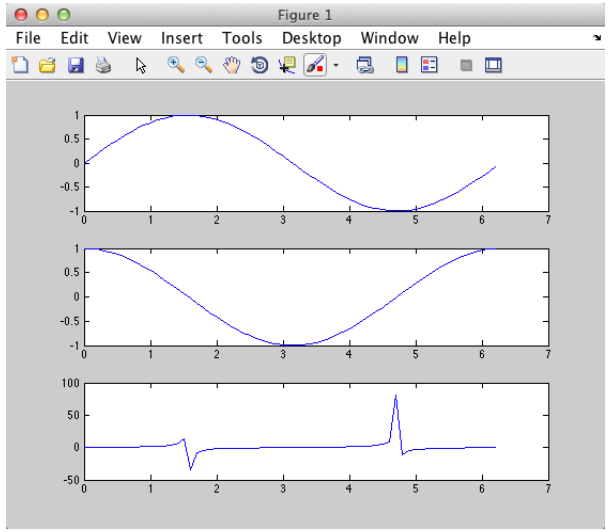
Plotting Subplot



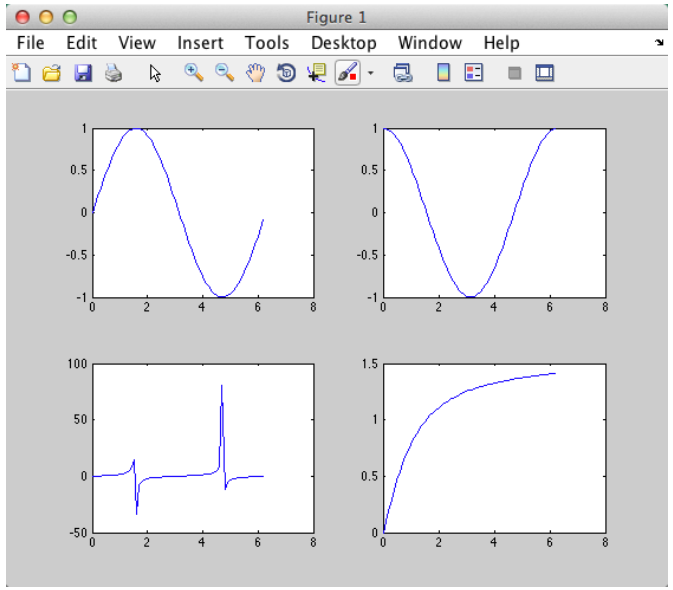
Students: Try these examples

```
>> x=0:0.1:2*pi;
>> y=sin(x);
>> y2=cos(x);

>> subplot(2,1,1)
>> plot(x,y)

>> subplot(2,1,2)
>> plot(x,y2)
```





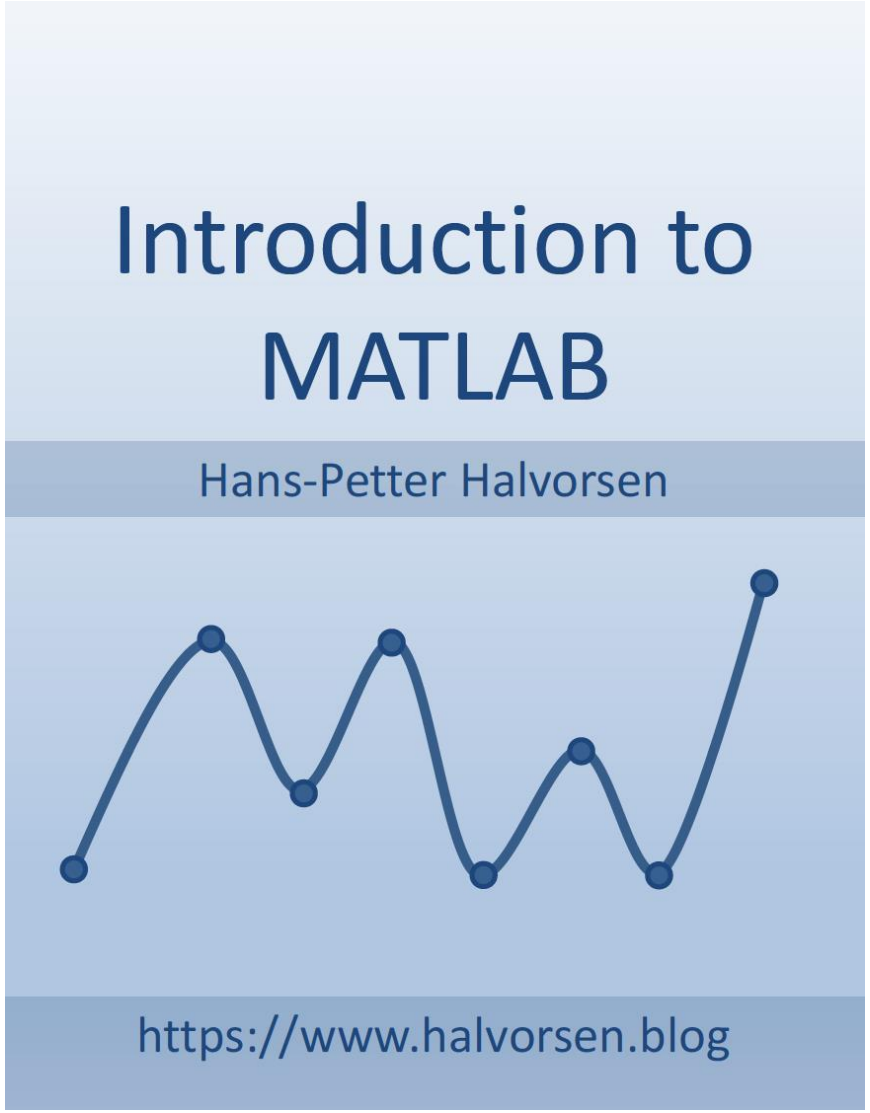

```
>> x=0:0.1:2*pi;
>> y=sin(x);
>> y2=cos(x);
>> y3=tan(x);

>> subplot(3,1,1)
>> plot(x,y)

>> subplot(3,1,2)
>> plot(x,y2)

>> subplot(3,1,3)
>> plot(x,y3)
```




```
>> x=0:0.1:2*pi;
>> y=sin(x);
>> y2=cos(x);
>> y3=tan(x);
>> y4=atan(x);

>> subplot(2,2,1)
>> plot(x,y)

>> subplot(2,2,2)
>> plot(x,y2)

>> subplot(2,2,3)
>> plot(x,y3)

>> subplot(2,2,4)
>> plot(x,y4)
```

Whats next?

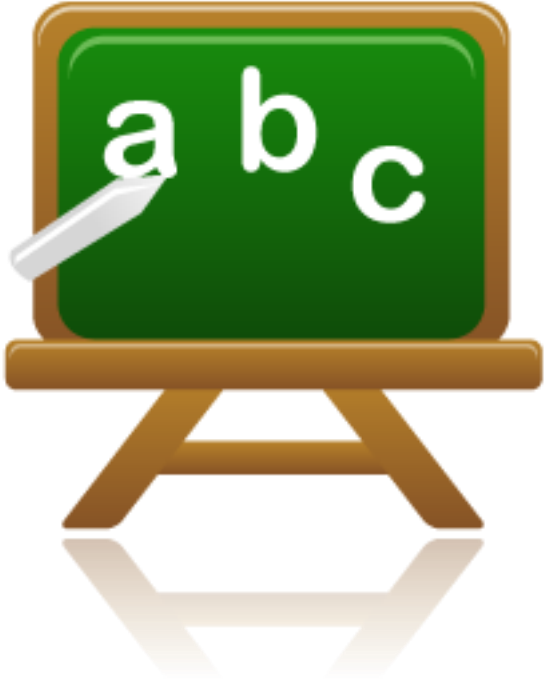
Learning by Doing!



Self-paced Tutorials with lots of Exercises and Video resources

Do as many Exercises as possible! The only way to learn MATLAB is by doing Exercises and hands-on Coding!!!

Lesson 3



- Scripts (m-files)
- User-defined Functions

Scripts (m-files)

Writing a MATLAB Program

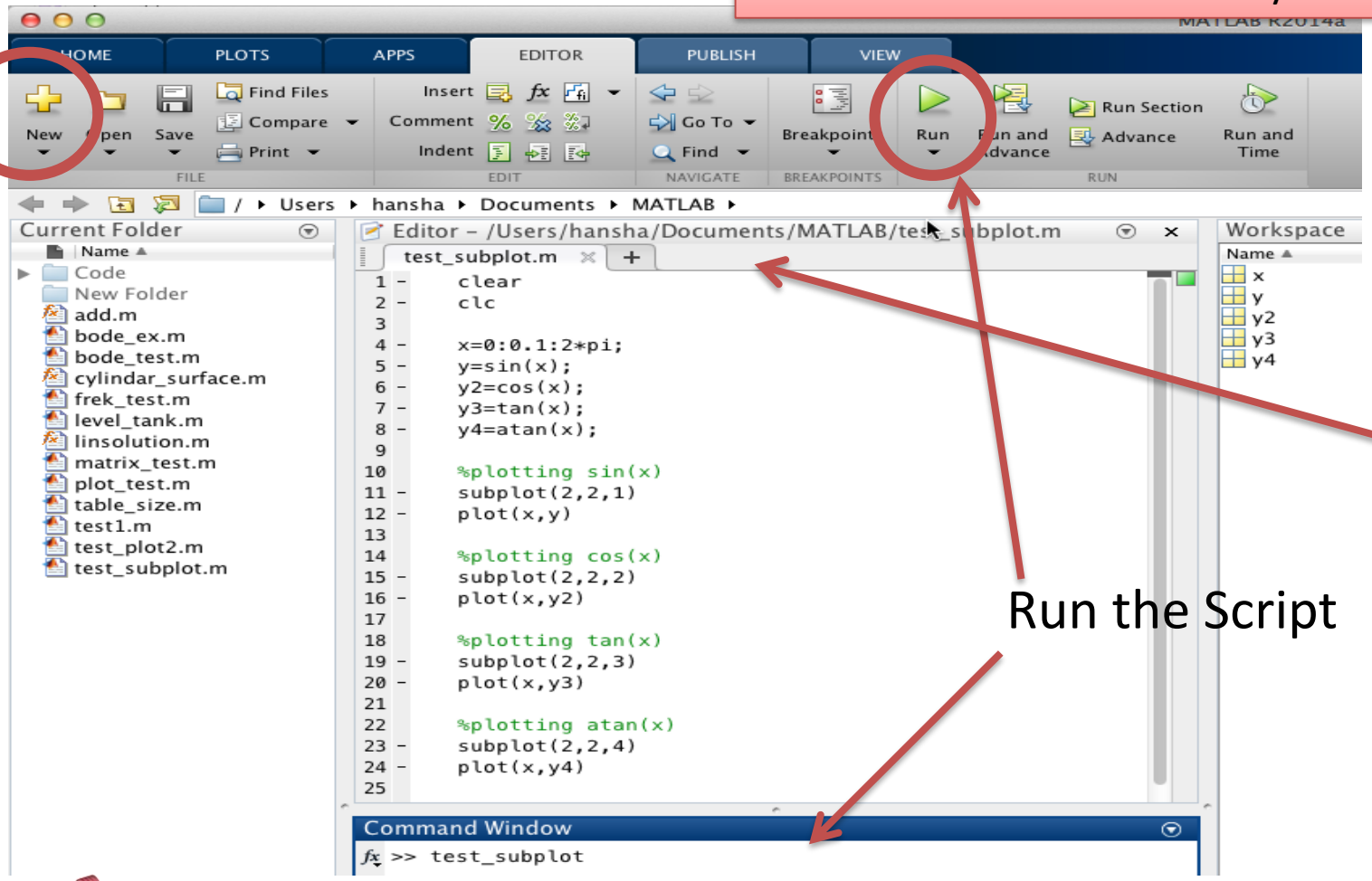
http://www.mathworks.com/videos/writing-a-matlab-program-69023.html

Scripts (m-files)

MATLAB Scripts are saved as so-called .m files (file extension is .m)

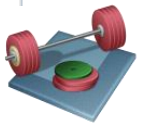
Script Editor

When using the Script Editor, you may create several lines of code and execute all in one batch. You can easily do changes in your code, create comments, etc.



Run the Script

```
clear
clc

x=0:0.1:2*pi;
y=sin(x);
y2=cos(x);
y3=tan(x);
y4=atan(x);

%plotting sin(x)
subplot(2,2,1)
plot(x,y)

%plotting cos(x)
subplot(2,2,2)
plot(x,y2)

%plotting tan(x)
subplot(2,2,3)
plot(x,y3)

%plotting atan(x)
subplot(2,2,4)
plot(x,y4)
```



Students: Try this example

User-defined Functions

MATLAB contains hundreds of built-in functions, but very often you need to create your own functions

Editor – /Users/hansha/Documents/MATLAB/add.m

add.m

```
1   function answer = add(x,y)
2   % This function adds 2 numbers
3
4 -   answer = x + y;
```

Input

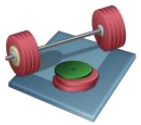
Return value

You Create the Function in the Editor

Command Window

```
>> x=2;
>> y=4;
>> add(x,y)

ans =

     6

fx >>
```

You Use the Function in the Command Window or in a Script



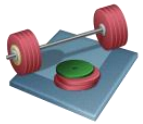
Students: Try this example

```
function output = function_name(input)
```

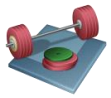
User-defined Functions

Example: Convert from Celsius to Fahrenheit

$$T_F = \frac{9}{5}T_C + 32$$



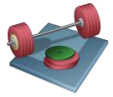
Students: Create a User-defined Function that converts from Temperature in Celsius to Temperature in Fahrenheit



Try the function in the Command window like this:

```
>> Tc = 20;
>> Tf =
fahrenheit(Tc)

Tf =

    68
```

You need to create this function



Try the function in a Script like this:

Editor – /Users/hansha/Documents/MATLAB/temp_sim.m

fahrenheit.m | temp_sim.m

```
1 -    clear
2 -    clc
3
4 -    t = 0:0.1:24;
5 -    Tc = (sin(t)+1)*20;
6 -    Tf = fahrenheit(Tc);
7
8 -    plot(t,Tc, t,Tf)
9
10 -   title('Temperature Simulation')
11 -   xlabel('t')
12 -   ylabel('Temperature')
13 -   grid on
14 -   axis([0,24, 0, 120]);
15 -   legend('Celcius', 'Fahrenheit')
16
```

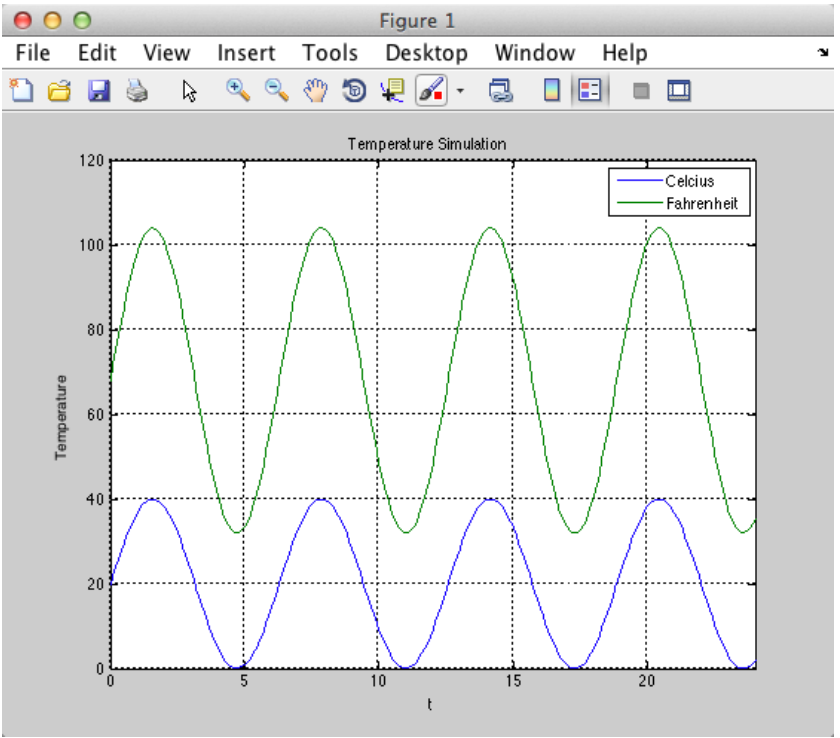
User-defined Functions

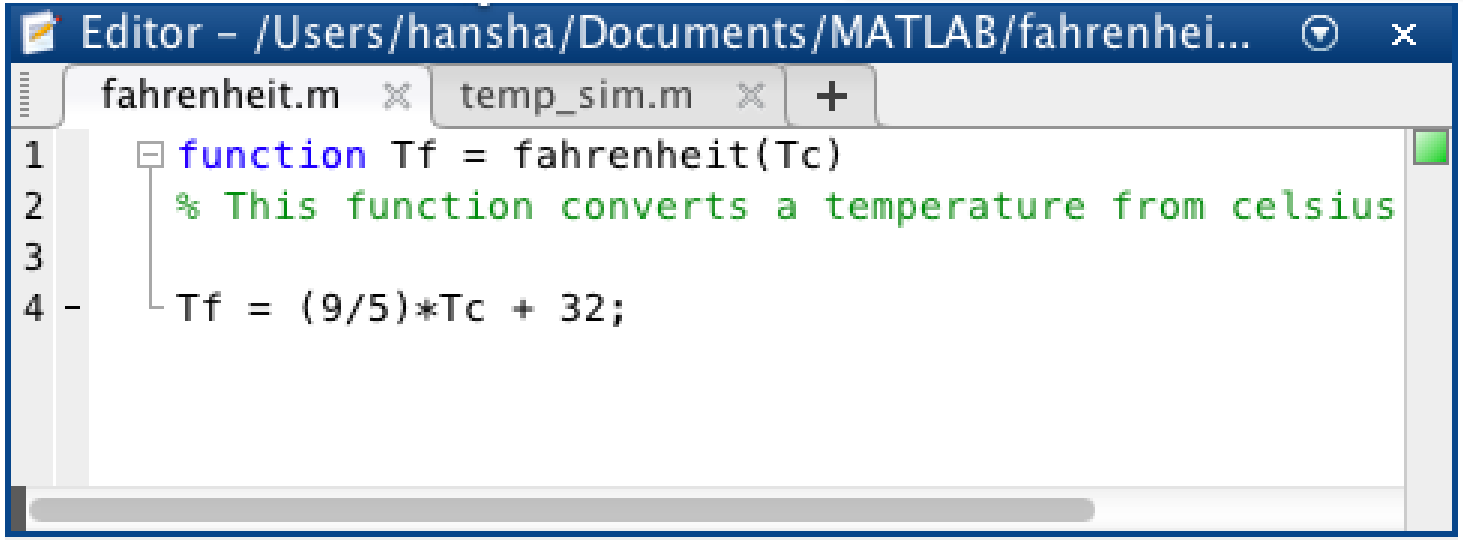
Solutions: Convert from Celsius to Fahrenheit

$$T_F = \frac{9}{5}T_C + 32$$

```
function Tf = fahrenheit(Tc)
% This function converts a temperature from celsius to
fahrenheit

Tf = (9/5)*Tc + 32;
```

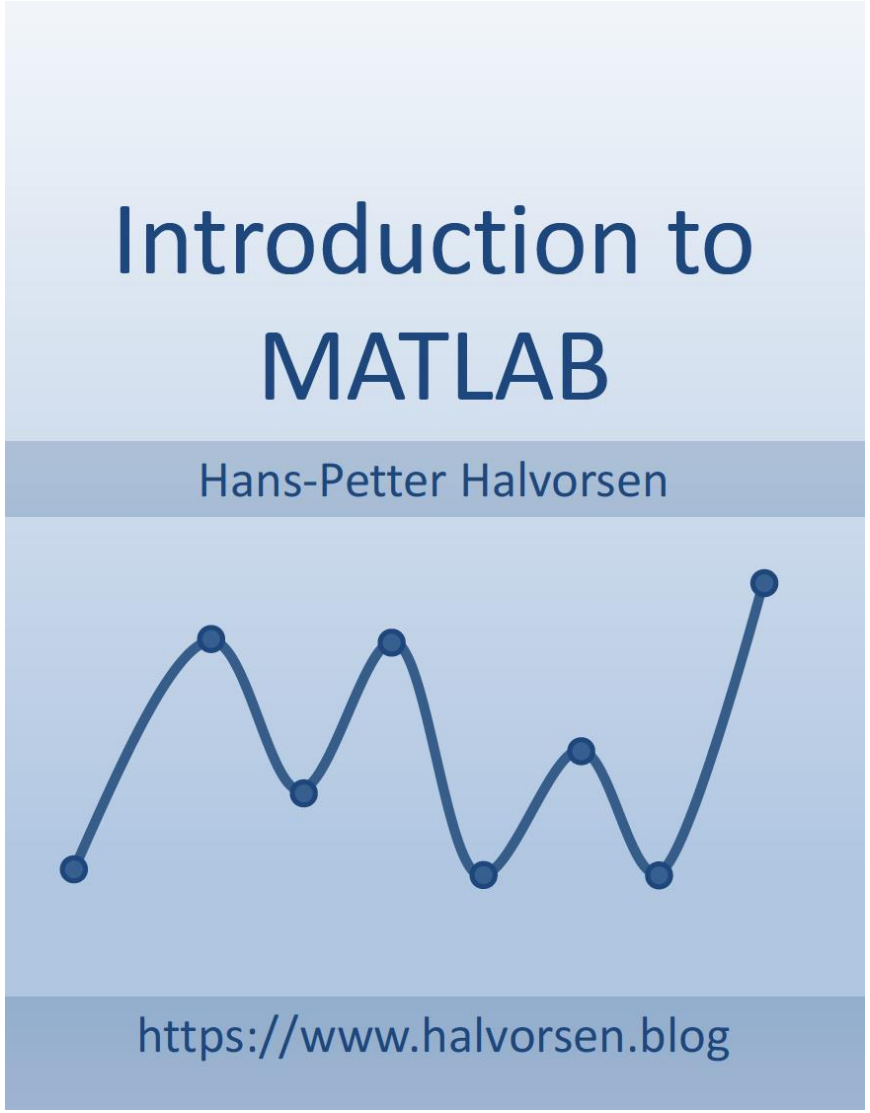
```
clear
clc

t = 0:0.1:24;
Tc = (sin(t)+1)*20;
Tf = fahrenheit(Tc);

plot(t,Tc, t,Tf)

title('Temperature Simulation')
xlabel('t')
ylabel('Temperature')
grid on
axis([0,24, 0,120]);
legend('Celcius', 'Fahrenheit')
```





Whats next?

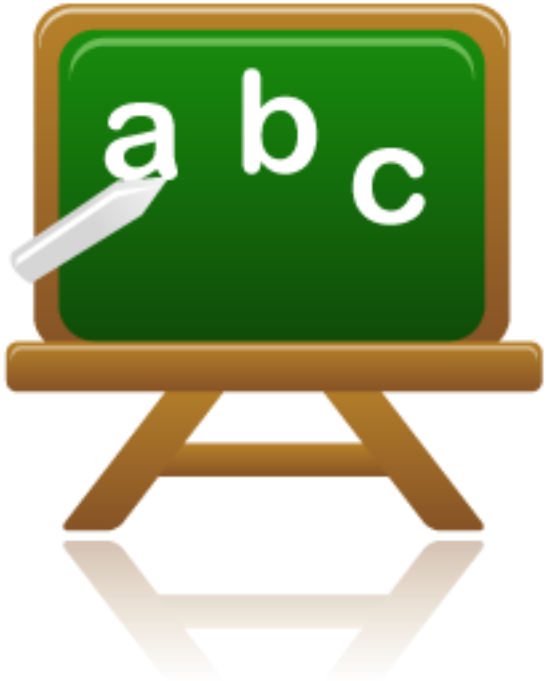
Learning by Doing!



Self-paced Tutorials with lots of Exercises and Video resources

Do as many Exercises as possible! The only way to learn MATLAB is by doing Exercises and hands-on Coding!!!

Lesson 4



- Flow Control
  - –if...elseif...else
  - –while
  - –switch...case

Flow Control

Flow Control:

- if-elseif-else statement
- switch-case-otherwise statement

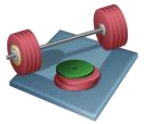
Loops:

- for Loop
- while Loop

The behavior is the same as in other programming languages. It is assumed you know about For Loops, While Loops, If-Else and Switch statements from other programming languages, so we will briefly show the syntax used in MATLAB and go through some simple examples.

Flow Control

if –elseif-else



Students: Try this example

Run the Script several times with different values of n and see what happens

Note!!!

Note! You have to use `if n==1` and NOT `if n=1`

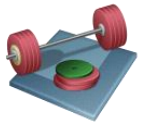
```
clear
clc

n=2;

if n==1
    disp('n=1')
elseif n==2
    disp('n=2')
elseif n==3
    disp('n=3')
else
    disp('n is not 1, 2 or 3')
end
```

Operator	Description
<	Less Than
<=	Less Than or Equal To
>	Greater Than
>=	Greater Than or Equal To
==	Equal To
~=	Not Equal To

Students: Try the different operators

Flow Control

switch-case-otherwise



Students: Try this example

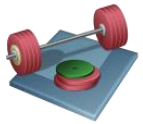
Run the Script several times with different values of n and see what happens

```
clear
clc

n=1;

switch(n)
    case 1
        disp('n=1')
    case 2
        disp('n=2')
    case 3
        disp('n=3')
    otherwise
        disp('n is not 1, 2 or 3')
end
```

"if-elseif-else" and "switch-case-otherwise" is very similar in use

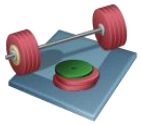
Flow Control

for loop

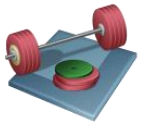


Students: Try this example

```
clear
clc

x = [4, 6, 3, 9, 22, 11];

N = length(x);

for i=1:N
    x(i)
end
```



Students: Try with different x vectors



Students: Create a script that sums all the numbers in a vector (array)

$$\sum_{i=1}^{N} x_i$$

Solution:

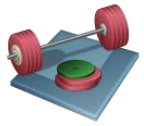
```
clear
clc

x = [4, 6, 3, 9, 22, 11];

N = length(x);
total = 0;

for i=1:N
    total = total + x(i)
end
```

Flow Control

while loop

We want to find for what value of x the function has its minimum value

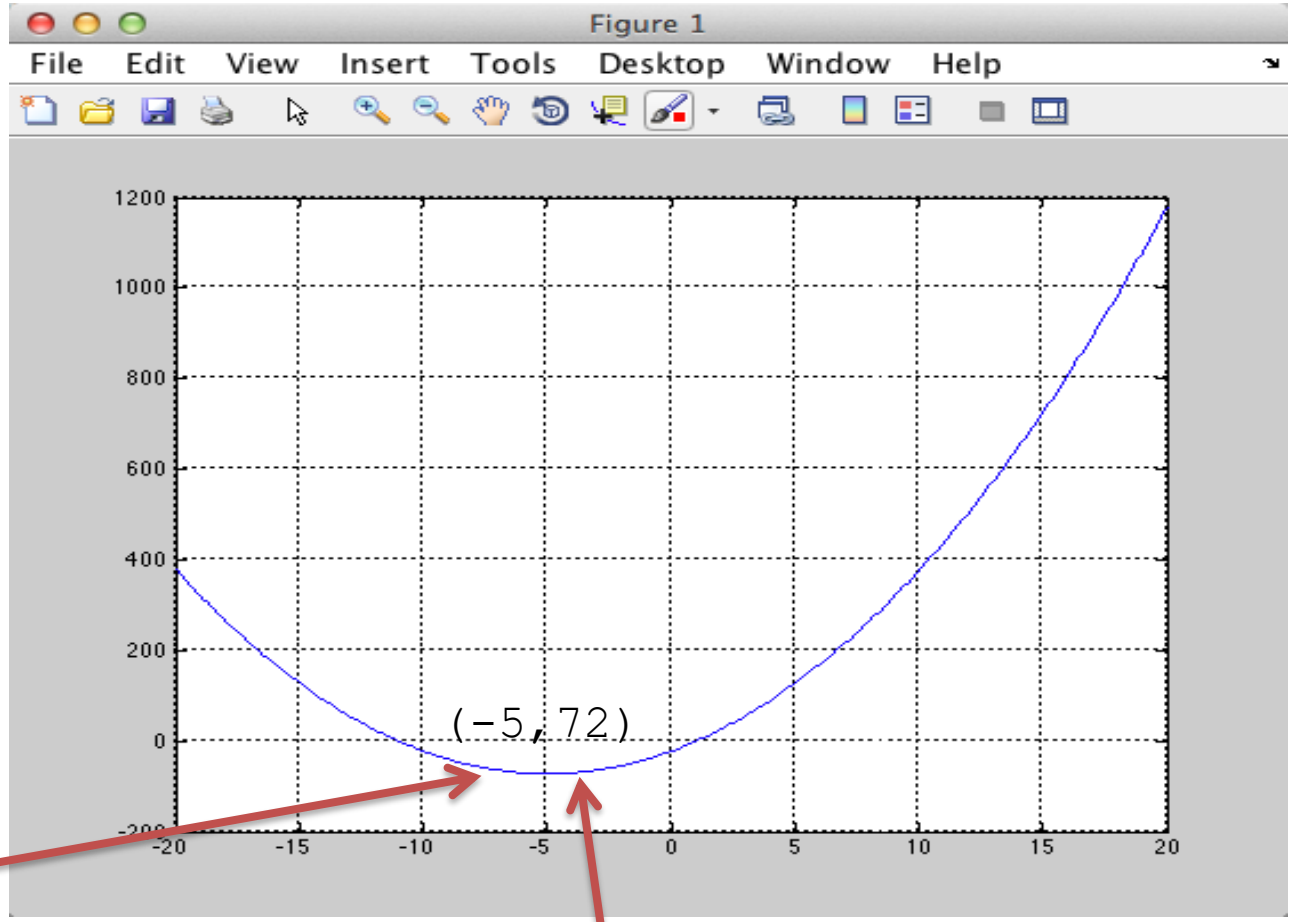
$$y(x) = 2x^2 + 20x - 22$$



Students: Try this example.
Try also with other 2.degree functions

```
clear
clc

x = -20:0.1:20;
y = 2.*x.^2 + 20.*x - 22;
plot(x,y)
grid

i=1;
while ( y(i) > y(i+1) )
    i = i + 1;
end

x(i)
y(i)
```

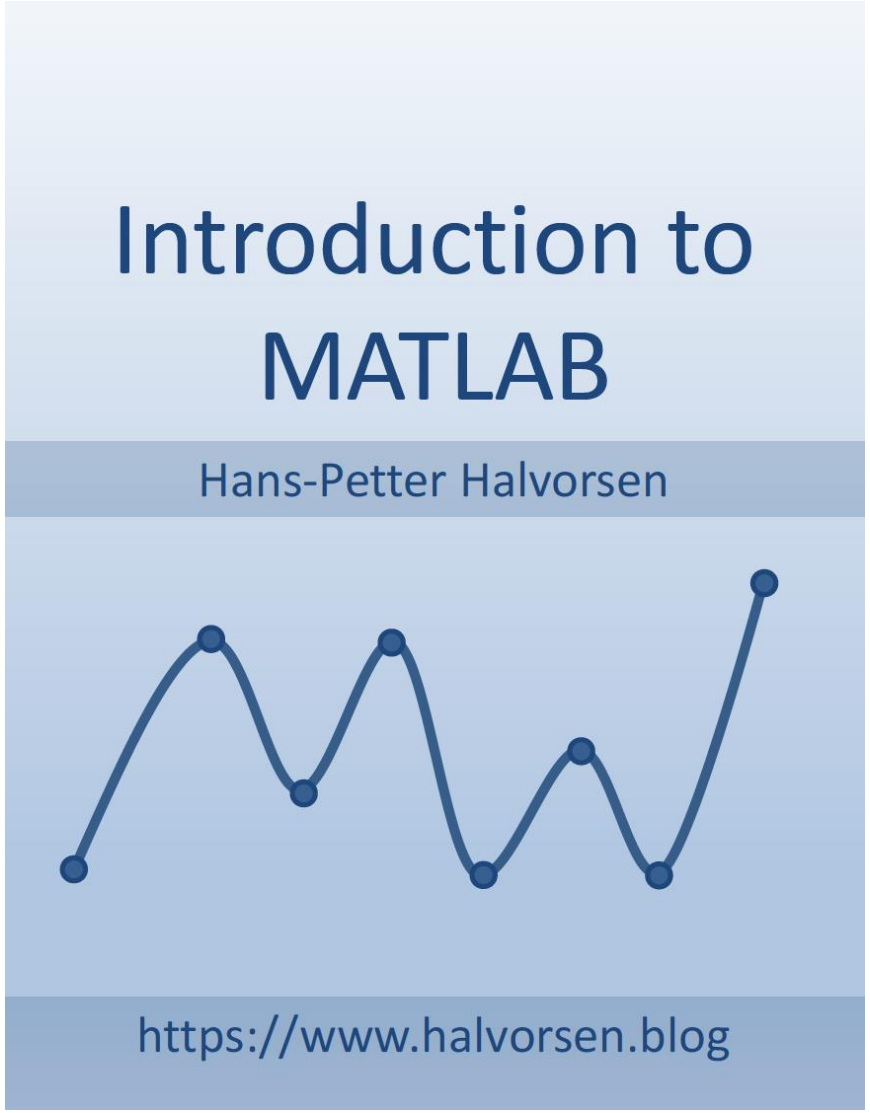
Element-wise multiplication



The minimum of the function

Whats next?

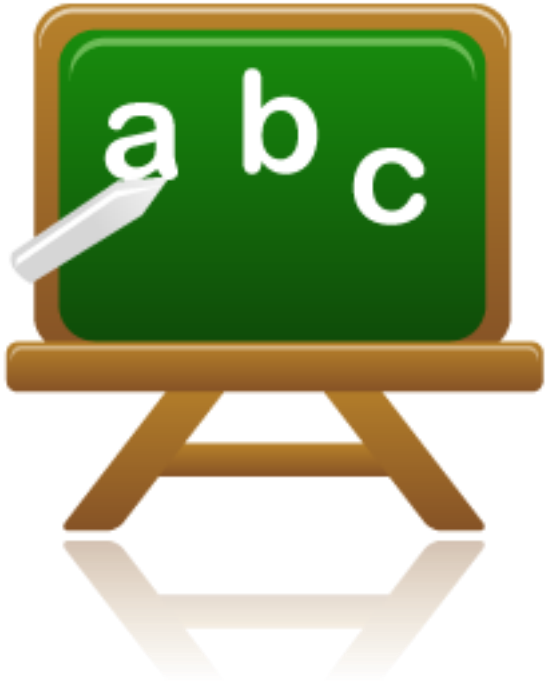
Learning by Doing!



Self-paced Tutorials with lots of Exercises and Video resources

Do as many Exercises as possible! The only way to learn MATLAB is by doing Exercises and hands-on Coding!!!

Tips & Tricks



Tips & Tricks

Use Comments (**%**)

```
% This is a comment
x=2; % Comment2
y=3*x % Comment3
```

- but they have to make sense!

DO NOT use ”spaces” in Filename or names that are similiar to built-in functions in MATLAB!

Use the arrows keys to ”browse” in previous commands used in the Command Window

Decimal sign: Use ”.”– NOT ”,” !
i.e. *y=3.2* – not *y=3,2*

Use english names on variables, functions, files, etc. This is common practice in programming!
Use always variables – Do not use numbers directly in the expressions!

Yes:

```
a=2;
b=4;
y=a+b
```

No:

```
y=2+4
```

Functions:

- Only ONE function in each File!
- The Filename (.m) AND the Name of the Function MUST be the same!

Always include these lines in your Script:

```
clear
clc
close all
…
```

Tips & Tricks

Greek letters: In maths and control theory it is common to use greek letters in formulas, etc. These cannot be used directly in MATLAB, so you need to find other good alternatives. Examples:

ω_0 – w0

ζ – zeta or just z

etc.

Use help in order to find out how to use a function in MATLAB. In order to get help for the tf function, type the following in the Command window:

```
>>help tf
```

A Golden Rule: One Task – one m file, i.e. DON'T put all the Tasks in one single m file!!

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

$$z(2,2) = ?$$

```
x = 2;
y = 2;
z = 3*x^2 + sqrt(x^2 + y^2)+ exp(log(x))
```

Mathematical expressions:
The following applies in MATLAB

x^2	`x^2`
$\sqrt{x}$	**sqrt**(x)
$ln(x)$	**log**(x)
$\log(x)$	**log10**(x)
e^x	**exp**(x)
π	**pi**

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog