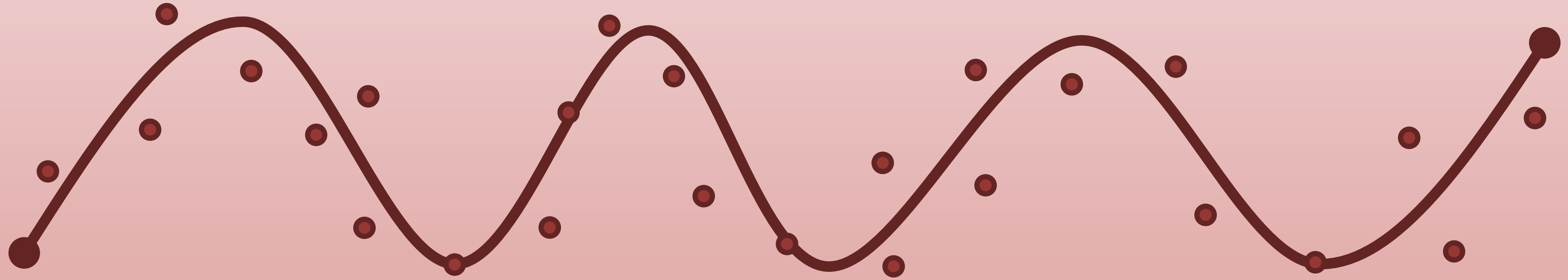
Quick Start Tutorial

Modelling, Simulation and Control in MATLAB

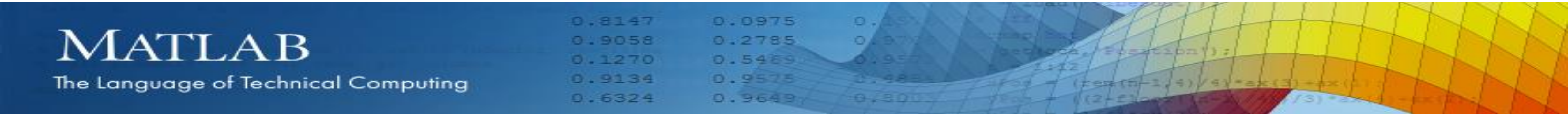
Hans-Petter Halvorsen



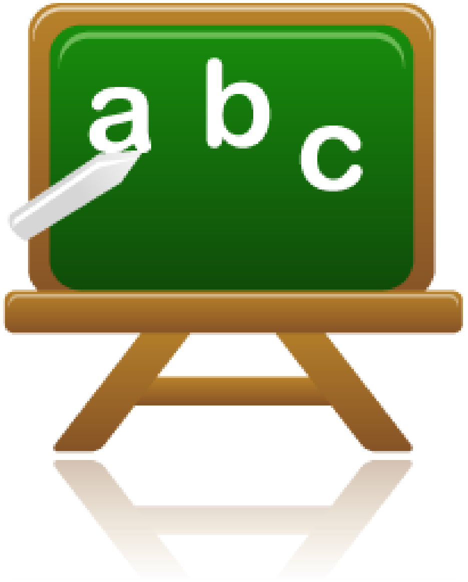
https://www.halvorsen.blog

What is MATLAB?

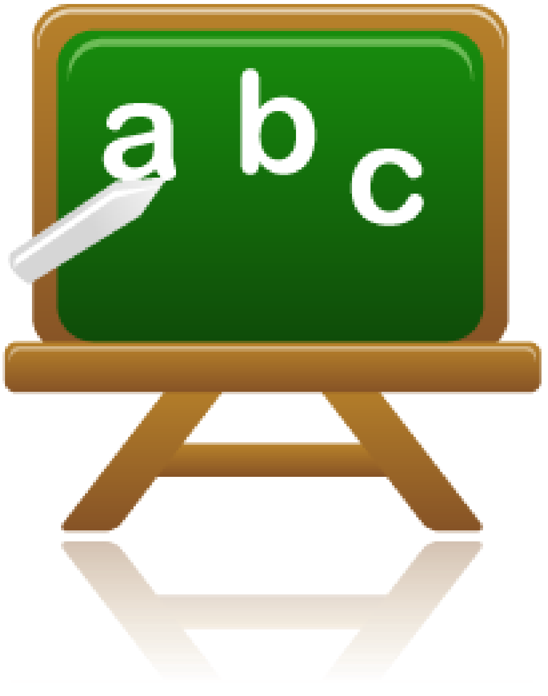
- MATLAB is a tool for technical computing, computation and visualization in an integrated environment.
- MATLAB is an abbreviation for MATrix LABoratory
- It is well suited for Matrix manipulation and problem solving related to Linear Algebra, Modelling, Simulation and Control Applications
- Popular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



Lessons



1. Solving Differential Equations (ODEs)
2. Discrete Systems
3. Interpolation/Curve Fitting
4. Numerical Differentiation/Integration
5. Optimization
6. Transfer Functions/State-space Models
7. Frequency Response



Lesson 1

Solving ODEs in MATLAB

- Ordinary Differential Equations

$$\ddot{x} = -\frac{k}{m}x - \frac{c}{m}\dot{x} + \frac{1}{m}F$$

Differential Equations

Example:

$$\boxed{\dot{x} = ax}$$

Where $a = -\frac{1}{T}$

T is the Time constant

Note! $\dot{x} = \frac{dx}{dt}$

The Solution can be proved to be (will not be shown here):

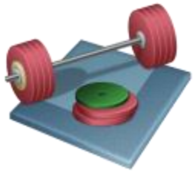
$$\boxed{x(t) = e^{at}x_0}$$

Use the following: $T = 5$

$$x(0) = 1$$

$$0 \le t \le 25$$

```
T = 5;
a = -1/T;
x0 = 1;
t = [0:1:25];

x = exp(a*t)*x0;

plot(t,x);
grid
```



Students: Try this example

Differential Equations

$$x(t) = e^{at}x_0$$

$T = 5$

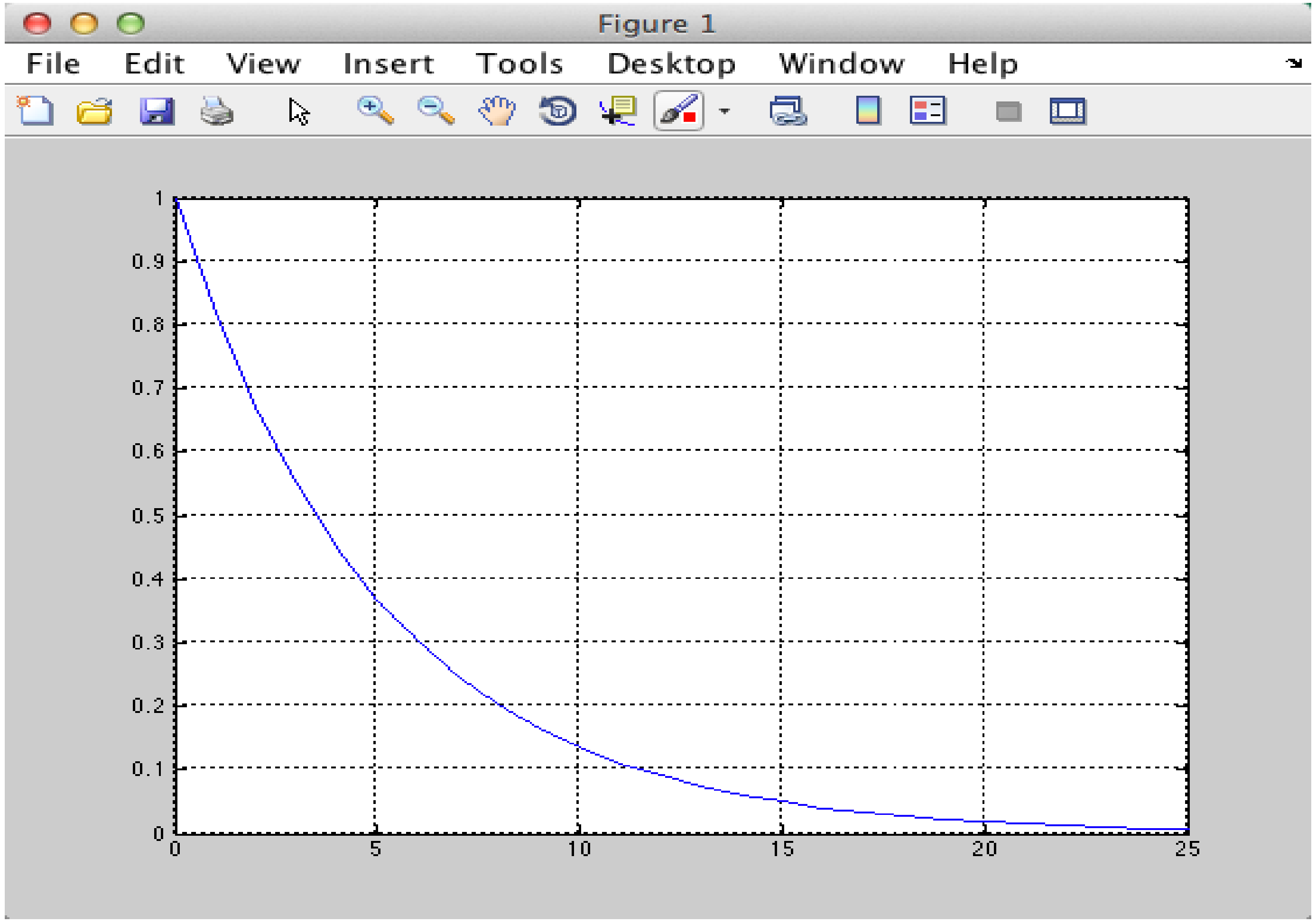
$x(0) = 1$

$a = -\frac{1}{T}$

$0 \le t \le 25$

```
T = 5;
a = -1/T;
x0 = 1;
t = [0:1:25];

x = exp(a*t)*x0;

plot(t,x);
grid
```

Problem with this method:
We need to solve the ODE before we can plot it!!

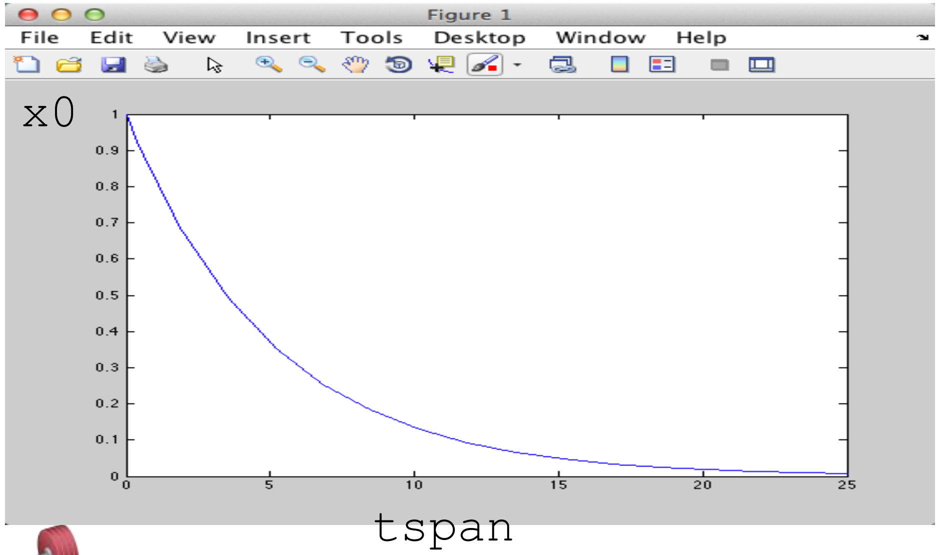


Using ODE Solvers in MATLAB

Example: $\dot{x} = ax$

Step 1: Define the differential equation as a MATLAB function (`mydiff.m`):

```
function dx = mydiff(t,x)
T = 5;
a = -1/T;
dx = a*x;
```



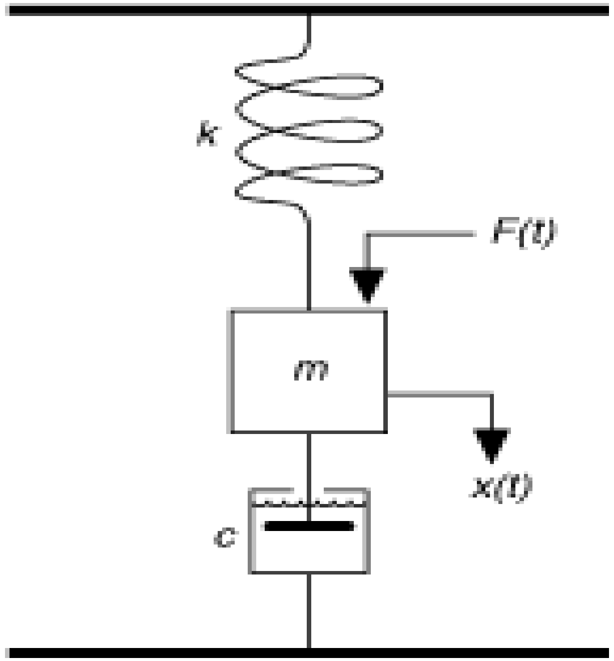
Step 2: Use one of the built-in ODE solver (ode23, ode45, ...) in a Script.

```
clear
clc

tspan = [0 25];
x0 = 1;

[t,x] = ode23(@mydiff,tspan,x0);
plot(t,x)
```

Students: Try this example. Do you get the same result?

Higher Order ODEs

Mass-Spring-Damper System



Example (2.order differential equation):

$$\boxed{\ddot{x} = -\frac{k}{m}x - \frac{c}{m}\dot{x} + \frac{1}{m}F}$$

x – position, $\dot{x}$ – speed/velocity, $\ddot{x}$ – acceleration

c - damping constant, m - mass, k - spring constant, F – force

In order to use the ODEs in MATLAB we need reformulate a higher order system into a system of first order differential equations

Higher Order ODEs

Mass-Spring-Damper System:

$$\boxed{\ddot{x} = -\frac{k}{m}x - \frac{c}{m}\dot{x} + \frac{1}{m}F}$$

In order to use the ODEs in MATLAB we need reformulate a higher order system into a system of first order differential equations

We set:

$$\boxed{\begin{gathered} x_1 = x \\ x_2 = \dot{x} = \dot{x}_1 \end{gathered}}$$

This gives:

$$\boxed{\begin{gathered} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{x} \end{gathered}}$$

Finally:

$$\boxed{\begin{gathered} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}F \end{gathered}}$$

Now we are ready to solve the system using MATLAB

Step 1: Define the differential equation as a MATLAB function (`mass_spring_damper_diff.m`):

```
function dx = mass_spring_damper_diff(t,x)

k = 1;
m = 5;
c = 1;
F = 1;

dx = zeros(2,1); %Initialization

dx(1) = x(2);
dx(2) = -(k/m)*x(1)-(c/m)*x(2)+(1/m)*F;
```

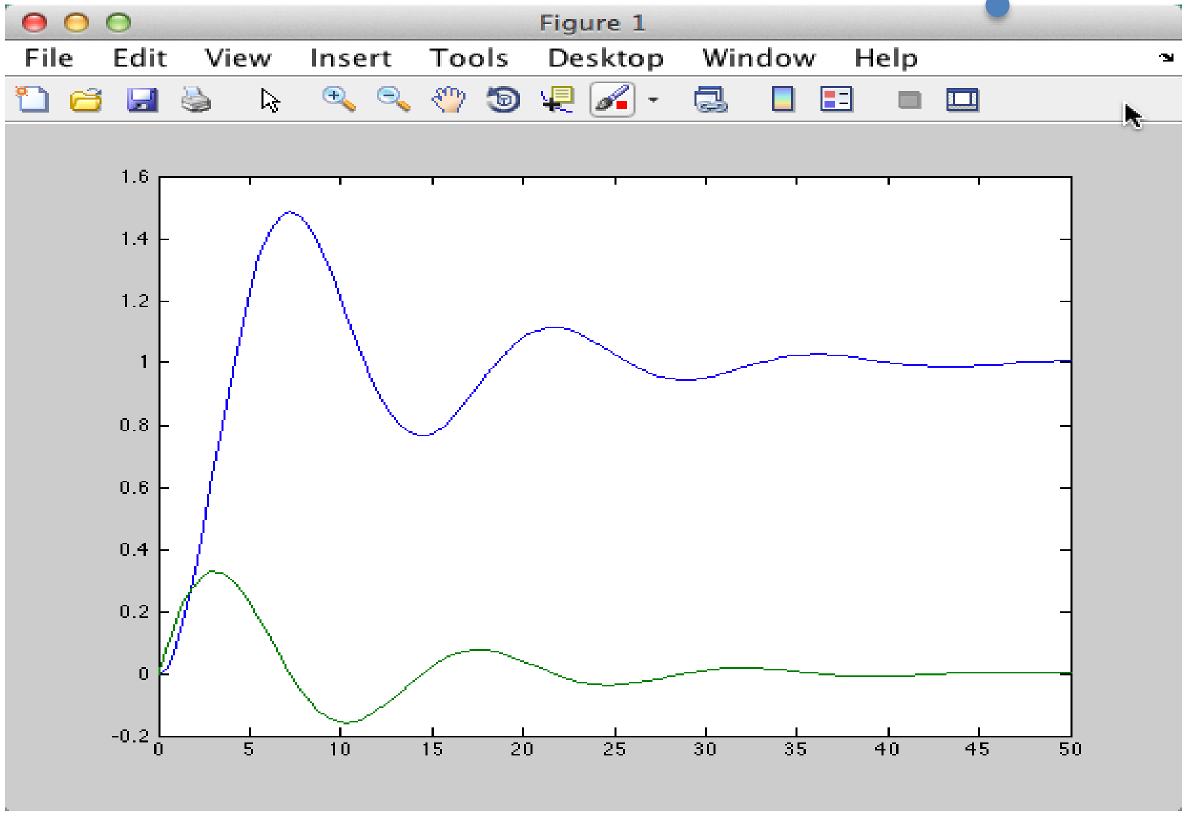
Students: Try with different values for k, m, c and F

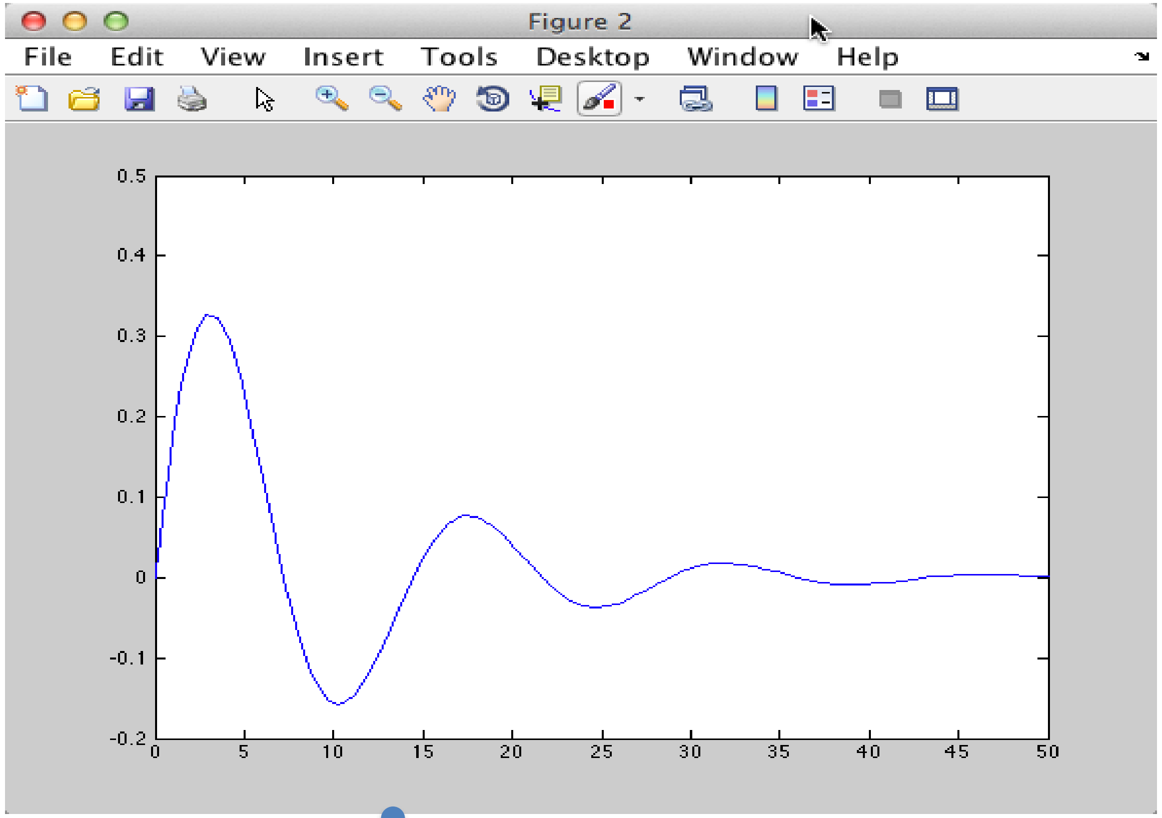
Students: Try this example

Step 2: Use the built-in ODE solver in a script.

```
clear
clc

tspan = [0 50];
x0 = [0;0];

[t,x] = ode23(@mass_spring_damper_diff,tspan,x0);
plot(t,x)
```

```
...
[t,x] = ode23(@mass_spring_damper_diff,tspan,x0);
plot(t,x)
```



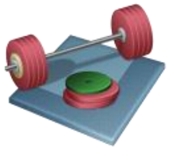



```
...
[t,x] = ode23(@mass_spring_damper_diff,tspan,x0);
plot(t,x(:,2))
```

For greater flexibility we want to be able to change the parameters k, m, c, and F without changing the function, only changing the script. A better approach would be to pass these parameters to the function instead.

Step 1: Define the differential equation as a MATLAB function (`mass_spring_damper_diff.m`):

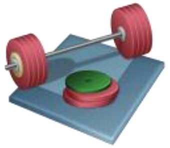
```
function dx = mass_spring_damper_diff(t,x, param)

k = param(1);
m = param(2);
c = param(3);
F = param(4);

dx = zeros(2,1);

dx(1) = x(2);
dx(2) = -(k/m)*x(1) - (c/m)*x(2) + (1/m)*F;
```



Students: Try this example

Step 2: Use the built-in ODE solver in a script:

```
clear
clc
close all

tspan = [0 50];
x0 = [0;0];

k = 1;
m = 5;
c = 1;
F = 1;
param = [k, m, c, F];

[t,x] = ode23(@mass_spring_damper_diff,tspan,x0, [], param);
plot(t,x)
```

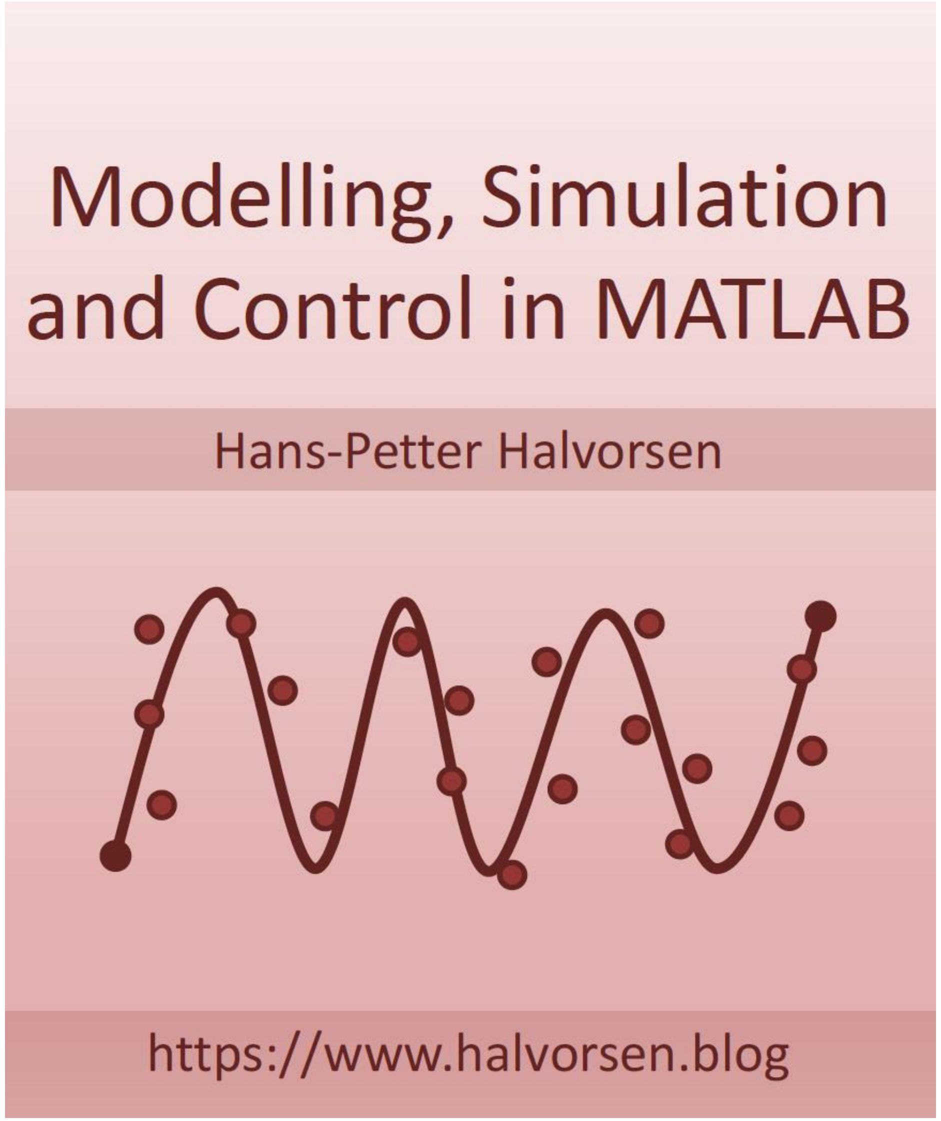


Students: Try this example

MATLAB for Students

Whats next?

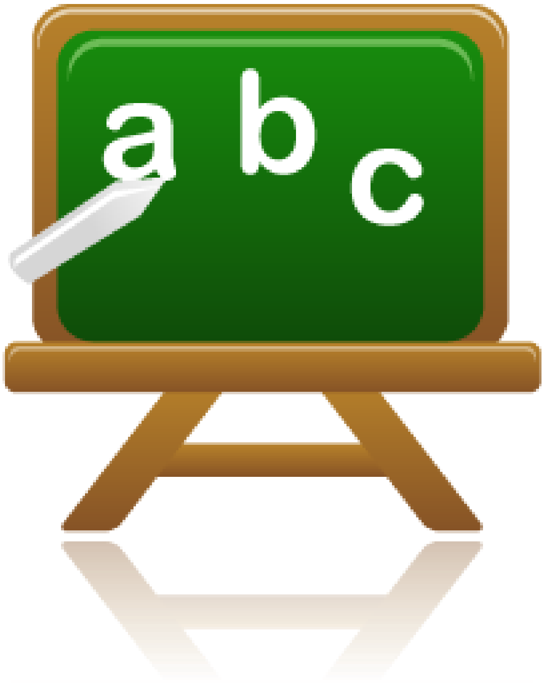
Learning by Doing!



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



Discrete Systems

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$

Discrete Systems

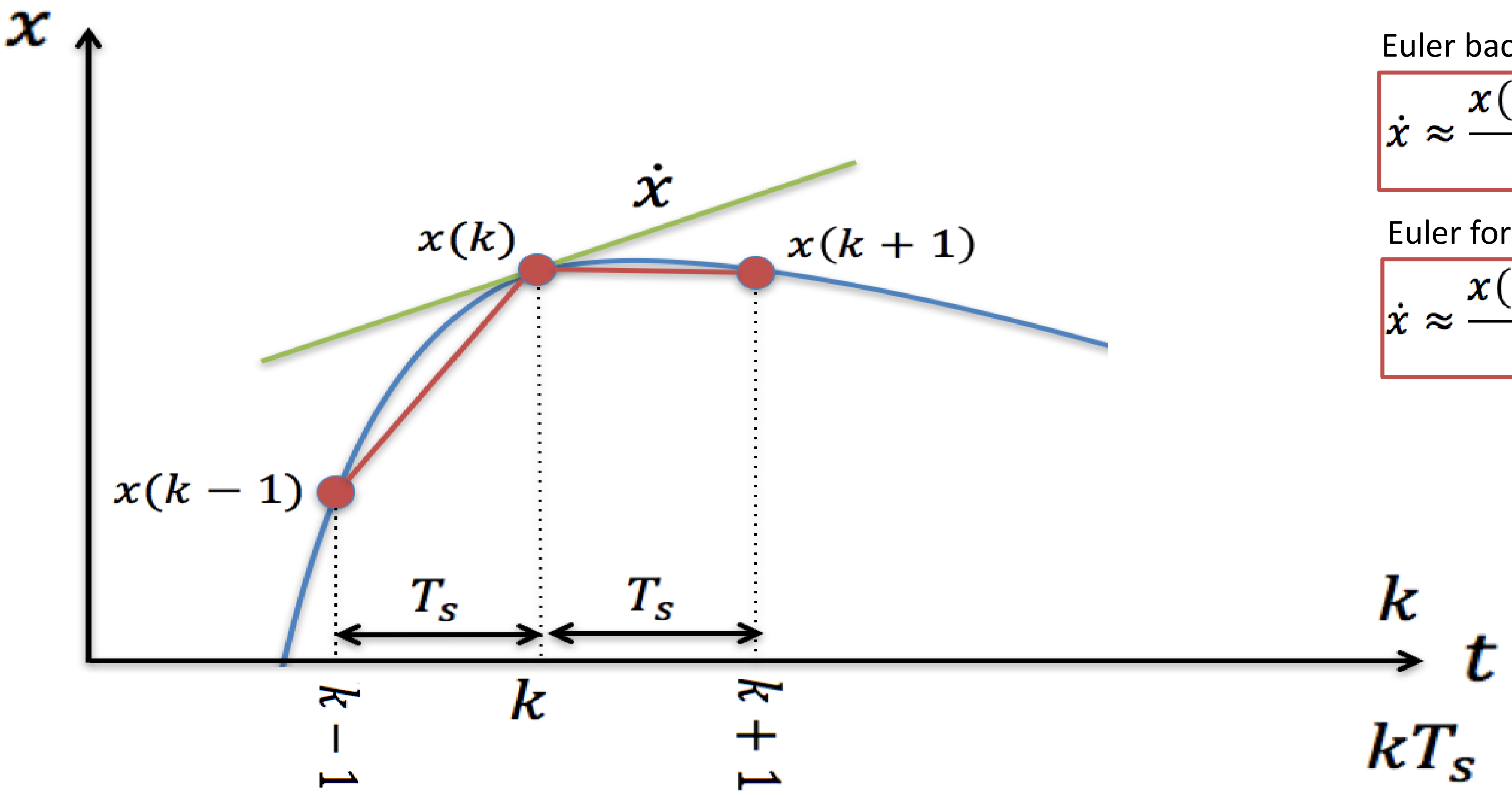
- When dealing with computer simulations, we need to create a discrete version of our system.
- This means we need to make a discrete version of our continuous differential equations.
- Actually, the built-in ODE solvers in MATLAB use different discretization methods
- Interpolation, Curve Fitting, etc. is also based on a set of discrete values (data points or measurements)
- The same with Numerical Differentiation and Numerical Integration
- etc.

Discrete values

x	y
0	15
1	10
2	9
3	6
4	2
5	0

Discrete Systems

Discrete Approximation of the time derivative



Euler backward method:

$$\dot{x} \approx \frac{x(k) - x(k-1)}{T_s}$$

Euler forward method:

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$

Discrete Systems

Discretization Methods

Euler backward method: $$\boxed{\dot{x} \approx \frac{x(k) - x(k-1)}{T_s}}$$ More Accurate!

Euler forward method: $$\boxed{\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}}$$ Simpler to use!

Where T_s is the sampling time, and $x(k+1)$, $x(k)$ and $x(k-1)$ are discrete values.

Other methods are Zero Order Hold (ZOH), Tustin's method, etc.

Discrete Systems

Different Discrete Symbols and meanings

Previous Value: $x(k-1) = x_{k-1} = x(t_{k-1})$

Present Value: $x(k) = x_k = x(t_k)$

Next (Future) Value: $x(k+1) = x_{k+1} = x(t_{k+1})$

Note! Different Notation is used in different litterature!

Example:

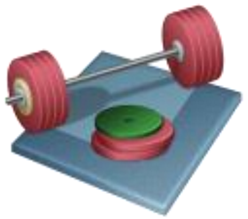
Discrete Systems

Given the following continuous system (differential equation):

$$\dot{x} = -ax + bu \qquad x(k+1) = ?$$

Where *u* may be the Control Signal from e.g., a PID Controller

We will use the Euler forward method :
$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$



Students: Find the discrete differential equation (pen and paper) and then simulate the system in MATLAB, i.e., plot the Step Response of the system. Tip! Use a for loop

Set $a = 0.25$ and $b = 2$

Solution:

Discrete Systems

Given the following continuous system:

$$\dot{x} = -ax + bu \qquad\qquad x(k+1) = ?$$

We will use the Euler forward method :

1. $$\frac{x(k+1) - x(k)}{T_s} = -ax(k) + bu(k)$$

2. $$x(k+1) = x(k) + T_s[-ax(k) + bu(k)]$$

3. $$x(k+1) = x(k) - T_s a x(k) + T_s b u(k)$$

4. $$\underline{\underline{x(k+1) = (1 - T_s a)x(k) + T_s b u(k)}}$$

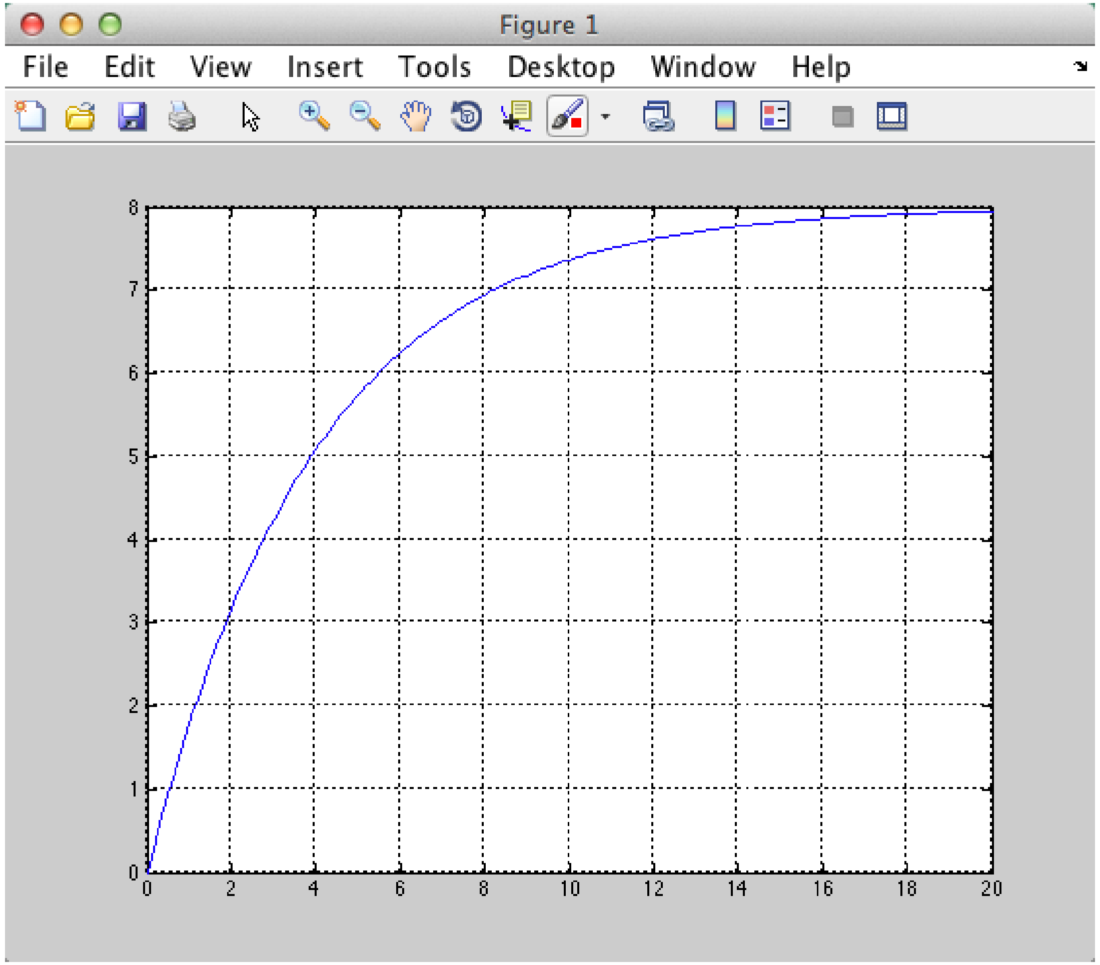
Solution:

Discrete Systems

Students: Try this example

MATLAB Code:

```
% Simulation of discrete model
clear, clc, close all

% Model Parameters
a = 0.25;b = 2;

% Simulation Parameters
Ts = 0.1; %s
Tstop = 20; %s
uk = 1; % Step in u
x(1) = 0; % Initial value

% Simulation
for k=1:(Tstop/Ts)
    x(k+1) = (1-a*Ts).*x(k) + Ts*b*uk;
end

% Plot the Simulation Results
k=0:Ts:Tstop;
plot(k, x)
grid on
```



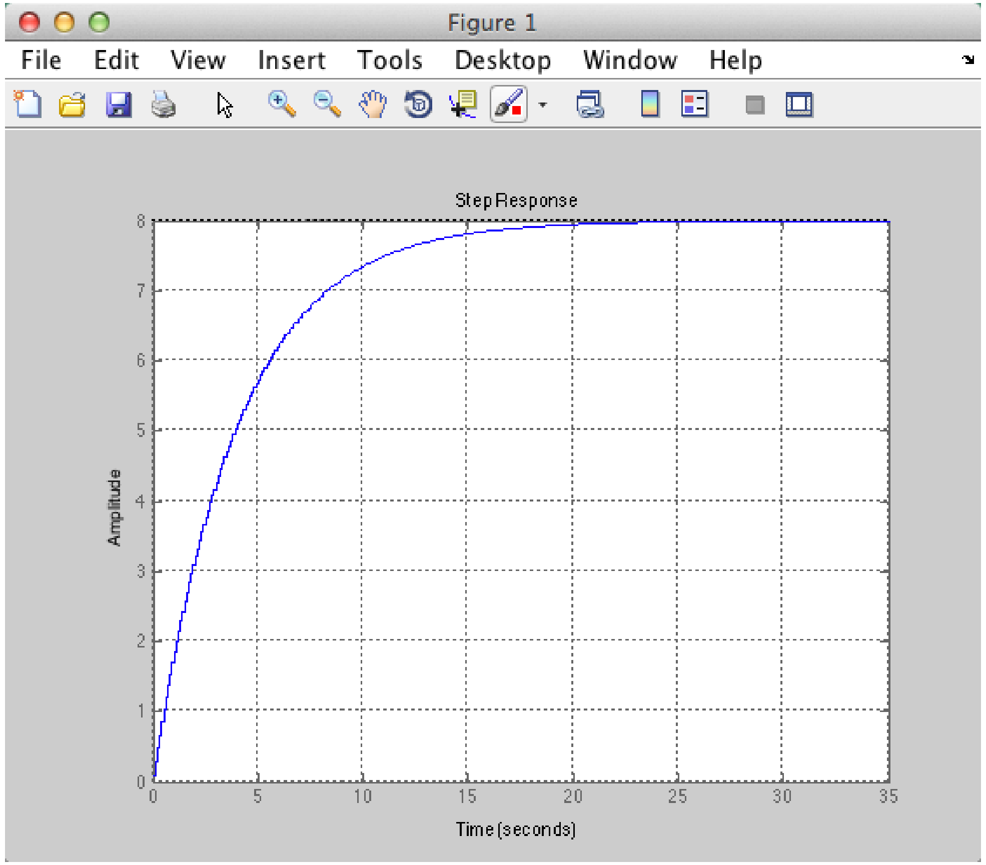
Students: An alternative solution is to use the built-in function **c2d()** (convert from continous to discrete). Try this function and see if you get the same results.

Solution:

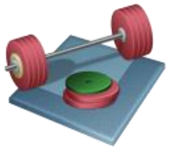
Discrete Systems

MATLAB Code:

```
% Find Discrete model
clear, clc, close all

% Model Parameters
a = 0.25;
b = 2;
Ts = 0.1; %s

% State-space model
A = [-a];
B = [b];
C = [1];
D = [0];

model = ss(A,B,C,D)
model_discrete = c2d(model, Ts, 'forward')
step(model_discrete)
grid on
```



Euler Forward method

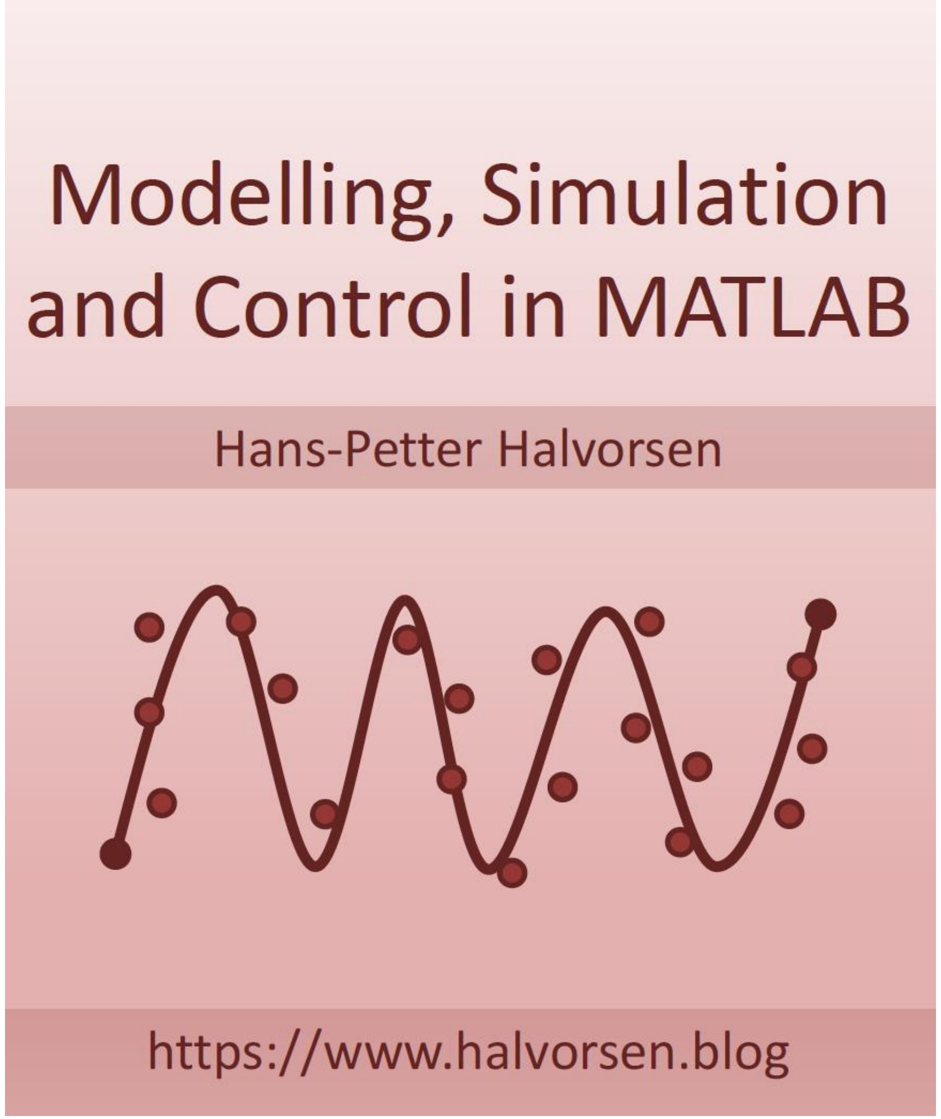


Students: Try this example

MATLAB for Students

Whats next?

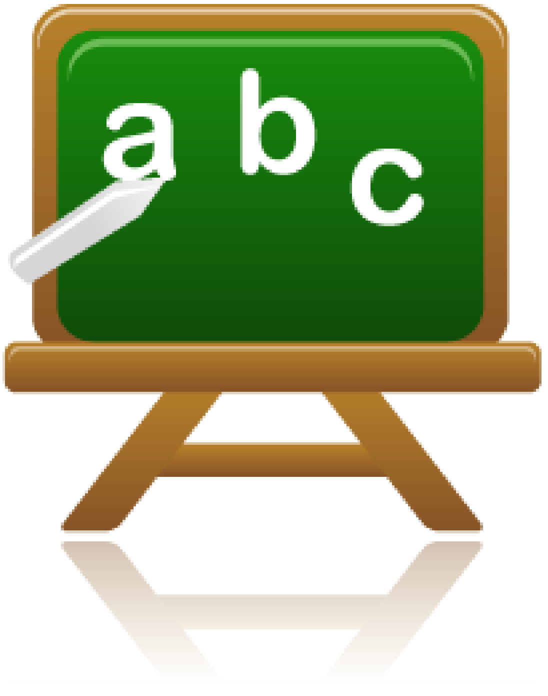
Learning by Doing!



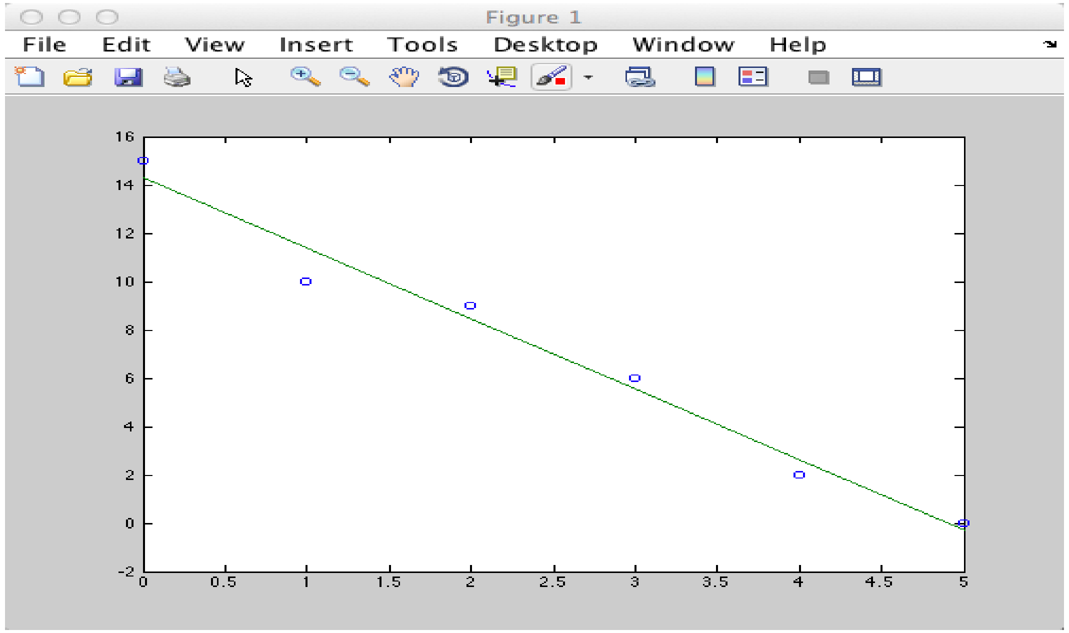
Self-paced Tutorials with lots of Exercises and Video resources

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

Lesson 3



- Interpolation
- Curve Fitting



Example

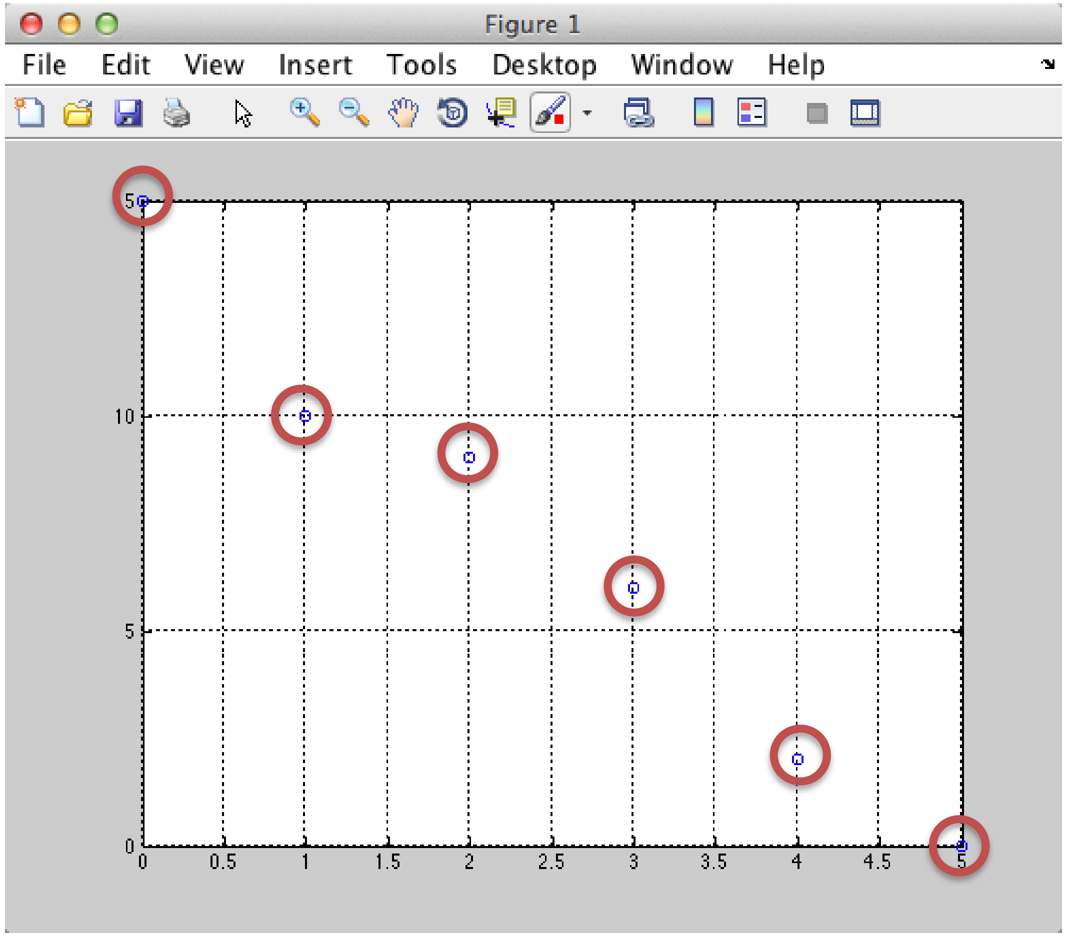
Interpolation

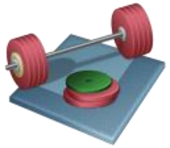
Given the following Data Points:

x	y
0	15
1	10
2	9
3	6
4	2
5	0

(Logged Data from a given Process)

```
x=0:5;
y=[15, 10, 9, 6, 2, 0];

plot(x,y ,'o')
grid
```





Students: Try this example

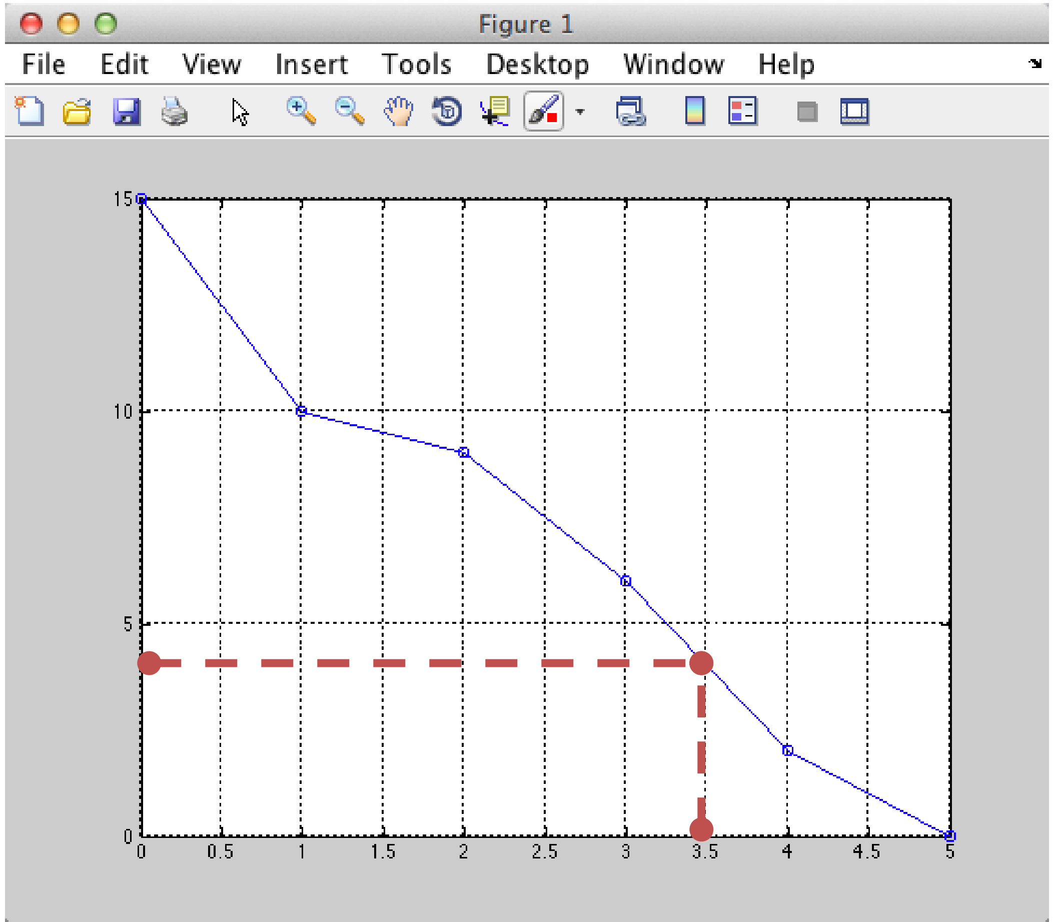
Problem: We want to find the interpolated value for, e.g., $x = 3.5$

Interpolation

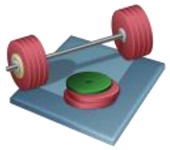
We can use one of the built-in Interpolation functions in MATLAB:

```
x=0:5;
y=[15, 10, 9, 6, 2, 0];

plot(x,y ,'-o')
grid on

new_x=3.5;
new_y = interp1(x,y,new_x)
```

```
new_y =

    4
```

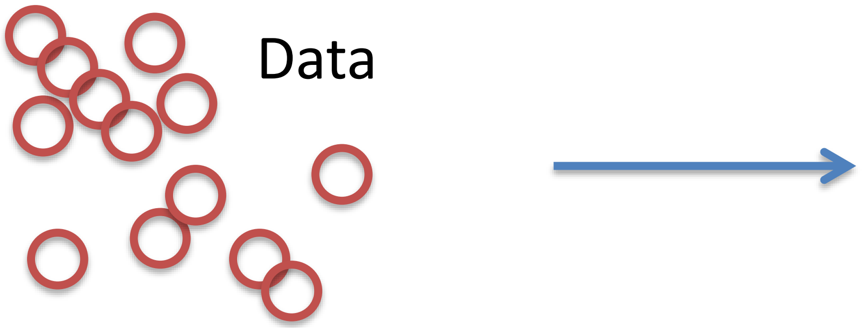


MATLAB gives us the answer 4.
From the plot we see this is a good guess:


Students: Try this example

Curve Fitting

- In the previous section we found interpolated points, i.e., we found values between the measured points using the interpolation technique.
- It would be more convenient to model the data as a mathematical function $y = f(x)$.
- Then we can easily calculate any data we want based on this model.



Mathematical Model

$$y = f(x)$$

Example: # Curve Fitting Linear Regression

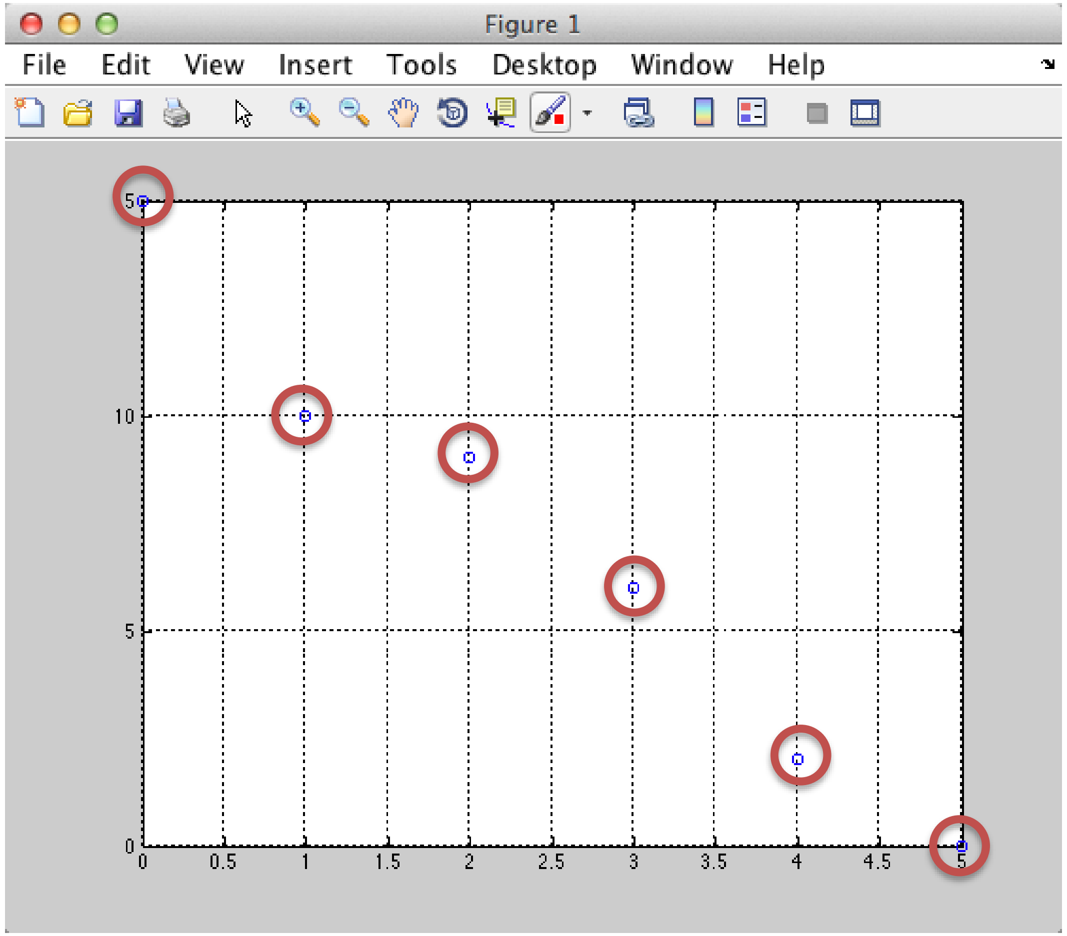
Given the following Data Points:

x	y
0	15
1	10
2	9
3	6
4	2
5	0

Based on the Data Points we create a Plot in MATLAB

```
x=0:5;
y=[15, 10, 9, 6, 2, 0];

plot(x,y ,'o')
grid
```

Students: Try this example



Based on the plot we assume a linear relationship:

$$\boxed{y = ax + b}$$

We will use MATLAB in order to find a and b.

Example

Curve Fitting

Linear Regression

Based on the plot we assume a linear relationship:

$$\boxed{y = ax + b}$$

We will use MATLAB in order to find a and b.

Students: Try this example

```
clear
clc

x=[0, 1, 2, 3, 4 ,5];
y=[15, 10, 9, 6, 2 ,0];
n=1; % 1.order polynomial
p = polyfit(x,y,n)
```

```
p =

    -2.9143    14.2857
```

$$y \approx -2.9x + 14.3$$

Next: We will then plot and validate the results in MATLAB

Example

Curve Fitting

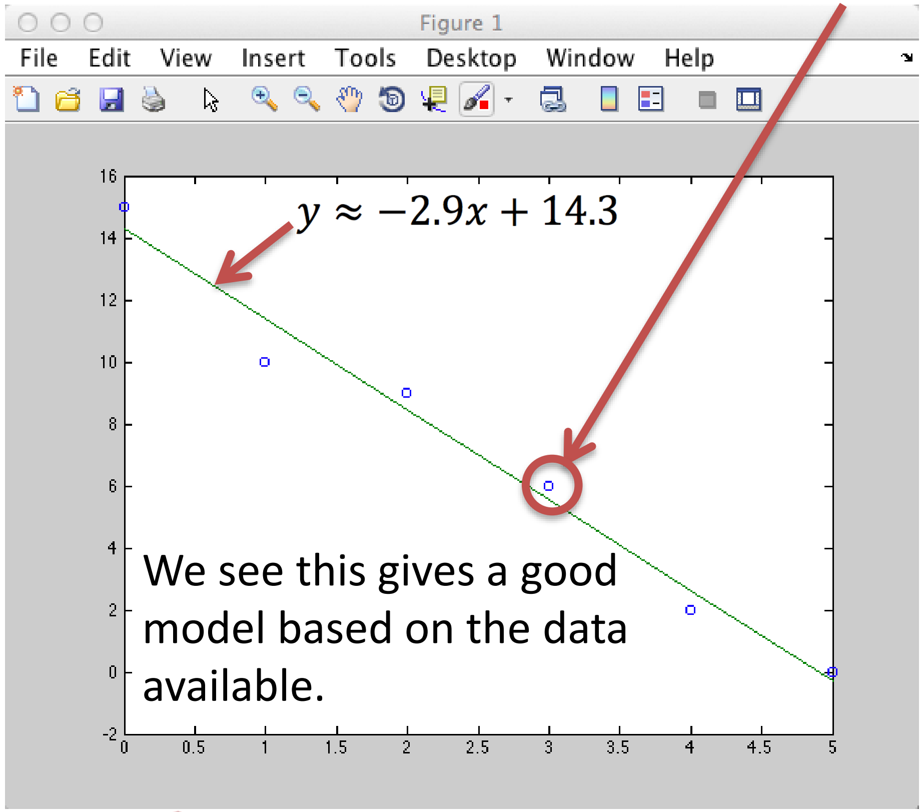
Linear Regression

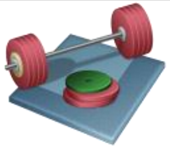
$$y \approx -2.9x + 14.3$$

```
clear
clc
close all

x=[0, 1, 2, 3, 4 ,5];
y=[15, 10, 9, 6, 2 ,0];
n=1; % 1.order polynomial
p=polyfit(x,y,n);

a=p(1);
b=p(2);

ymodel = a*x+b;

plot(x,y,'o',x,ymodel)
```

We will plot and validate the results in MATLAB

x	y
0	15
1	10
2	9
3	6
4	2
5	0




Students: Try this example

x	y
0	15
1	10
2	9
3	6
4	2
5	0

Curve Fitting

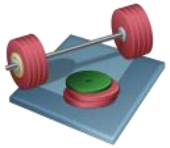
Linear Regression

Problem: We want to find the interpolated value for, e.g., *x=3.5*

3 ways to do this:

- Use the `interp1` function (shown earlier)
- Implement *y=-2.9+14.3* and calculate *y(3.5)*
- Use the **`polyval`** function

```
... (see previus examples)

new_x=3.5;

new_y = interp1(x,y,new_x)

new_y = a*new_x + b

new_y = polyval(p, new_x)
```



Students: Try this example

Curve Fitting

Polynomial Regression

$$\boxed{y(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n}$$

1.order: $y(x) = ax + b$

```
p = polyfit(x,y,1)
```

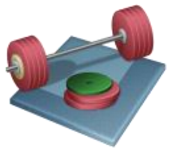
2.order: $y(x) = ax^2 + bx + c$

```
p = polyfit(x,y,2)
```

3.order: $y(x) = ax^3 + bx^2 + cx + d$

```
p = polyfit(x,y,3)
```

etc.

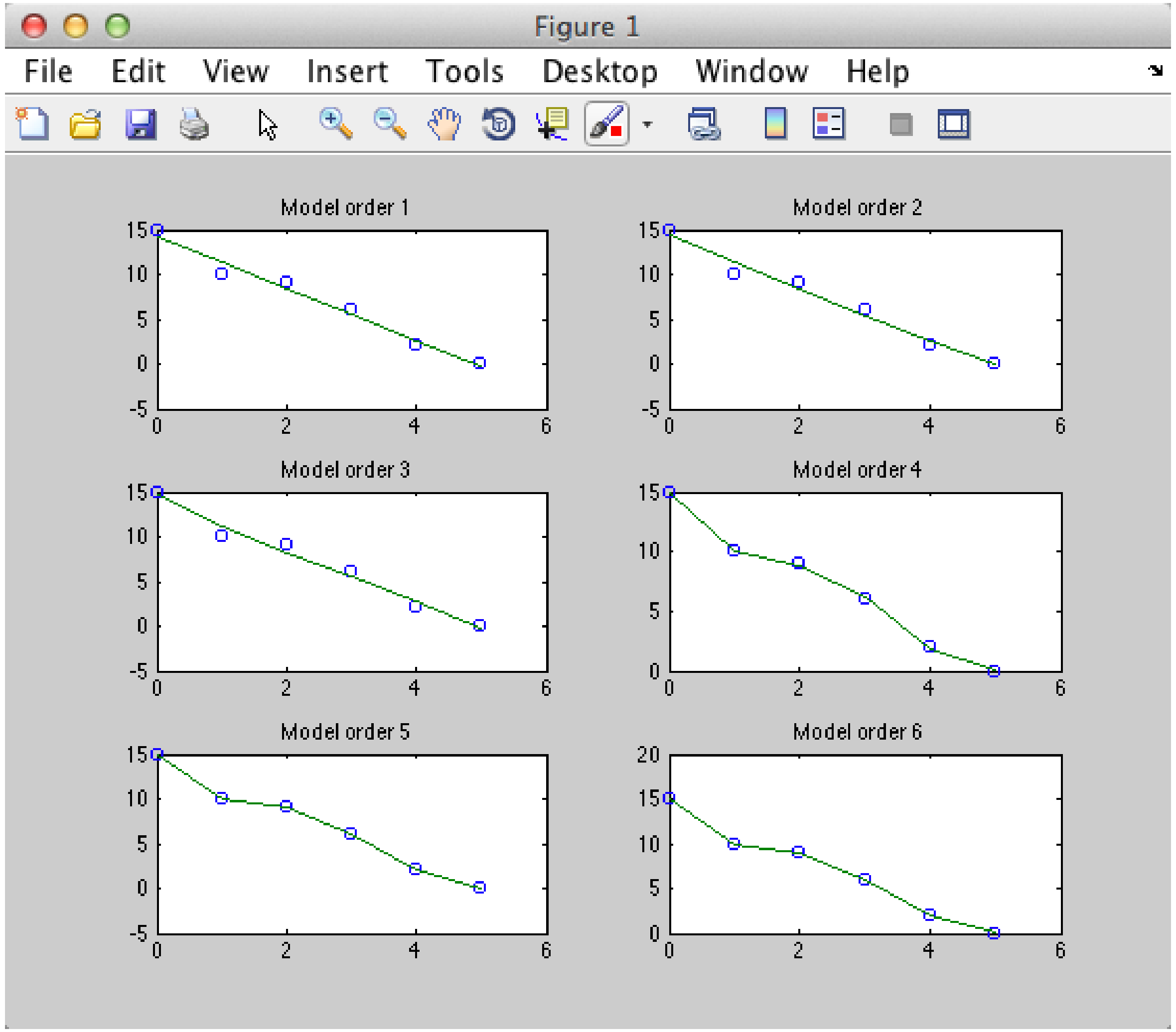


Students: Try to find models based on the given data using different orders (1. order – 6. order models).
Plot the different models in a subplot for easy comparison.

x	y
0	15
1	10
2	9
3	6
4	2
5	0

Curve Fitting

```
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
```

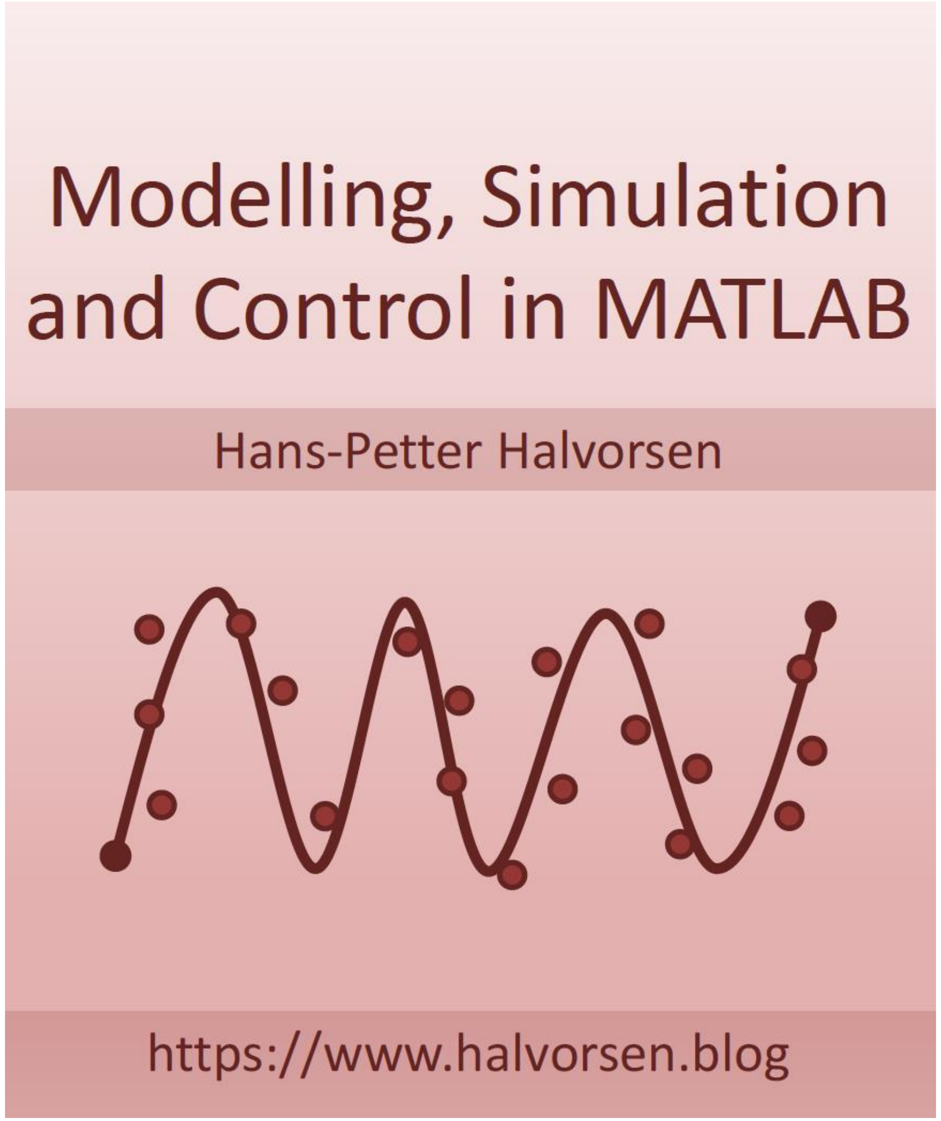


- As expected, the higher order models match the data better and better.
- Note! The fifth order model matches exactly because there were only six data points available.
- n > 5 makes no sense because we have only 6 data points

MATLAB for Students

Whats next?

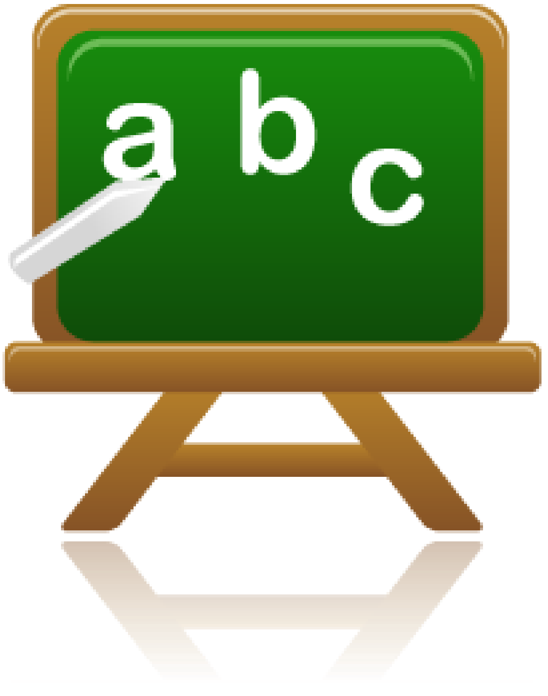
Learning by Doing!



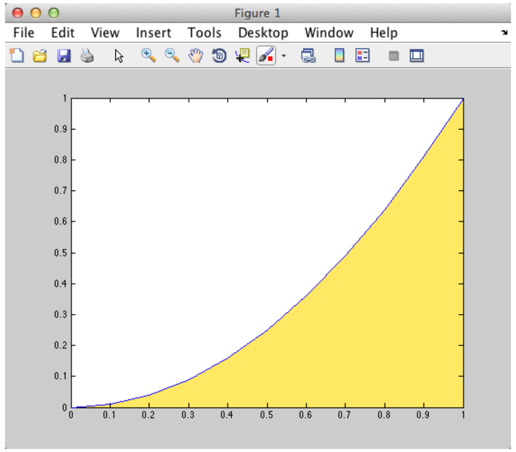
Self-paced Tutorials with lots of Exercises and Video resources

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

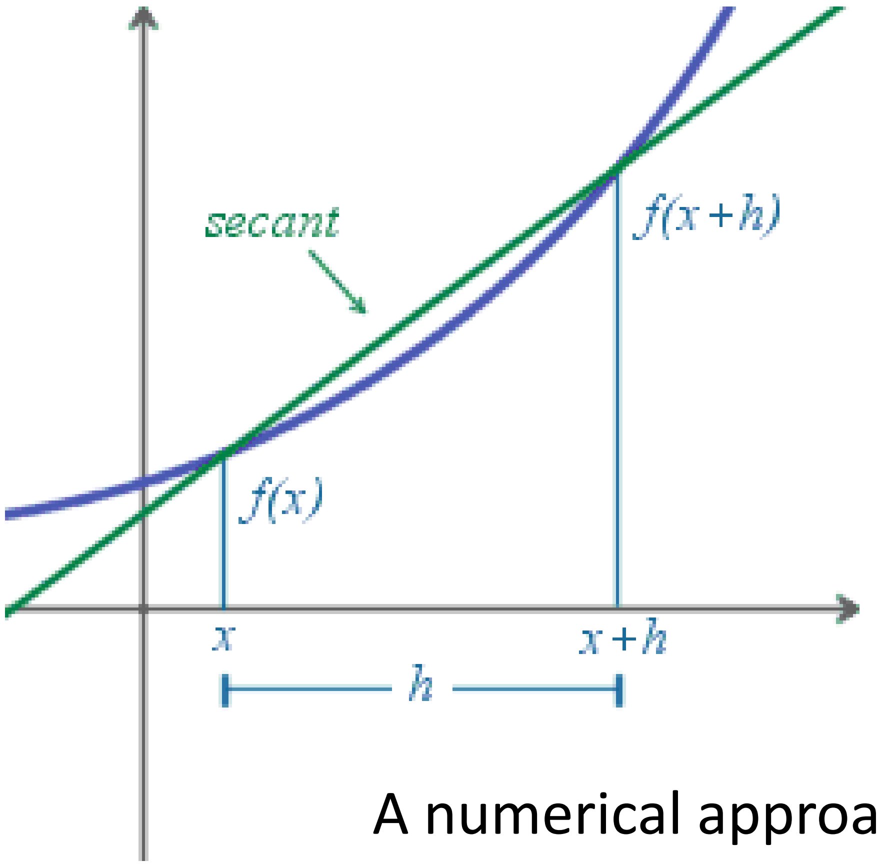
Lesson 4



- Numerical Differentiation
- Numerical Integration



Numerical Differentiation



$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

A numerical approach to the derivative of a function *y=f(x)* is:

$$\frac{dy}{dx} = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

Note! We will use MATLAB in order to find the <u>numeric</u> solution – not the analytic solution

Example:

Numerical Differentiation

$$y(x) = x^2$$

We know for this simple example that the exact analytical solution is:

$$\frac{dy}{dx} = 2x$$

Given the following values:

x	y
-2	4
-1	1
0	0
1	1
2	4

$$\frac{dy}{dx} = ?$$

$$\frac{dy}{dx}(x = -2) = -4$$

$$\frac{dy}{dx}(x = -1) = -2$$

$$\frac{dy}{dx}(x = 0) = 0$$

$$\frac{dy}{dx}(x = 1) = 2$$

$$\frac{dy}{dx}(x = 2) = 4$$

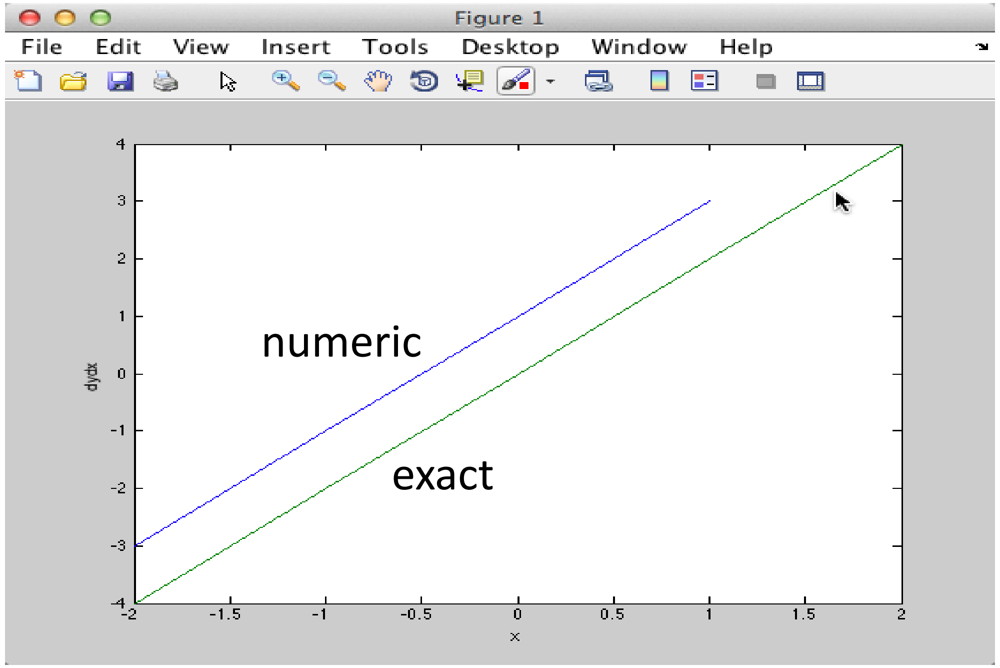
Example:

Numerical Differentiation

$$y(x) = x^2 \qquad \frac{dy}{dx} = ?$$

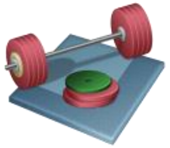
MATLAB Code:

```
x=-2:2;
y=x.^2;

% Exact Solution
dydx_exact = 2*x;

% Numerical Solution
dydx_num = diff(y)./diff(x);

% Compare the Results
dydx = [[dydx_num, NaN]', dydx_exact']
plot(x,[dydx_num, NaN]', x, dydx_exact')
```



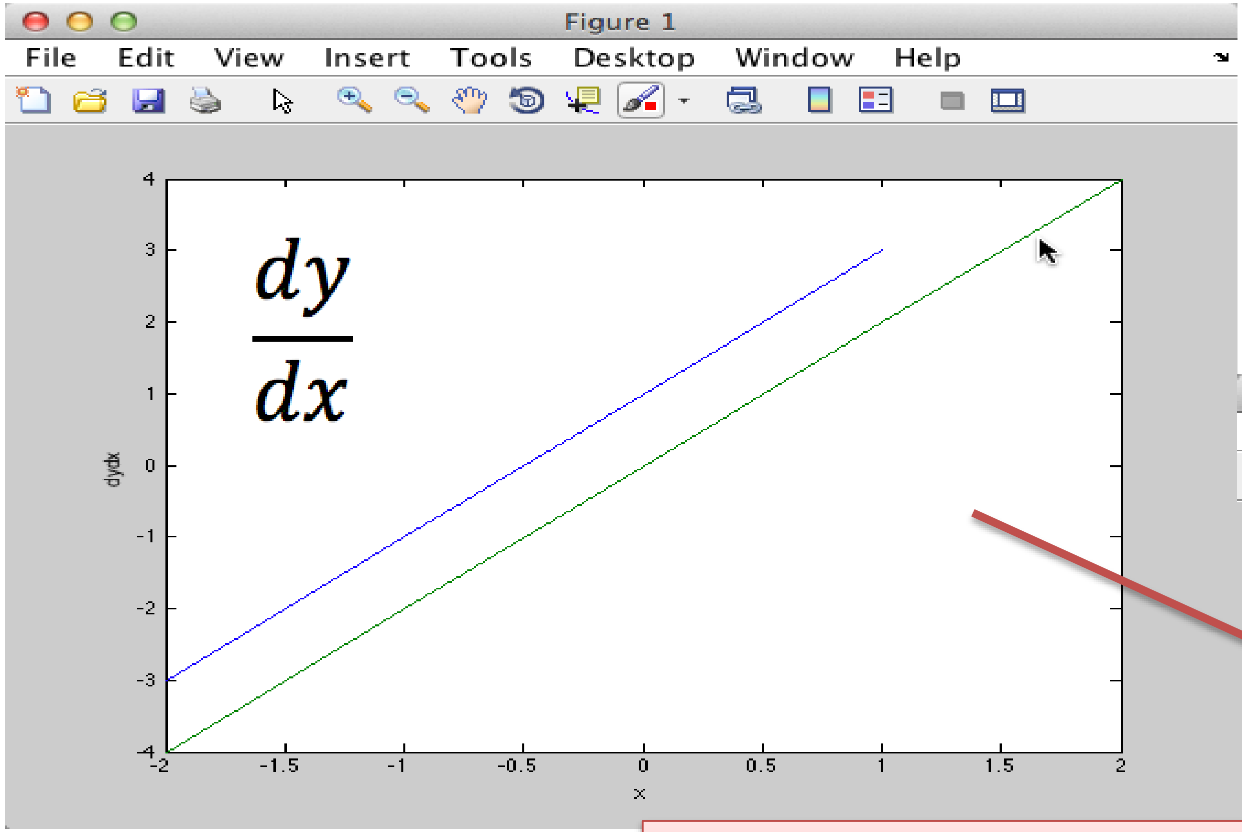
Numerical Solution | Exact Solution

```
dydx =

    -3    -4
    -1    -2
     1     0
     3     2
   NaN     4
```

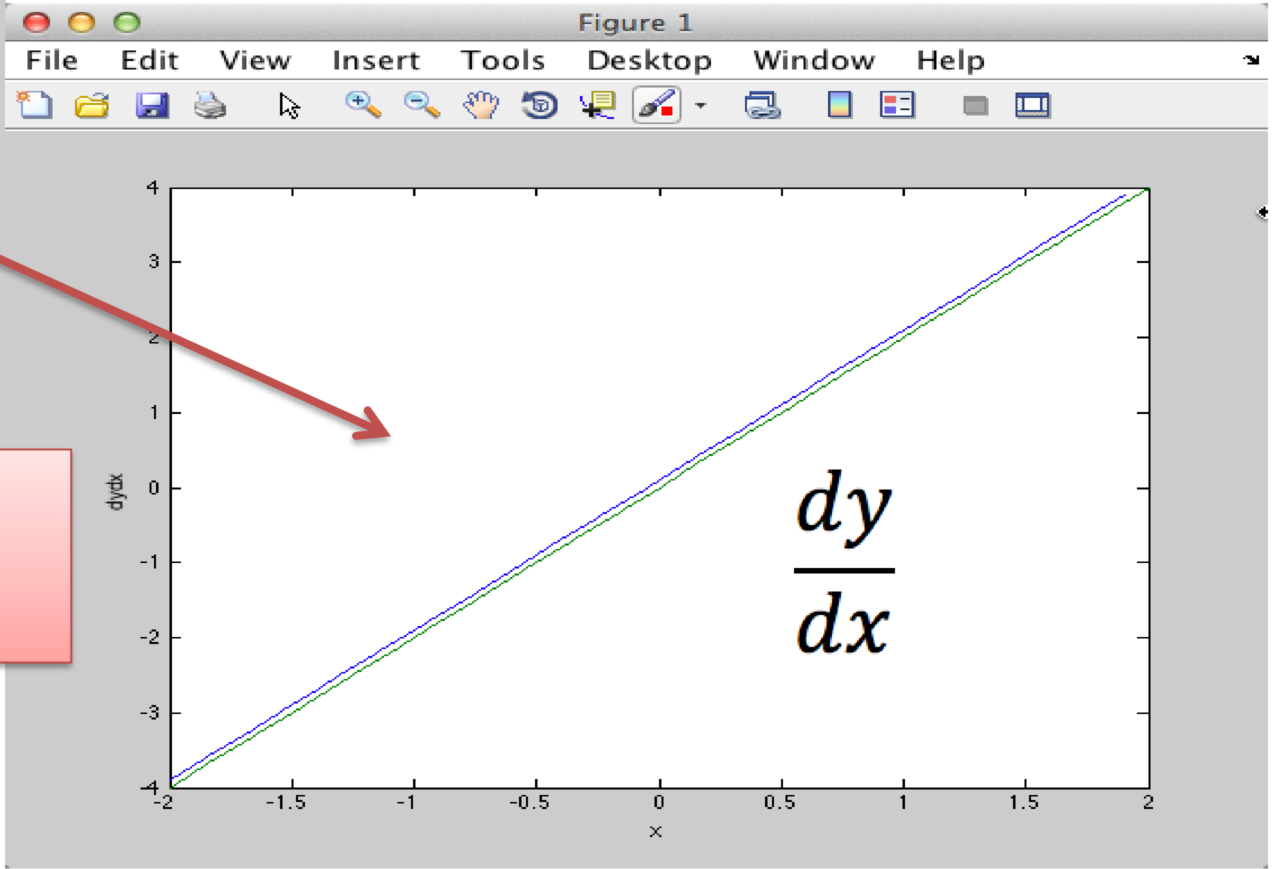


Students: Try this example.
Try also to increase number of data points, `x=-2:0.1:2`

Numerical Differentiation



x=-2:0.1:2



x=-2:2

The results become more accurate when increasing number of data points

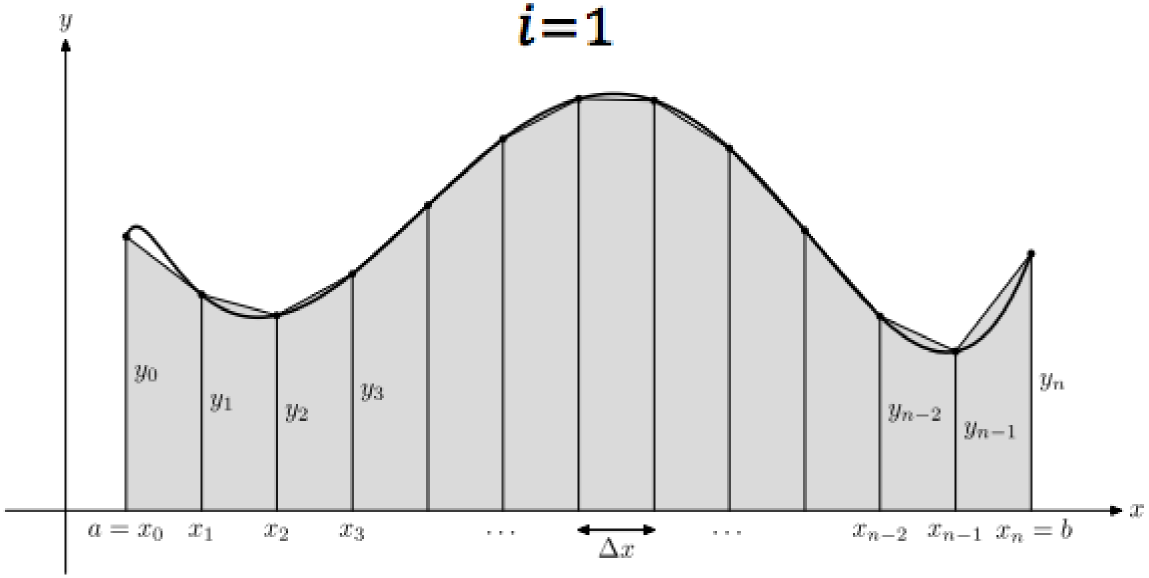
Numerical Integration

$$\int_a^b f(x)dx$$

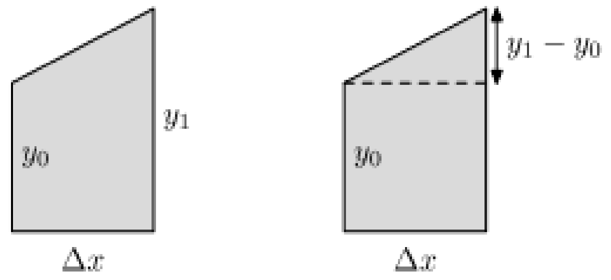
An integral can be seen as the area under a curve.

Given *y=f(x)* the approximation of the Area (*A*) under the curve can be found dividing the area up into rectangles and then summing the contribution from all the rectangles (trapezoid rule):

$$A = \sum_{i=1}^{n-1} (x_{i+1} - x_i) \cdot (y_{i+1} + y_i)/2$$



$$A = y_0\,\Delta x + \frac{1}{2}(y_1 - y_0)\Delta x = \frac{(y_0 + y_1)\Delta x}{2}.$$



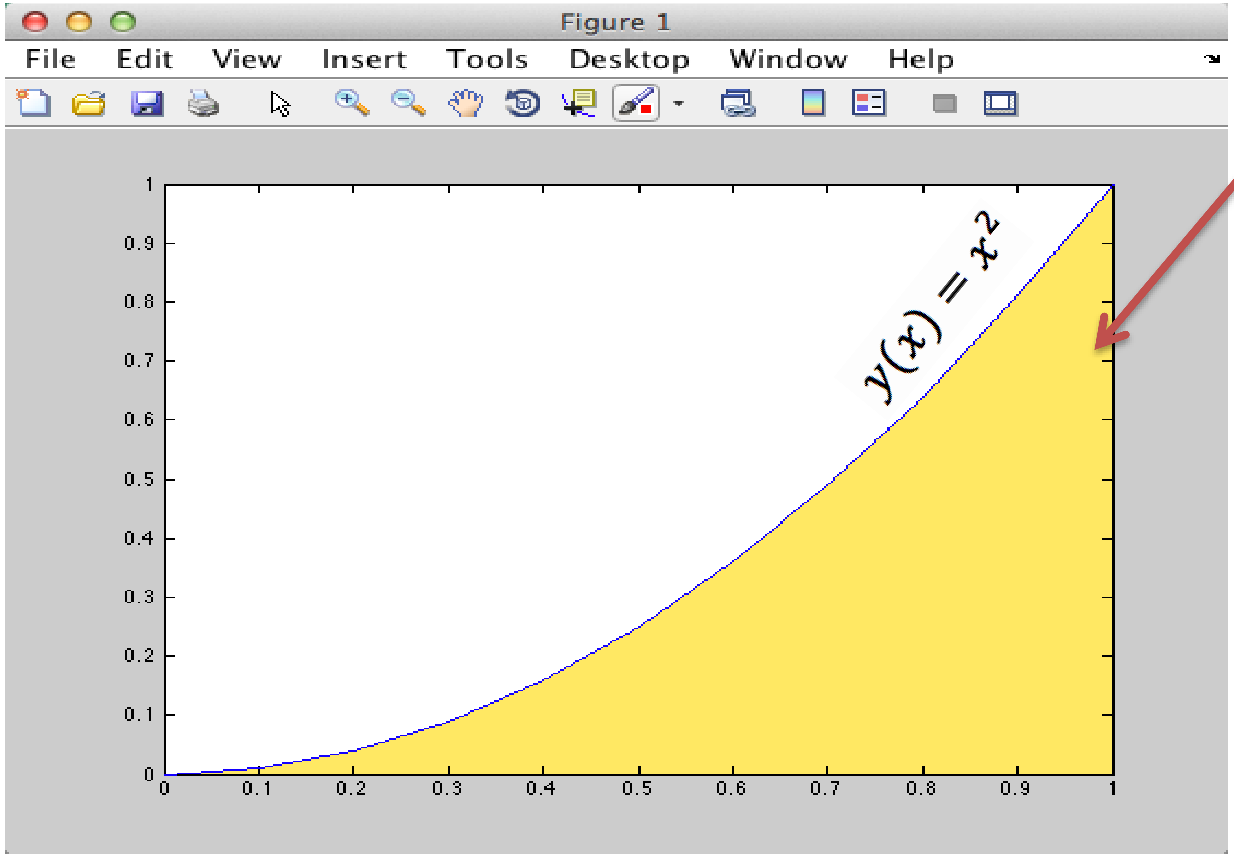
Example:

Numerical Integration

We know that the exact solution is:

$$y(x) = x^2 \rightarrow \int_a^b y(x)\,dx = ? \rightarrow \int_0^a x^2 dx = \frac{a^3}{3}$$

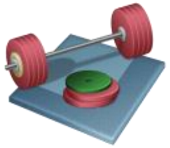
$$\int_0^1 x^2 dx = \frac{1}{3} \approx 0.3333$$



We use MATLAB (trapezoid rule):

```
x=0:0.1:1;
y=x.^2;
plot(x,y)

% Calculate the Integral:
avg_y=y(1:length(x)-1)+diff(y)/2;
A=sum(diff(x).*avg_y)
```

A = 0.3350



Students: Try this example

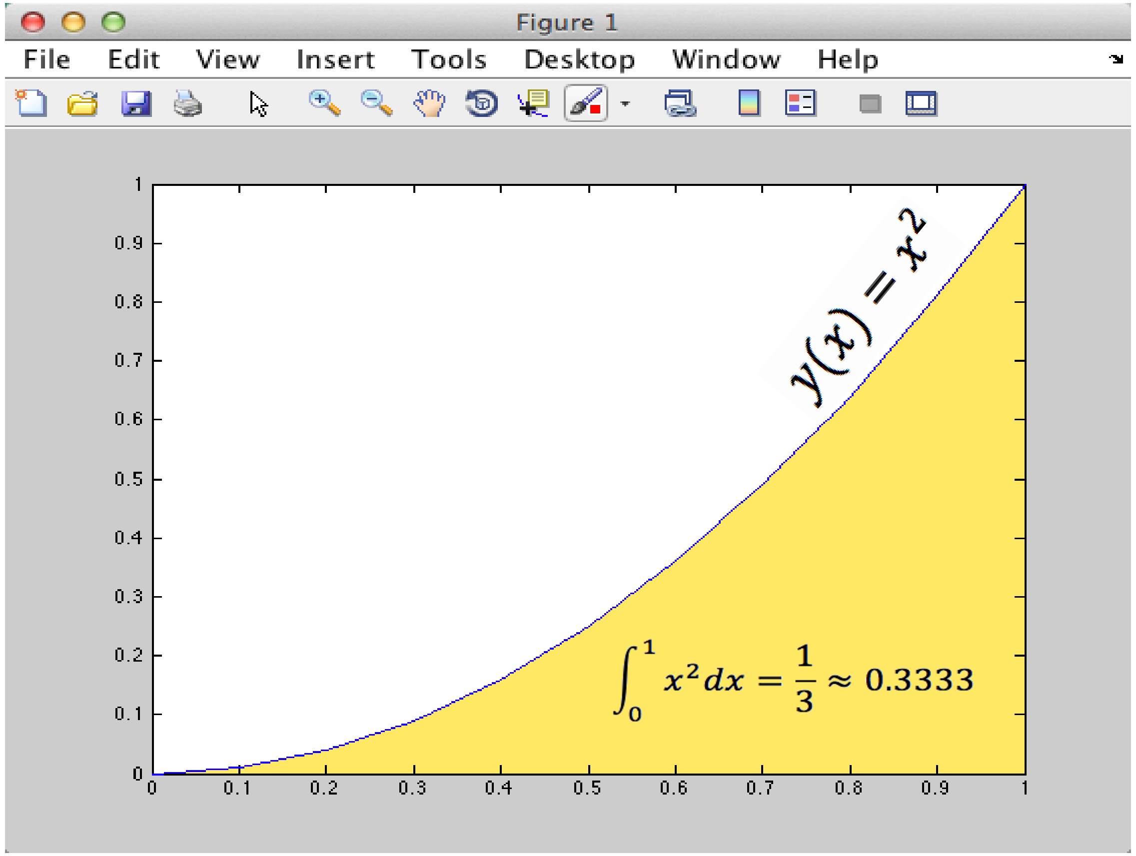
Numerical Integration

Example:

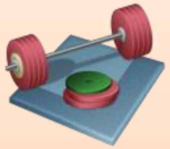
We know that the exact solution is:

$$y(x) = x^2 \rightarrow \int_a^b y(x)\,dx = ? \rightarrow \int_0^a x^2 dx = \frac{a^3}{3}$$

In MATLAB we have several built-in functions we can use for numerical integration:




```
clear
clc
close all

x=0:0.1:1;
y=x.^2;

plot(x,y)

% Calculate the Integral (Trapezoid method):
avg_y = y(1:length(x)-1) + diff(y)/2;
A = sum(diff(x).*avg_y)

% Calculate the Integral (Simpson method):
A = quad('x.^2', 0,1)

% Calculate the Integral (Lobatto method):
A = quadl('x.^2', 0,1)
```

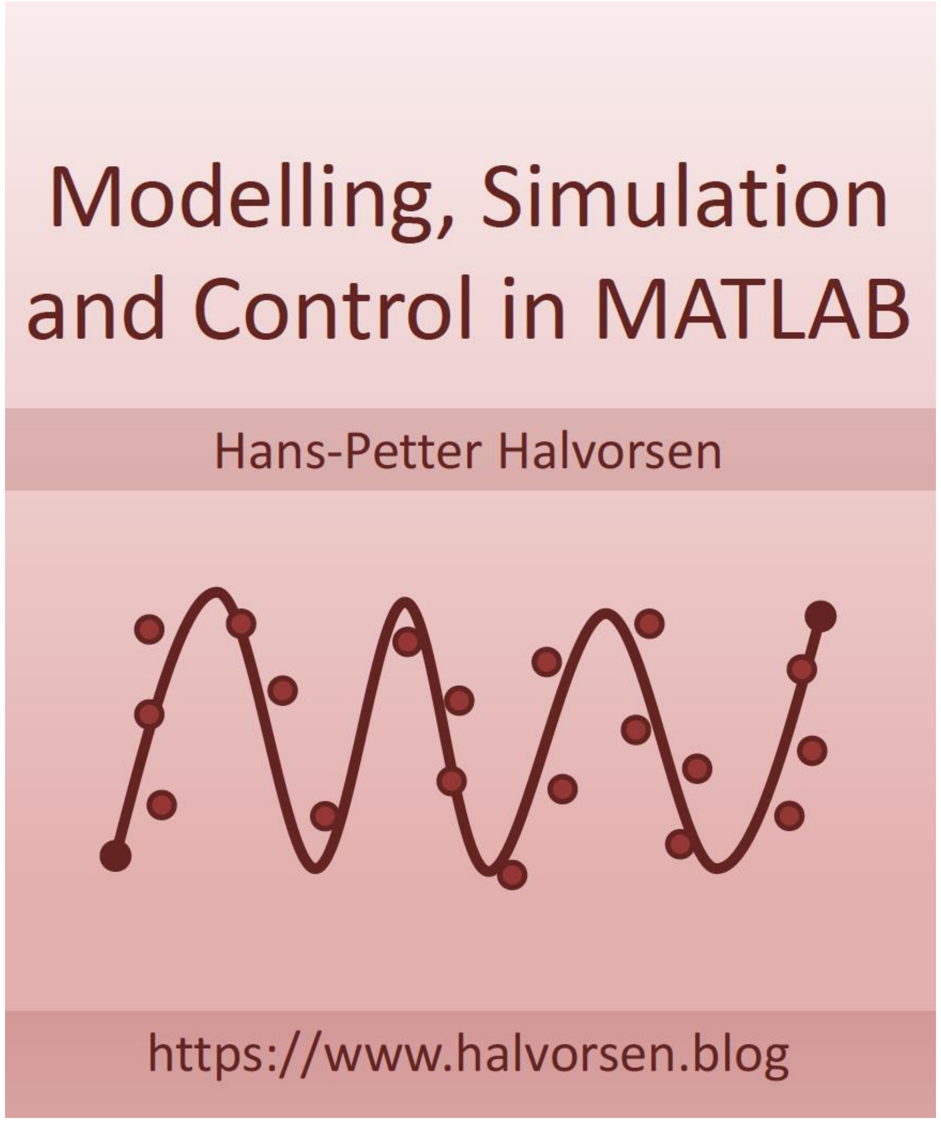


Students: Try this example.
Compare the results.
Which gives the best method?

MATLAB for Students

Whats next?

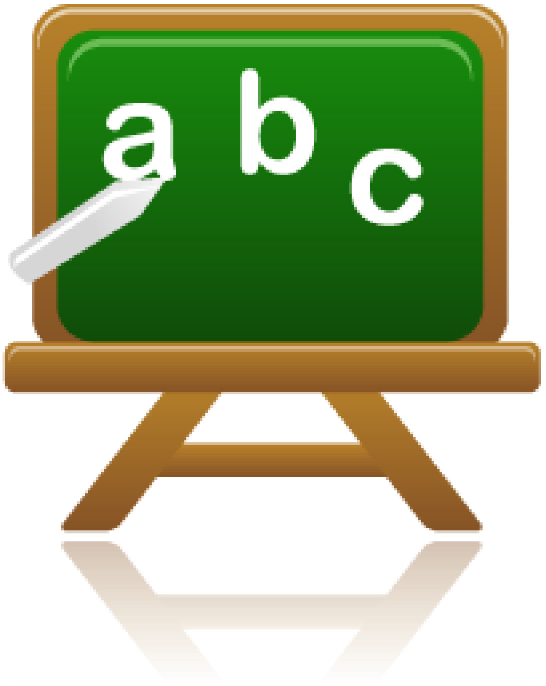
Learning by Doing!



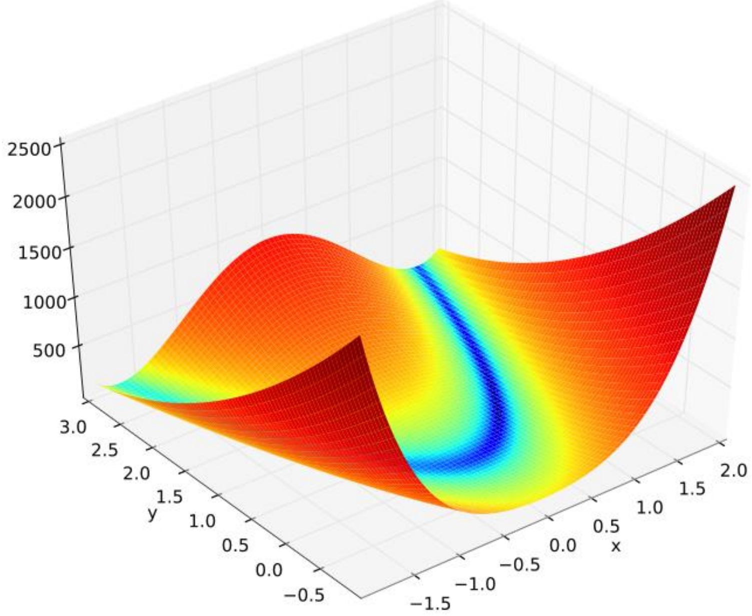
Self-paced Tutorials with lots of Exercises and Video resources

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

Lesson 5



- Optimization



Optimization

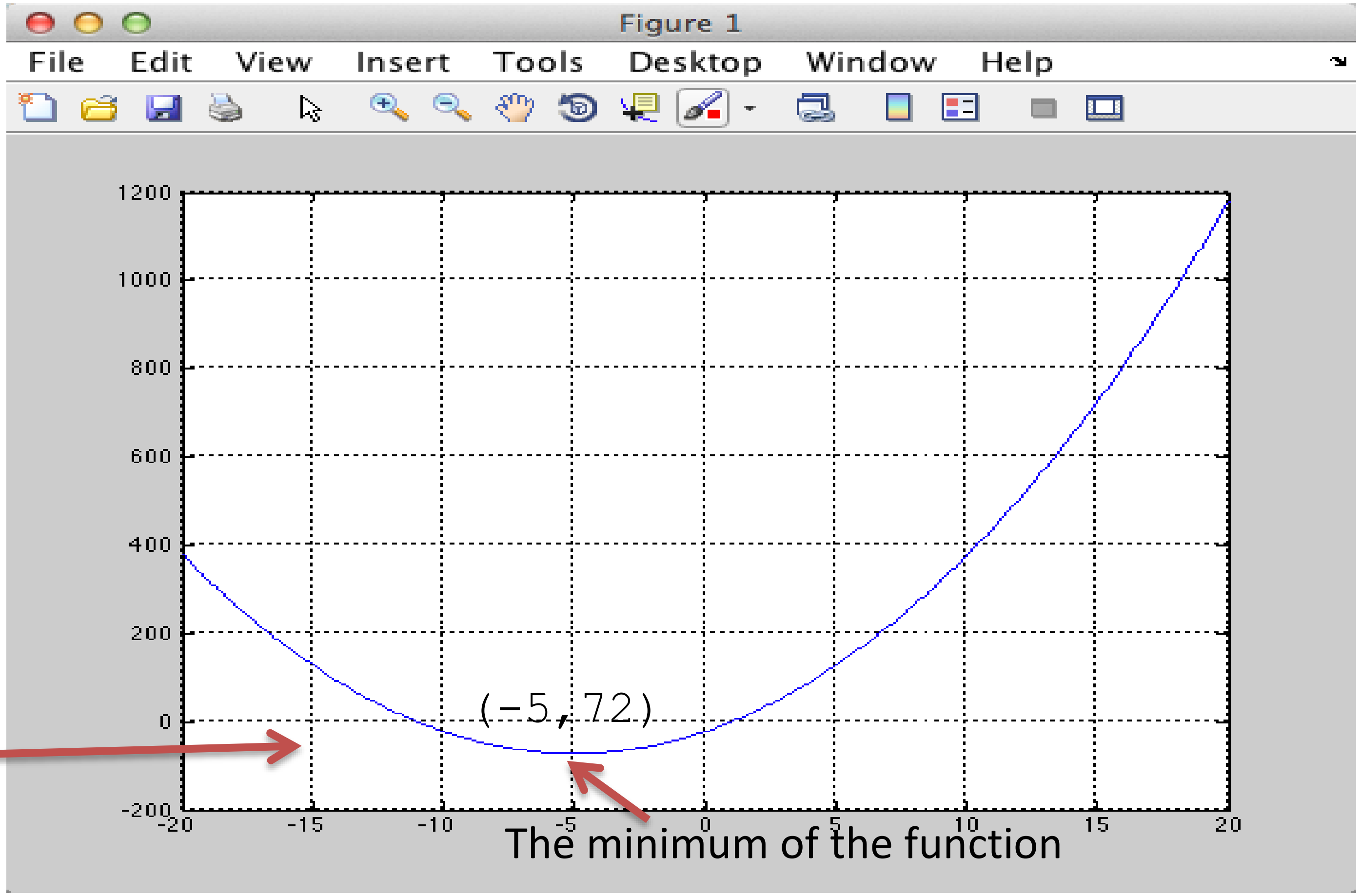
Optimization is important in modelling, control and simulation applications.
Optimization is based on finding the minimum of a given criteria function.

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

Students: Try this example

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

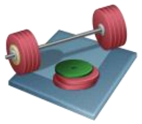
```
clear
clc

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)
```



Example:

Optimization

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



Students: Try this example

```
function f = mysimplefunc(x)

f = 2*x.^2 + 20.*x -22;
```

Note! if we have more than 1 variable, we have to use e.g., the `fminsearch` function

```
clear
clc
close all

x = -20:1:20;
f = mysimplefunc(x);
plot(x, f)
grid

x_min = fminbnd(@mysimplefunc, -20, 20)

y = mysimplefunc(x_min)
```

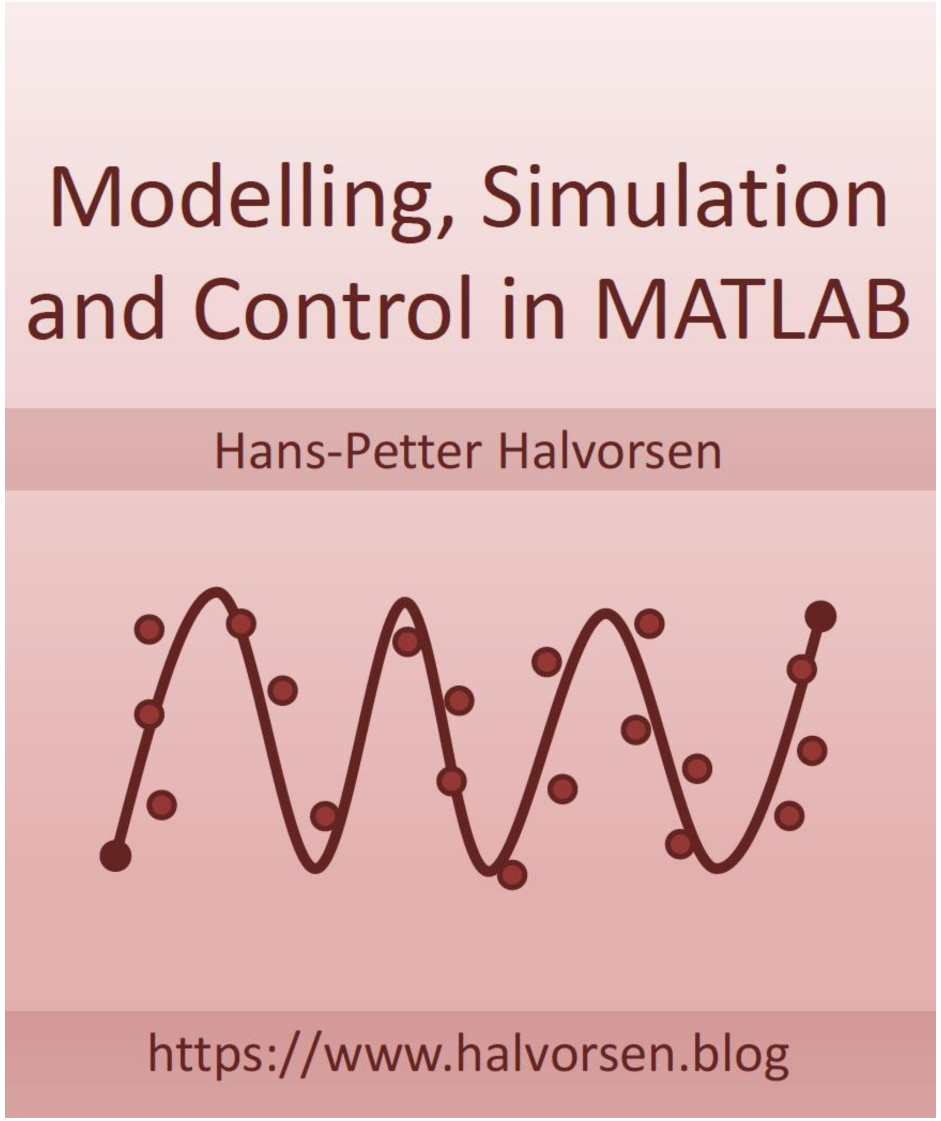
```
x_min =

    -5


y =

   -72
```

We got the same results as previous slide

MATLAB for Students

Whats next?

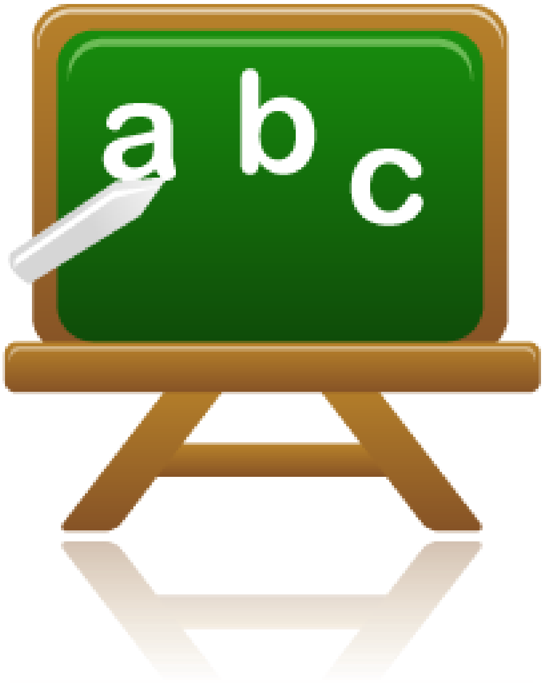
Learning by Doing!



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 6



- Transfer Functions
- State-space models

$$H(s) = \frac{y(s)}{u(s)} = \frac{2}{s^2 + 4s + 3}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Transfer functions

$$H(S) = \frac{x(s)}{u(s)}$$

Differential Equations $\xleftrightarrow{\text{Laplace}}$ **Transfer Functions**

$$\dot{x} = ax + bu \qquad\qquad H(S) = \frac{x(s)}{u(s)}$$

$$\underset{\text{Input}}{u(s)} \longrightarrow \boxed{H(s)} \longrightarrow \underset{\text{Output}}{x(s)}$$

A Transfer function is the ratio between the input and the output of a dynamic system when all the others input variables and initial conditions is set to zero

Example:

$$H(s) = \frac{x(s)}{u(s)} = \frac{3}{0.5s + 1}$$

Numerator

Denumerator

Transfer functions

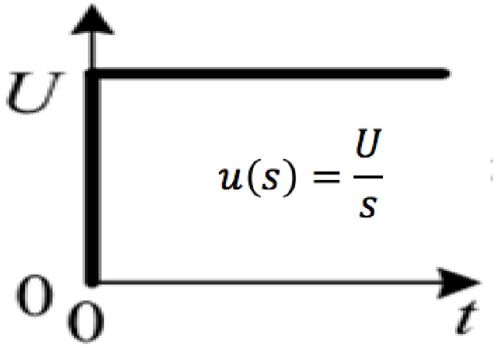
1.order Transfer function:

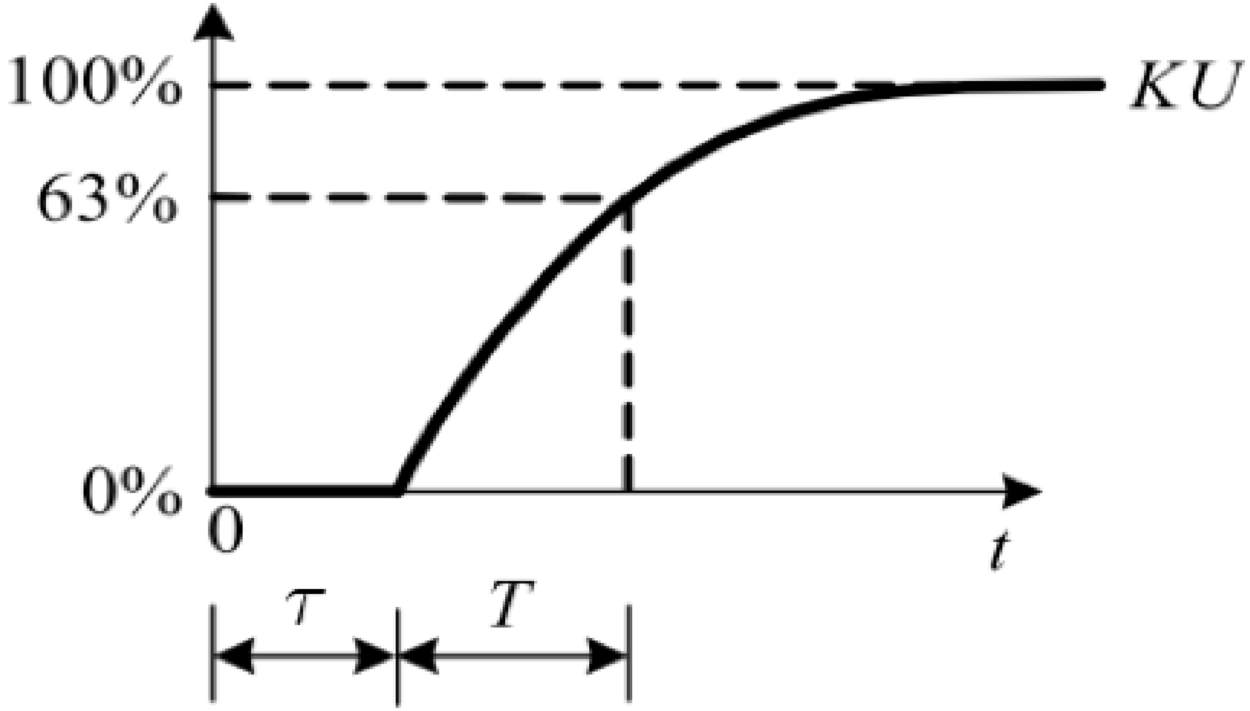
$$\boxed{H(s) = \frac{y(s)}{u(s)} = \frac{K}{Ts+1}}$$

1.order Transfer function with Time Delay:

$$H(s) = \frac{K}{Ts+1}e^{-\tau s}$$

Step Response:





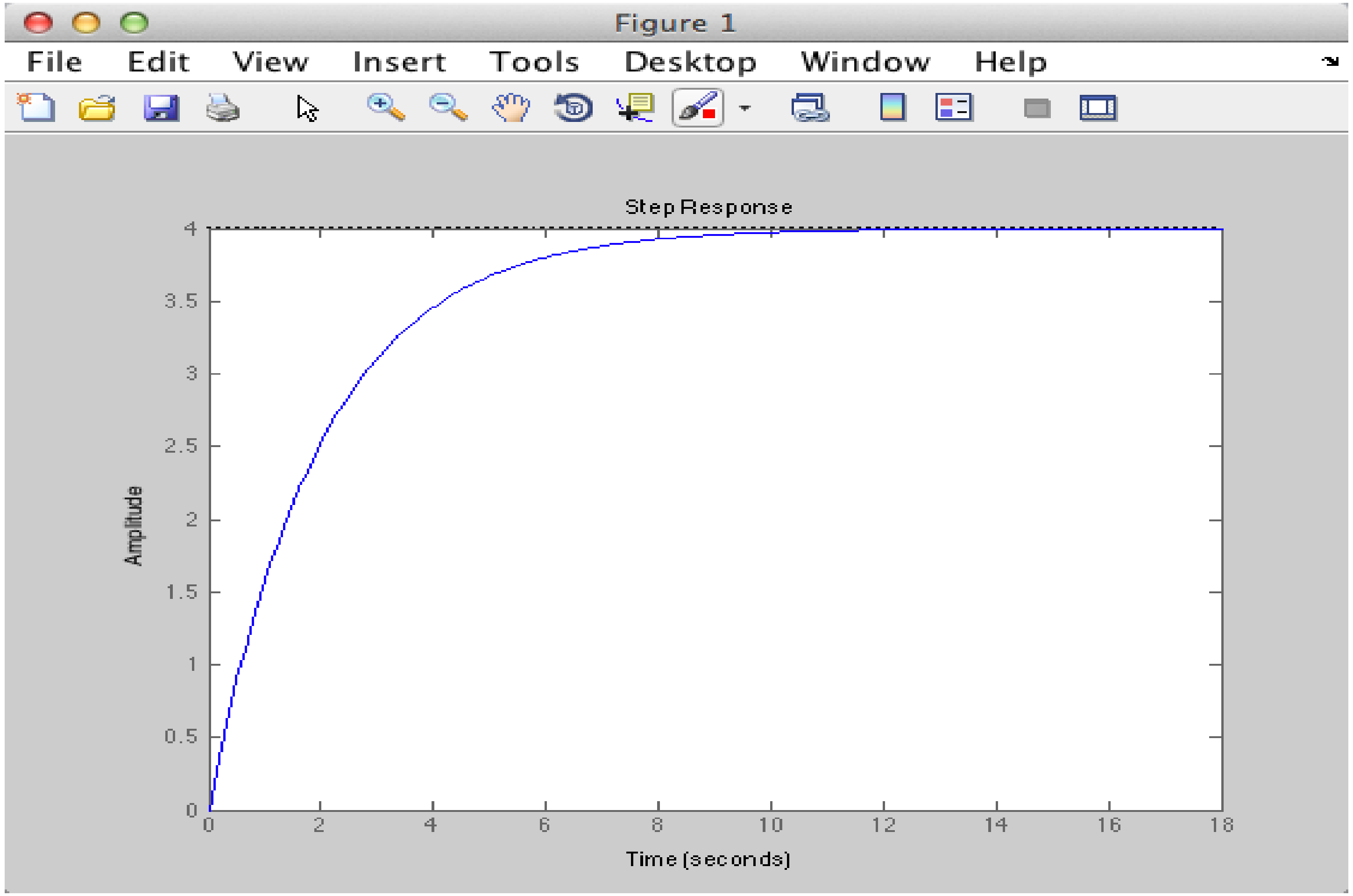
Transfer functions

Example:

$$H(s) = \frac{x(s)}{u(s)} = \frac{4}{2s+1}$$

MATLAB:

```
clear
clc
close all

% Transfer Function
num = [4];
den = [2, 1];
H = tf(num, den)

% Step Response
step(H)
```



Students: Try this example

Transfer functions

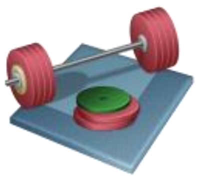
2.order Transfer function:

$$H(s) = \frac{K}{as^2 + bs + c} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 + \omega_0^2} = \frac{K}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\frac{s}{\omega_0} + 1}$$

Example: $$H(s) = \frac{y(s)}{u(s)} = \frac{2}{s^2 + 4s + 3}$$

MATLAB:

```
clear
clc
close all

% Transfer Function
num = [2];
den = [1, 4, 3];
H = tf(num, den)

% Step Response
step(H)
```



Students: Try this example.
Try with different values for *K*, *a*, *b* and *c*.

Value of ζ	Poles p_1 and p_2	Type of step response $y(t)$
$\zeta > 1$	Real and distinct	Overdamped
$\zeta = 1$	Real and multiple	Critically damped
$0 < \zeta < 1$	Complex conj.	Underdamped
$\zeta = 0$	Imaginary	Undamped
$\zeta < 0$	Pos. real part	Unstable

State-space models

A set with linear differential equations:

$$\dot{x}_1 = a_{11}x_1 + a_{21}x_2 + \cdots + a_{n1}x_n + b_{11}u_1 + b_{21}u_2 + \cdots + b_{n1}u_n$$

$$\vdots$$

$$\dot{x}_n = a_{1m}x_1 + a_{2m}x_2 + \cdots + a_{nm}x_n + b_{1m}u_1 + b_{2m}u_2 + \cdots + b_{n1}u_n$$

$$\vdots$$

Can be structured like this:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} b_{11} & \cdots & b_{n1} \\ \vdots & \ddots & \vdots \\ b_{1m} & \cdots & b_{nm} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}}_{u}$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} c_{11} & \cdots & c_{n1} \\ \vdots & \ddots & \vdots \\ c_{1m} & \cdots & c_{nm} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} d_{11} & \cdots & d_{n1} \\ \vdots & \ddots & \vdots \\ d_{1m} & \cdots & d_{nm} \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}}_{u}$$

Which can be stated on the following compact form:

$$\boxed{\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}}$$

State-space models

Example:

$$\dot{x}_1 = x_1 + 2x_2$$

$$\dot{x}_2 = 3x_1 + 4x_2 + u$$

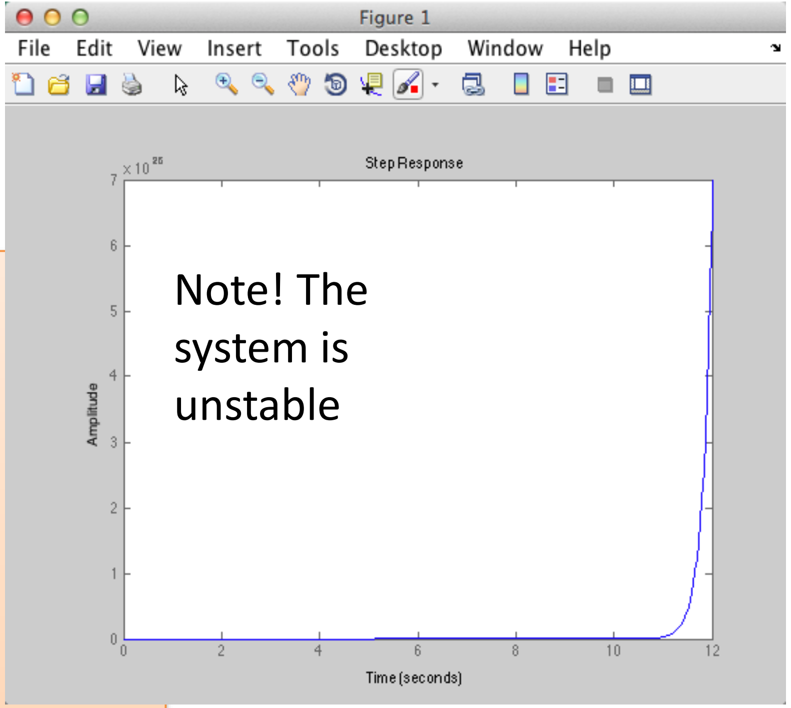
$$y = x_1$$

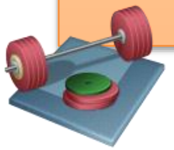
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

MATLAB:

```
clear
clc
close all

% State-space model
A = [1, 2; 3, 4];
B = [0; 1];

C = [1, 0];
D = [0];

ssmodel = ss(A, B, C, D)

% Step Response
step(ssmodel)

% Transfer function
H = tf(ssmodel)
```

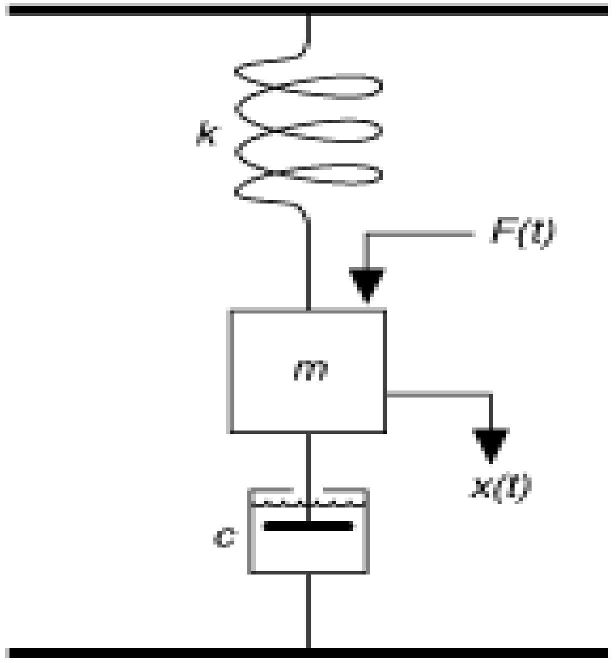




Students: Try this example

State-space models

Mass-Spring-Damper System



Example:

$$\boxed{\ddot{x} = -\frac{k}{m}x - \frac{c}{m}\dot{x} + \frac{1}{m}F}$$

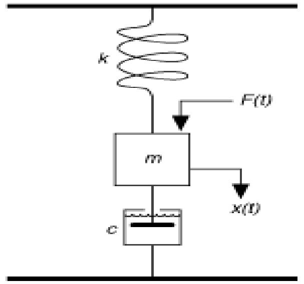
x – position, $\dot{x}$ – speed/velocity, $\ddot{x}$ – acceleration

c - damping constant, m - mass, k - spring constant, F – force

Students: Find the State-space model and find the step response in MATLAB.
Try with different values for k, m, c and F.
Discuss the results

State-space models

Mass-Spring-Damper System



We set:

$$x_1 = x$$
$$x_2 = \dot{x} = \dot{x}_1$$

This gives:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \ddot{x}$$

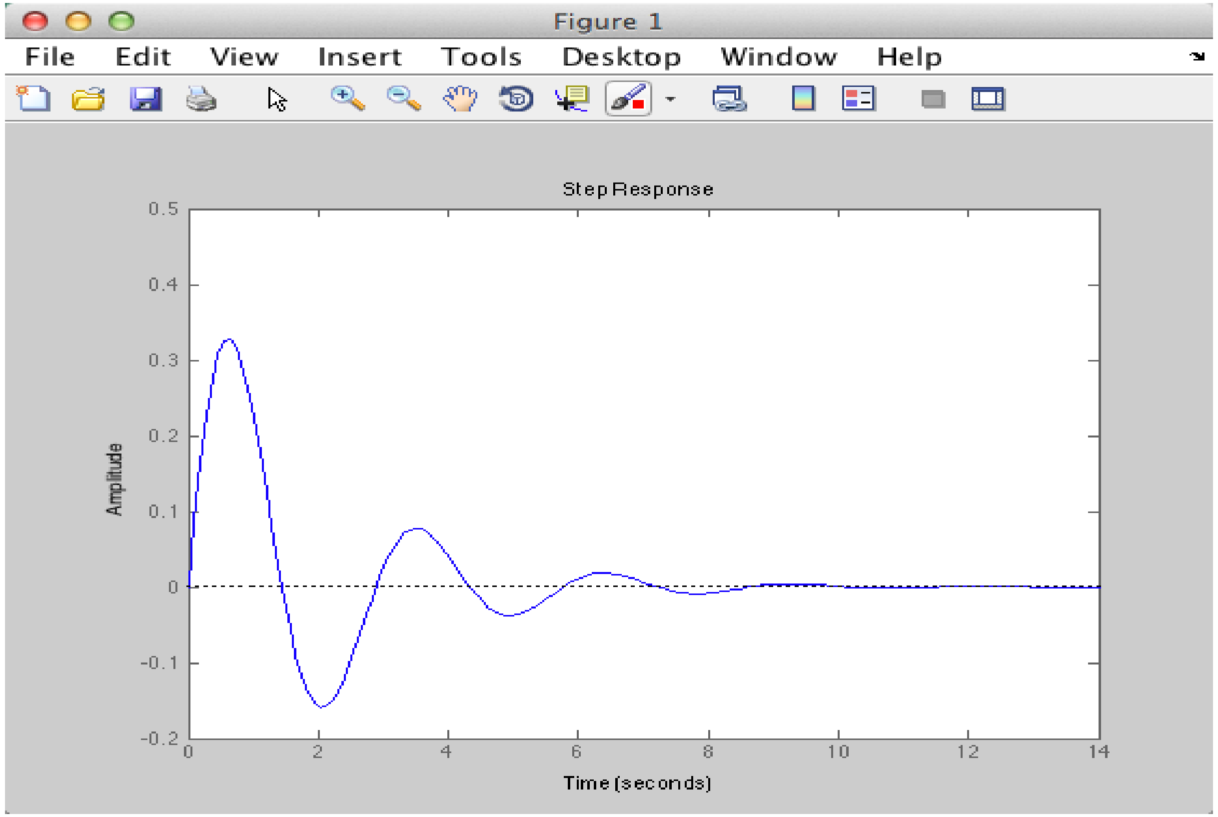
This gives:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}F$$

Finally: Note! we have set $F = u$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

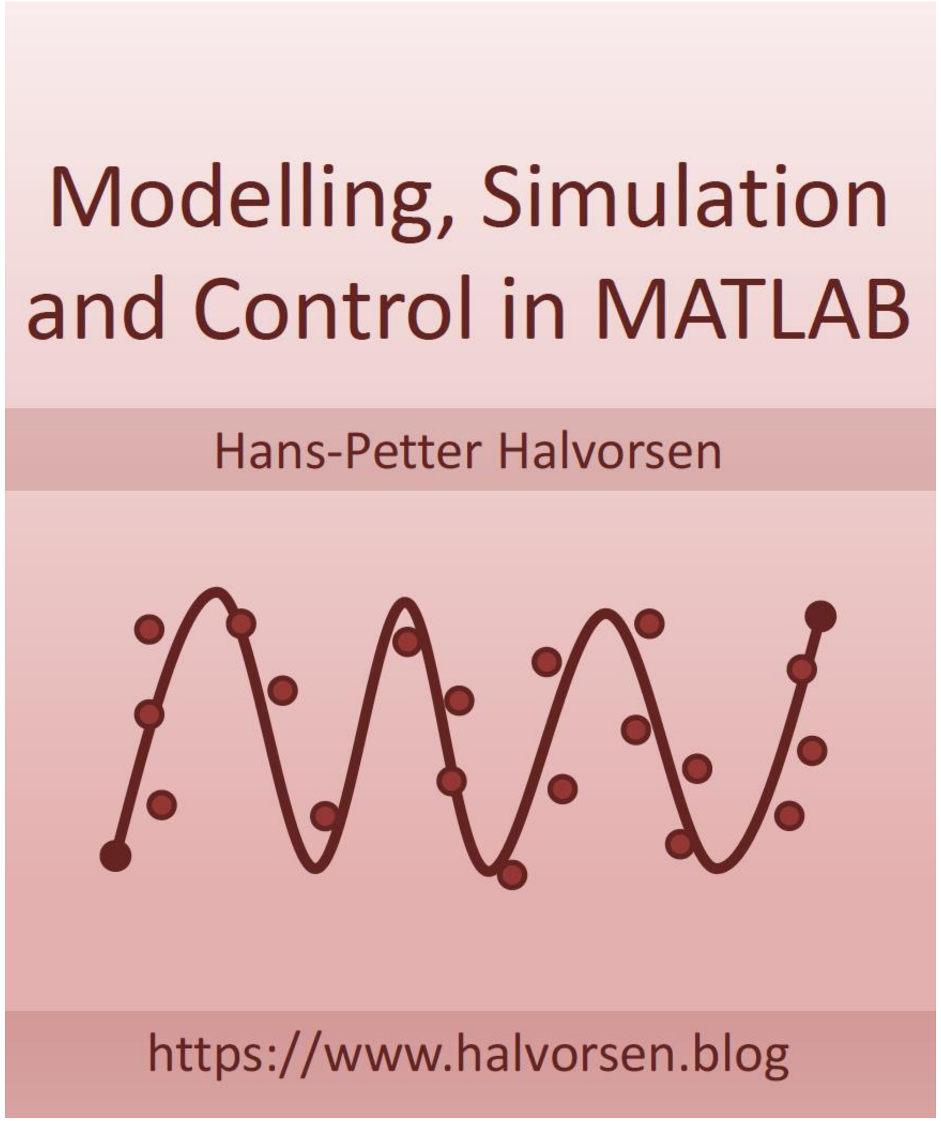
```
k = 5;
c = 1;
m = 1;

A = [0 1; -k/m -c/m];
B = [0; 1/m];
C = [0 1];
D = [0];
sys = ss(A, B, C, D)

step(sys)
```



MATLAB for Students

Whats next?

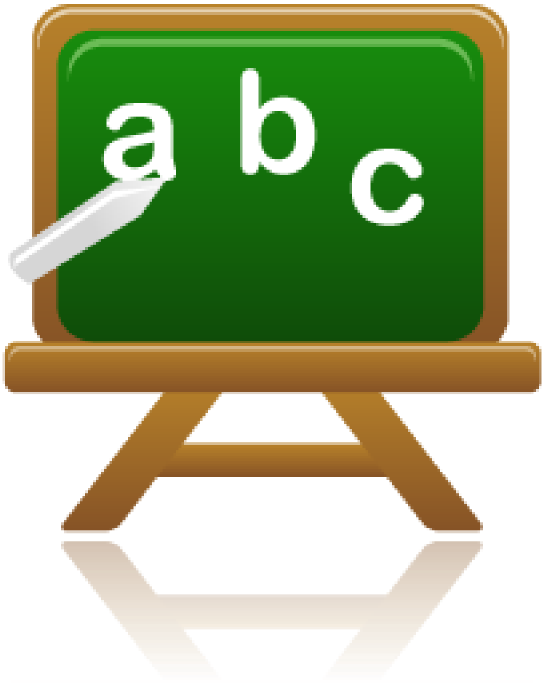
Learning by Doing!



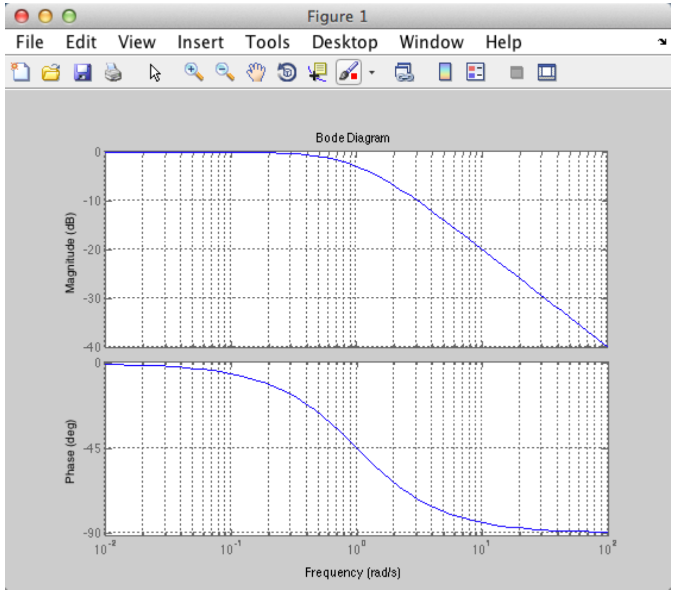
Self-paced Tutorials with lots of Exercises and Video resources

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

Lesson 7



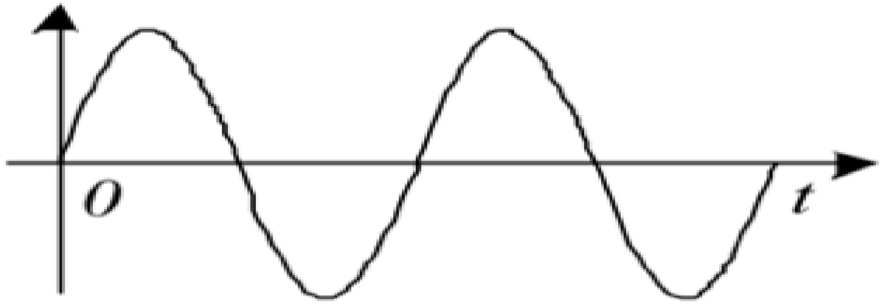
- Frequency Response



Frequency Response

Example:

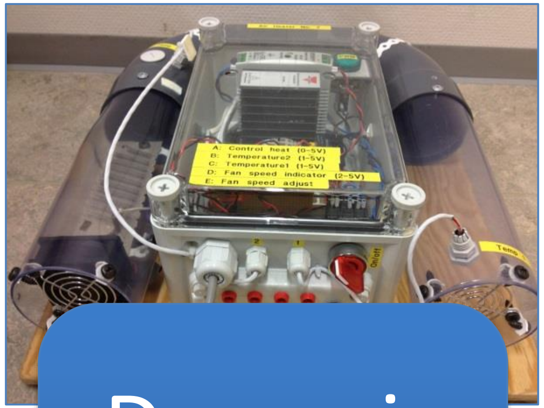
Imput Signal



$$u(t) = U\, sin\omega t$$

Amplitude Frequency

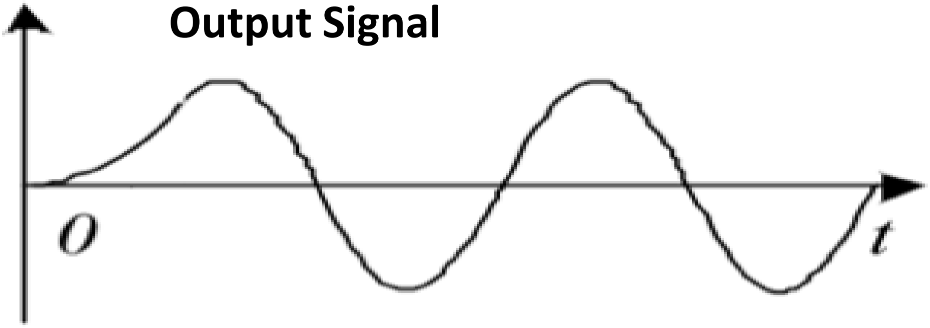
Air Heater



Dynamic System

$$H(s)$$

Output Signal

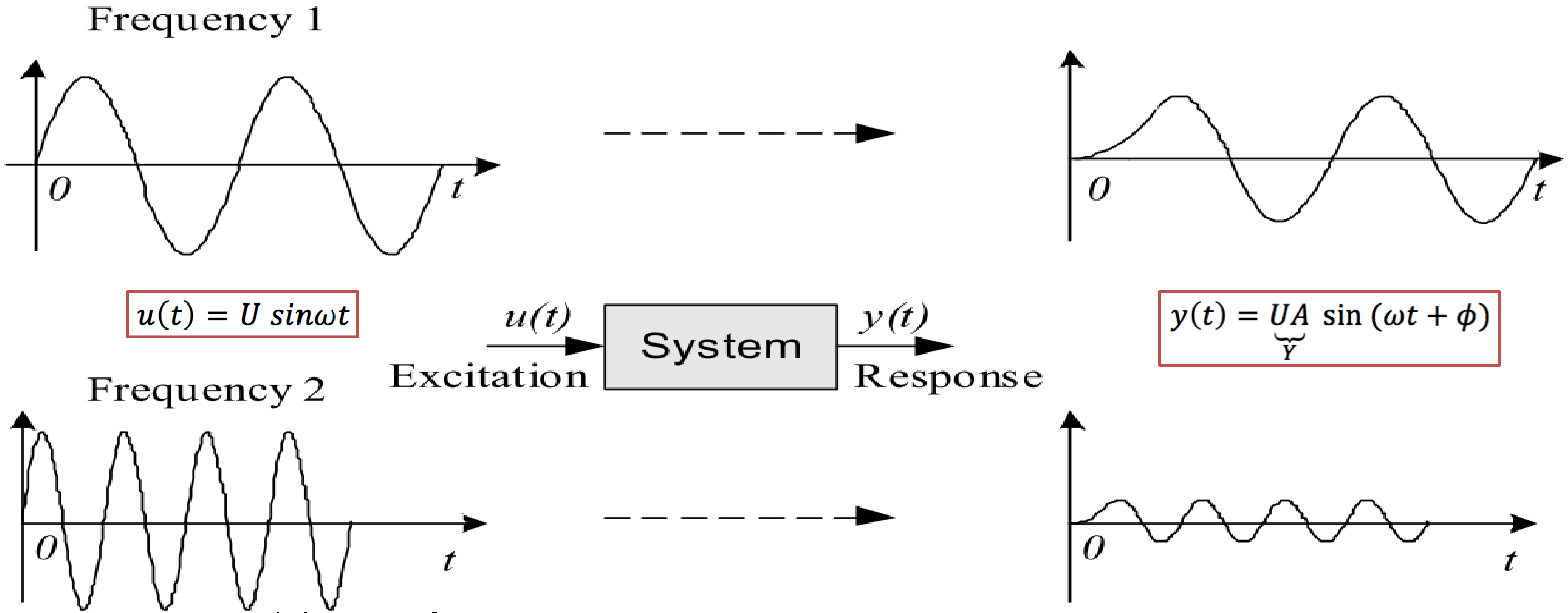


$$y(t) = \underbrace{UA}_{Y}\ \sin(\omega t + \phi)$$

Gain Phase Lag

The frequency response of a system expresses how a sinusoidal signal of a given frequency on the system input is transferred through the system.

Frequency Response - Definition



and the same for Frequency 3, 4, 5, 6, etc.

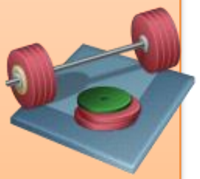
- The frequency response of a system is defined as the <u>**steady-state**</u> **response** of the system to a **sinusoidal** input signal.
- When the system is in steady-state, it differs from the input signal only in **amplitude/gain** (A) ("forsterkning") and **phase lag** (φ) ("faseforskyvning").

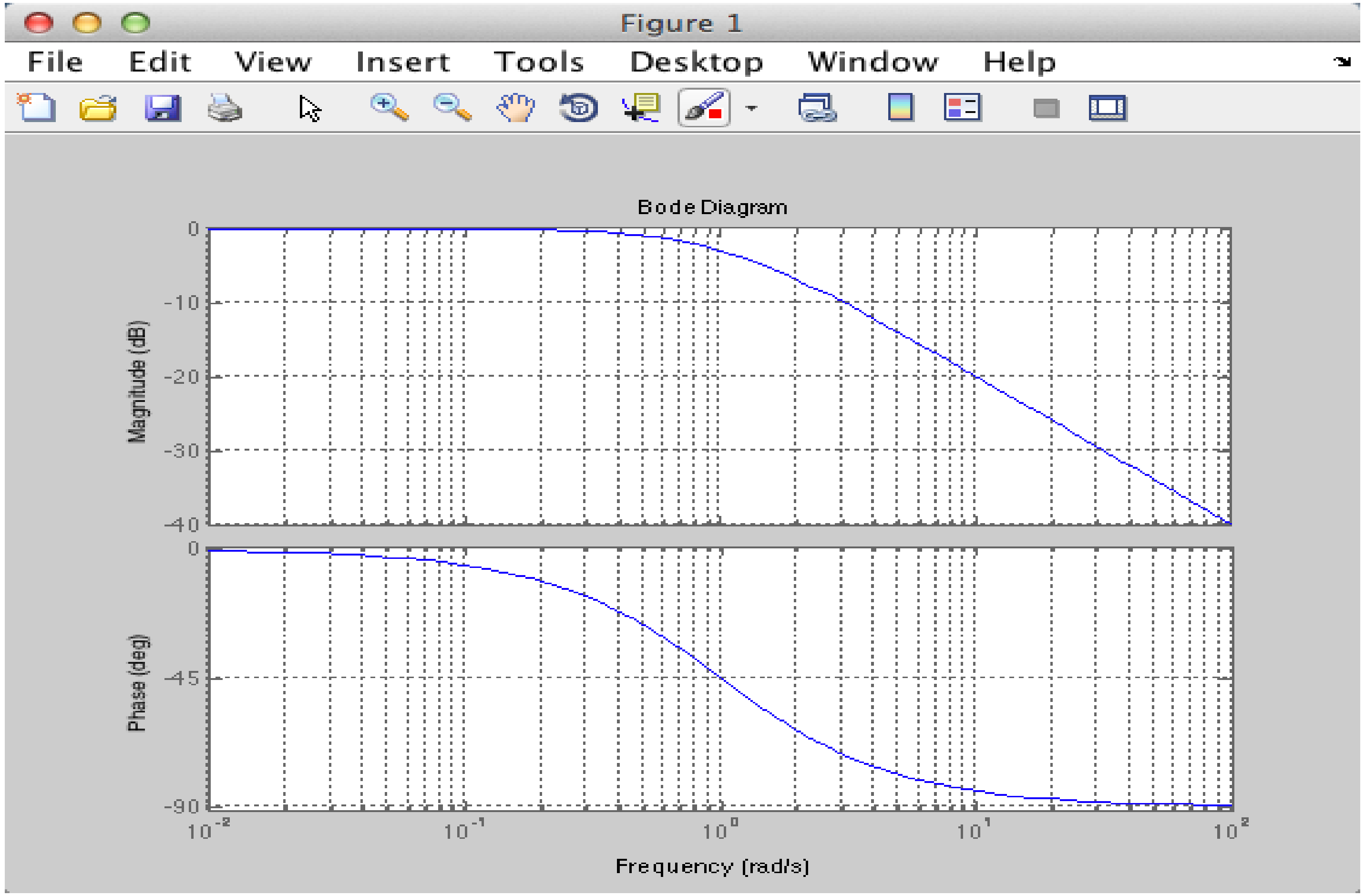
Frequency Response

Example:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s+1}$$

```
clear
clc
close all

% Define Transfer function
num=[1];
den=[1, 1];
H = tf(num, den)

% Frequency Response
bode(H);
grid on
```





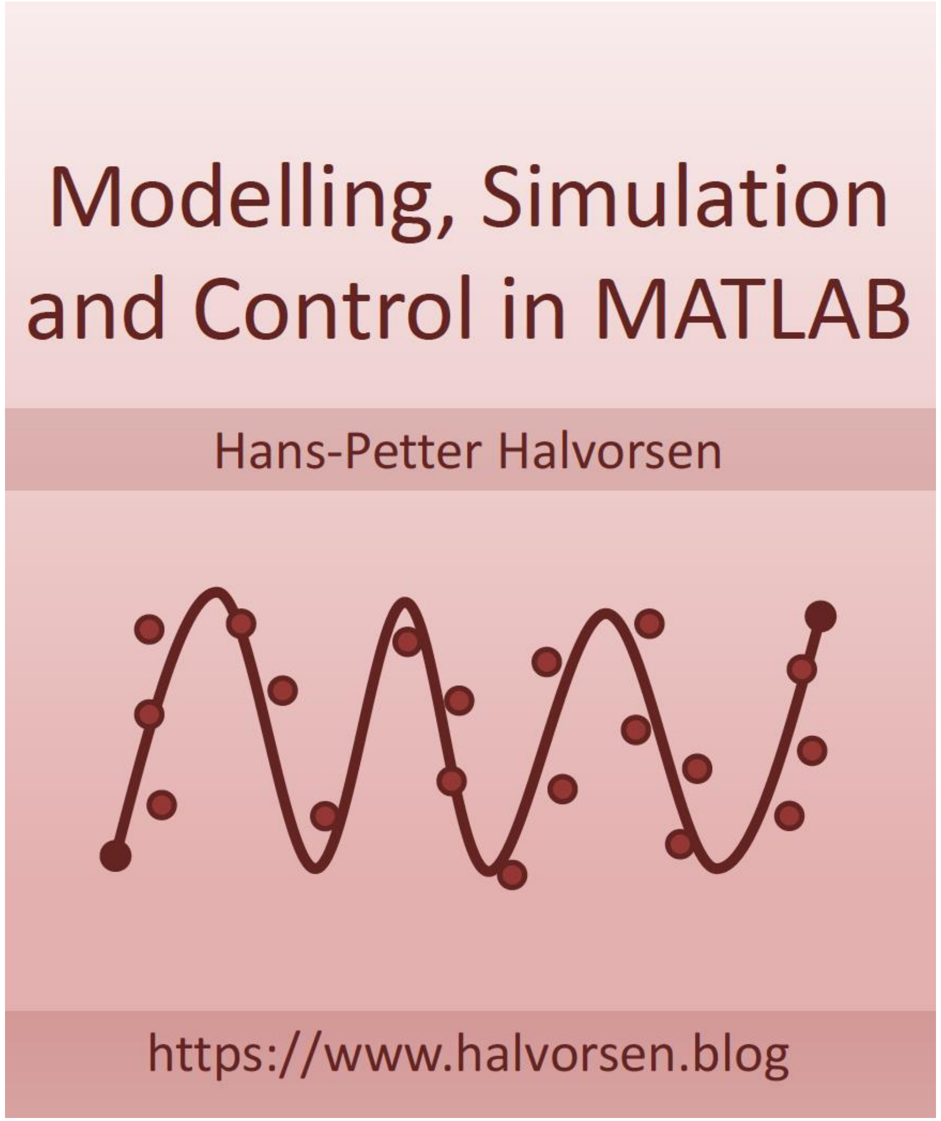
Students: Try this Example

The frequency response is an important tool for analysis and design of signal filters and for analysis and design of control systems.

MATLAB for Students

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!!!

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog