https://www.halvorsen.blog

# Simulink

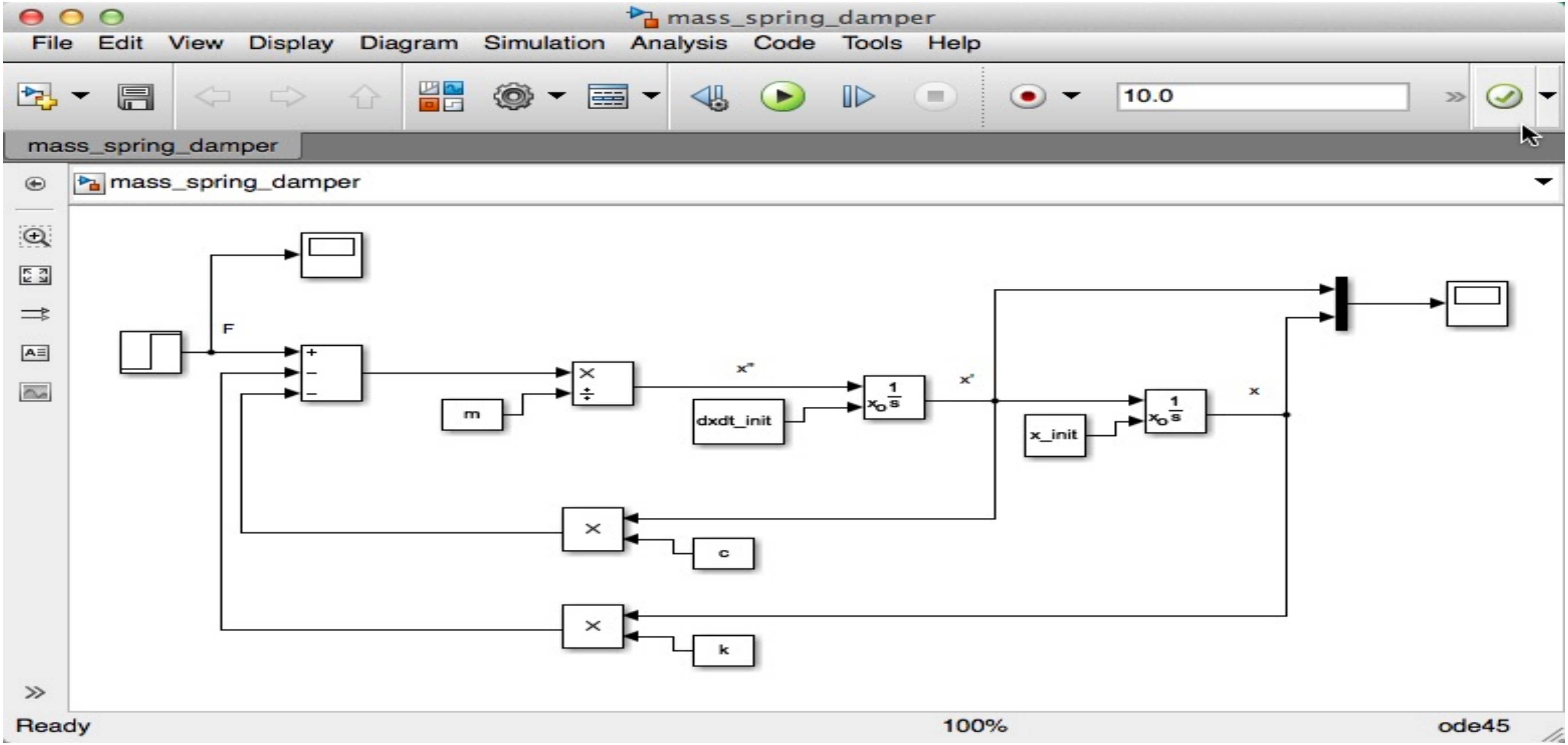
Graphical Programming and Simulation with MATLAB

Hans-Petter Halvorsen

# What is Simulink?

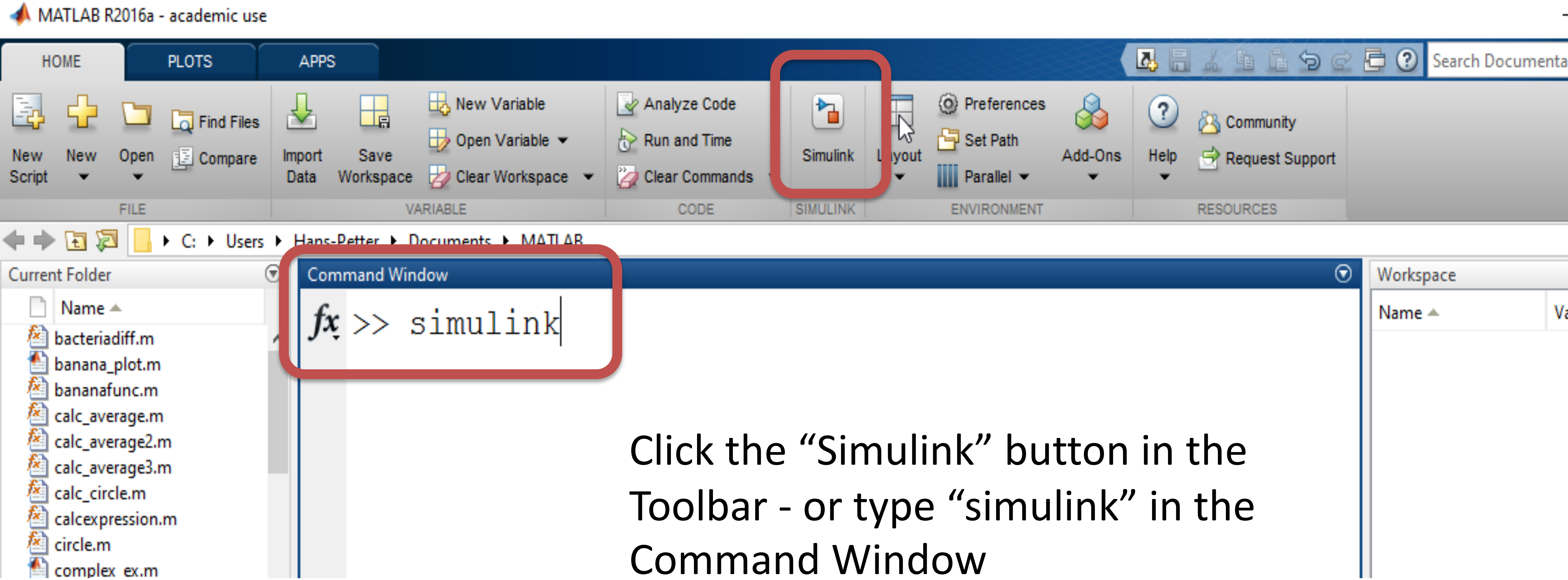
- Simulink is an “add-on” to MATLAB.
- You need to have MATLAB in order to use Simulink
- Simulink is used for Simulation of dynamic models
- In Simulink we create a Graphical Block Diagram for the system (based on the differential equations(s))

# Simulink Model



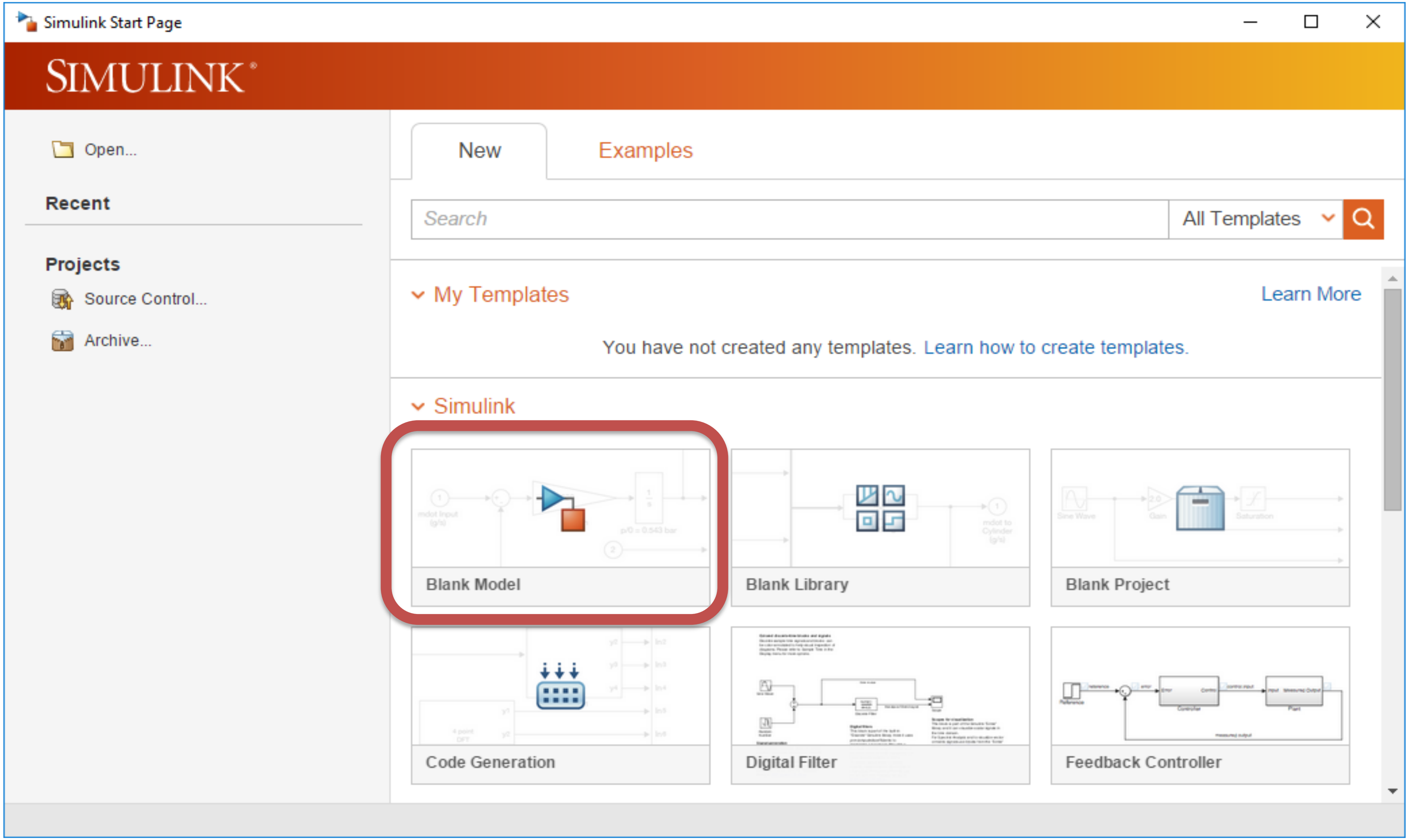

In Simulink we create and configure graphical blocks and wire them together (based on the differential equations)

# Start Simulink from MATLAB

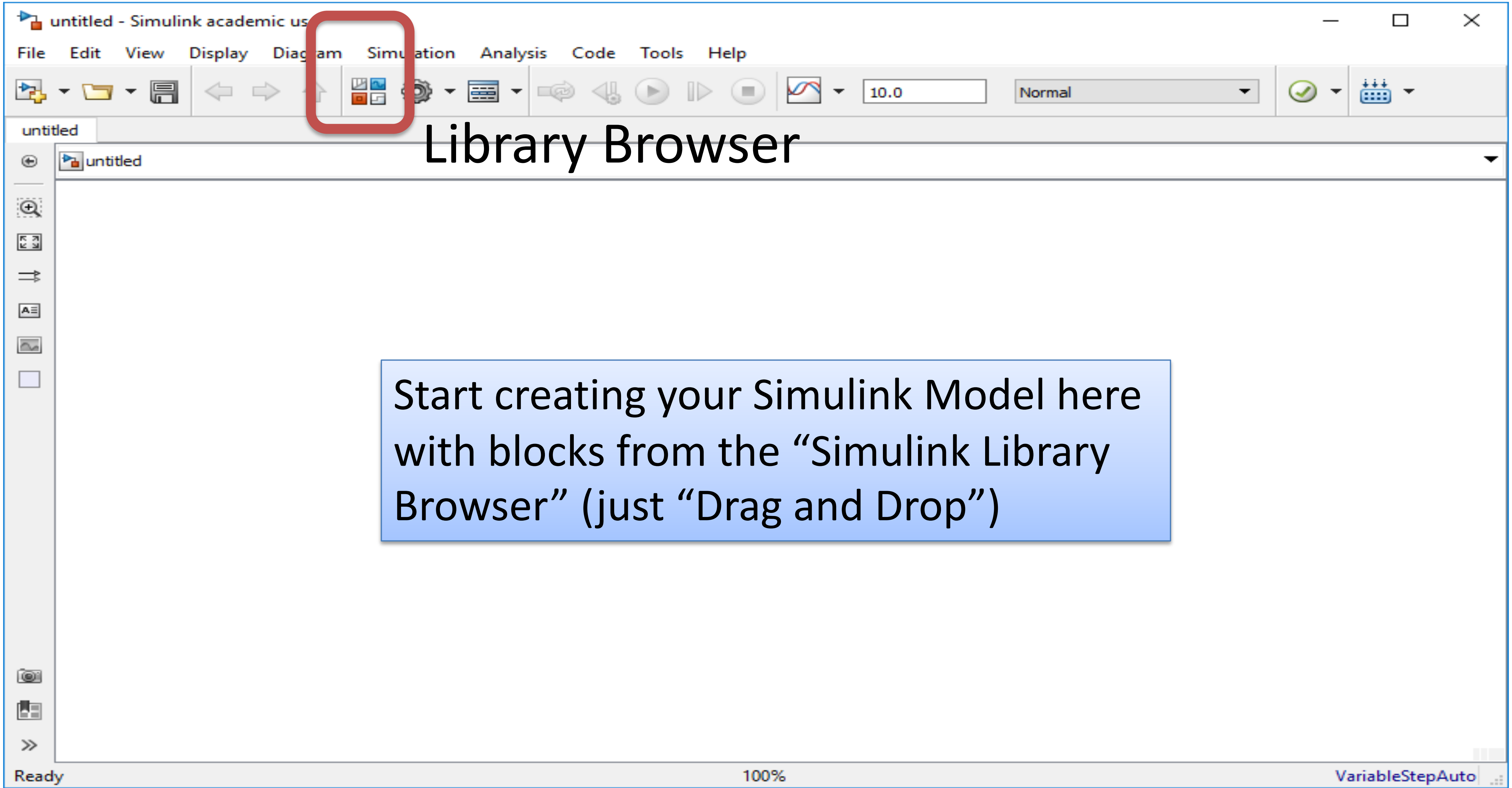



Click the “Simulink” button in the Toolbar - or type “simulink” in the Command Window

# Simulink Start Page



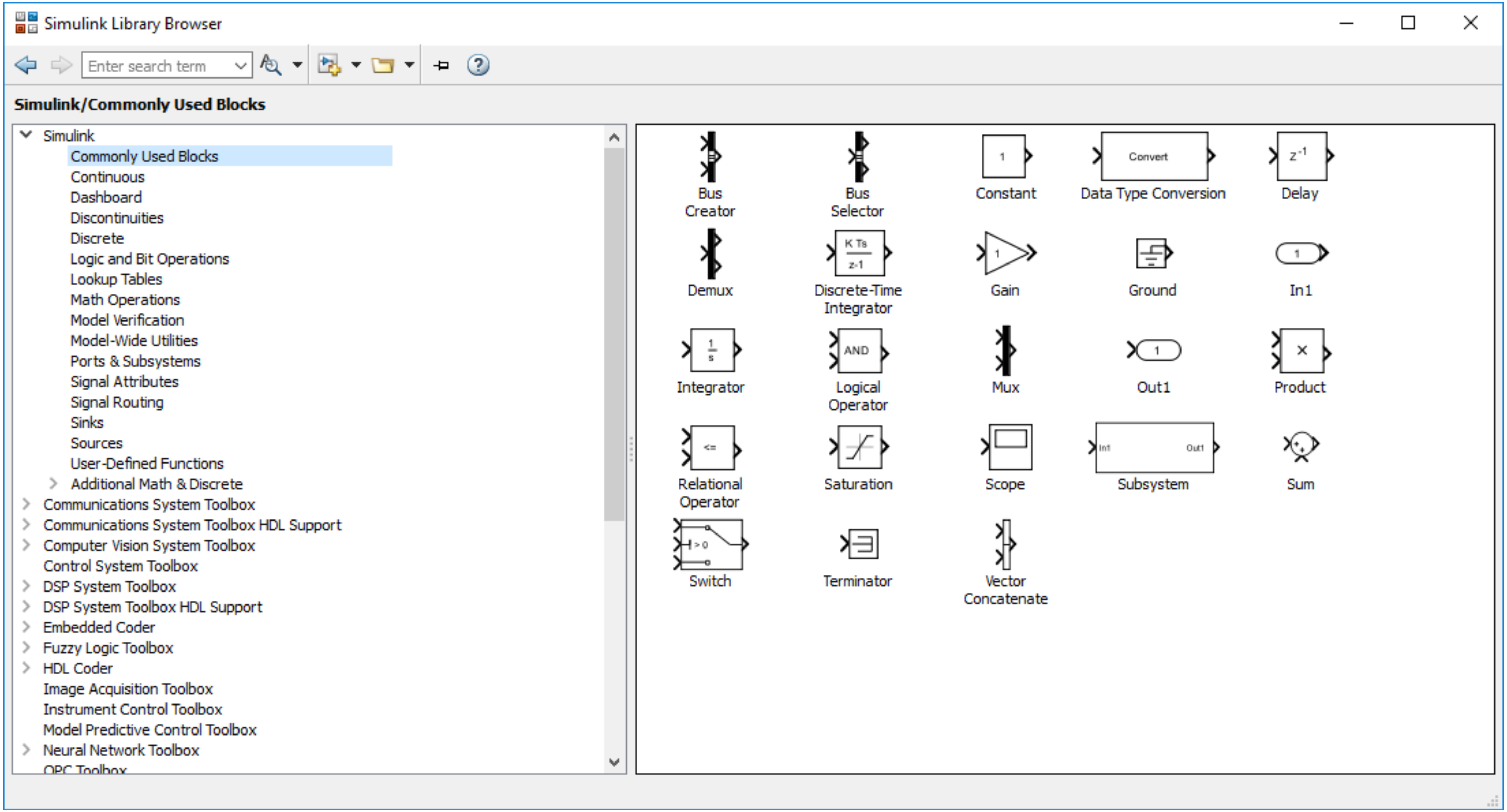
# Simulink Model Editor



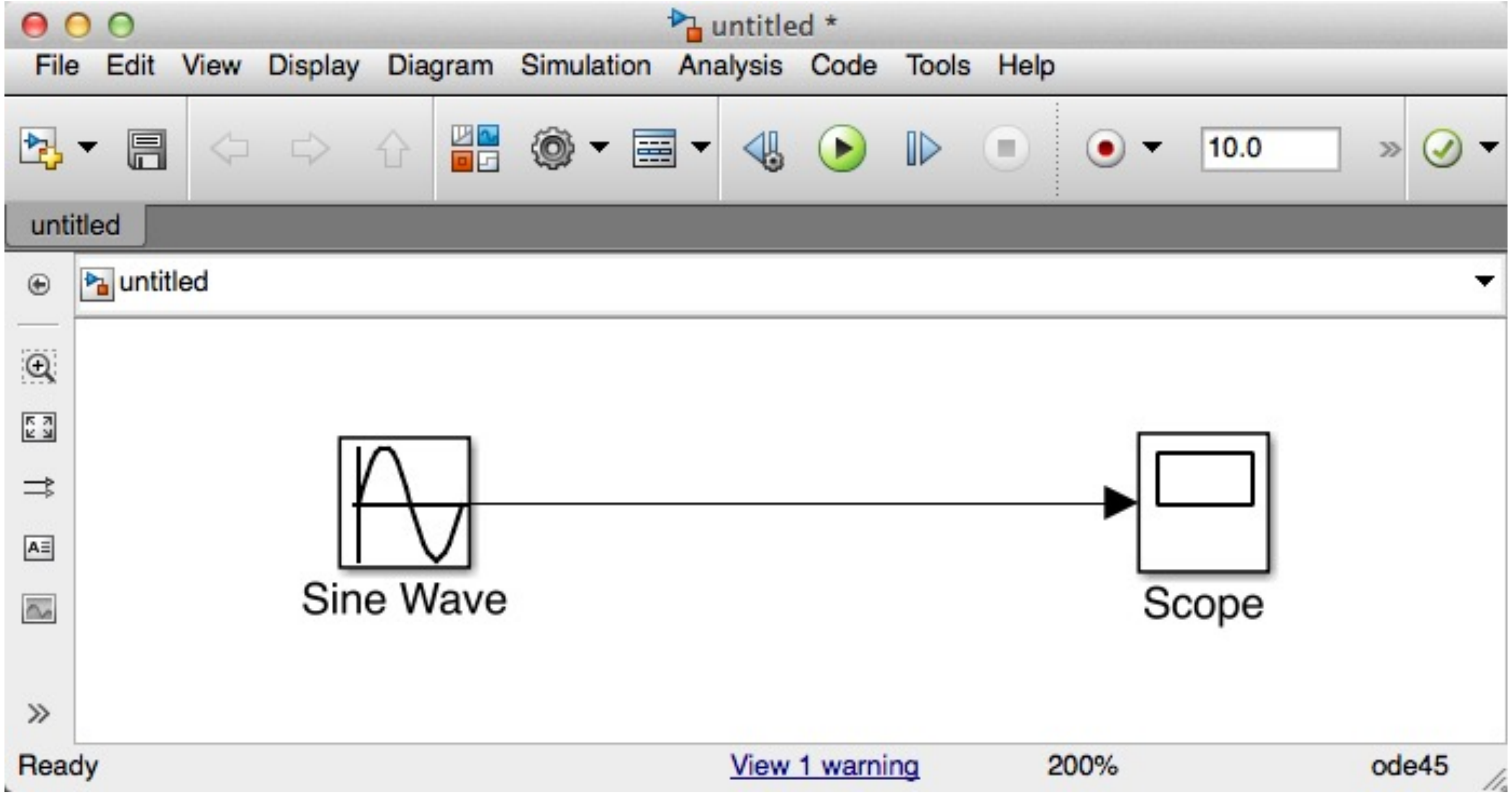

Library Browser

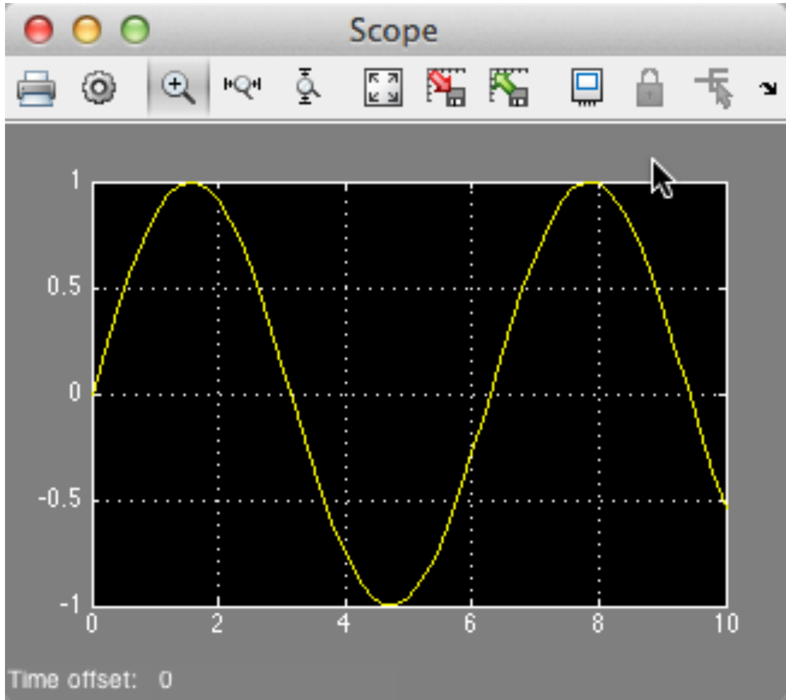
Start creating your Simulink Model here with blocks from the “Simulink Library Browser” (just “Drag and Drop”)

# Simulink Library Browser

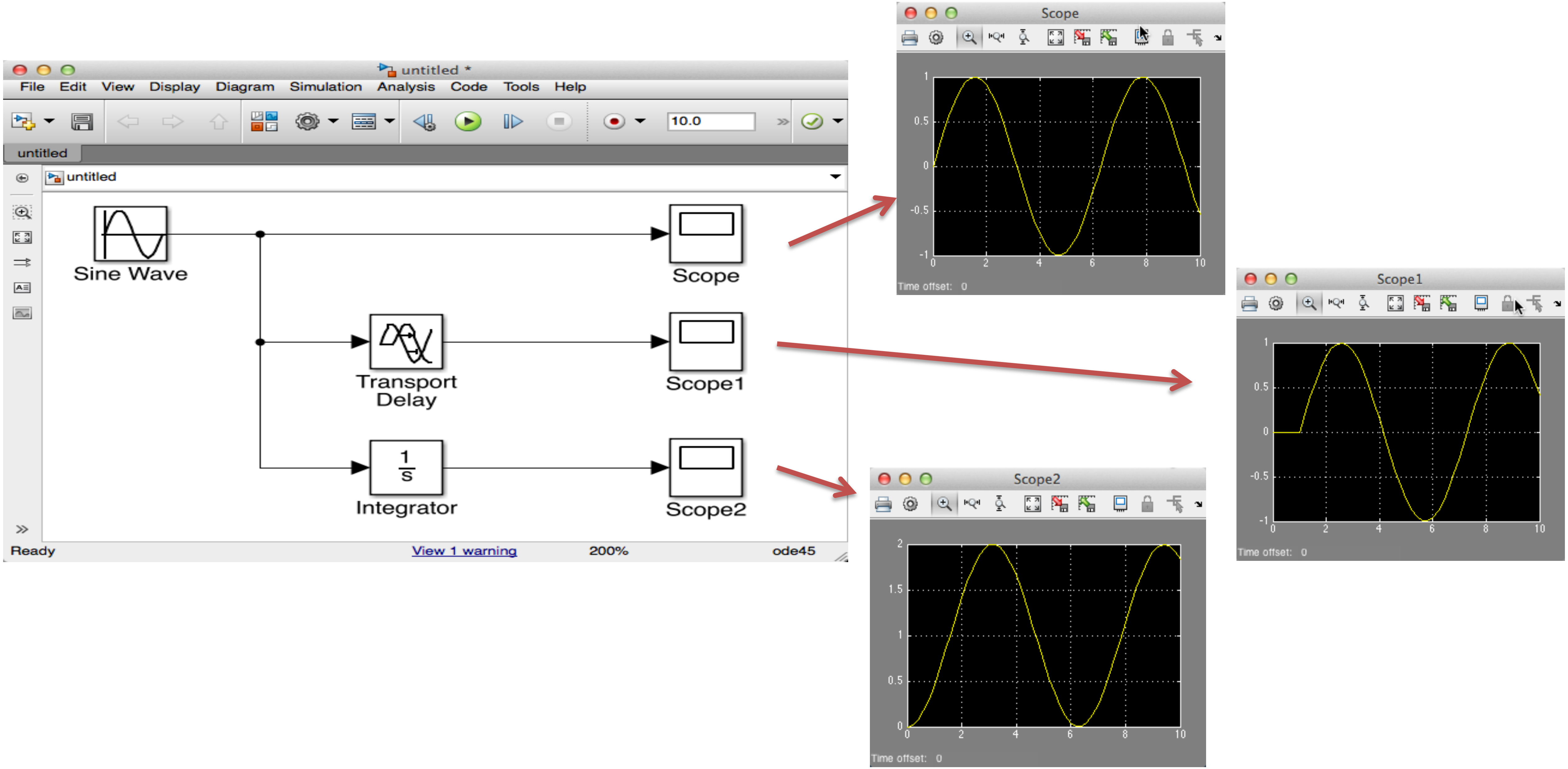


# Simulink Example



# Simulink Example II



Example

# My First Simulink Model

Model:

$$\dot{x} = ax$$

Where

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

$T$ is the Time constant

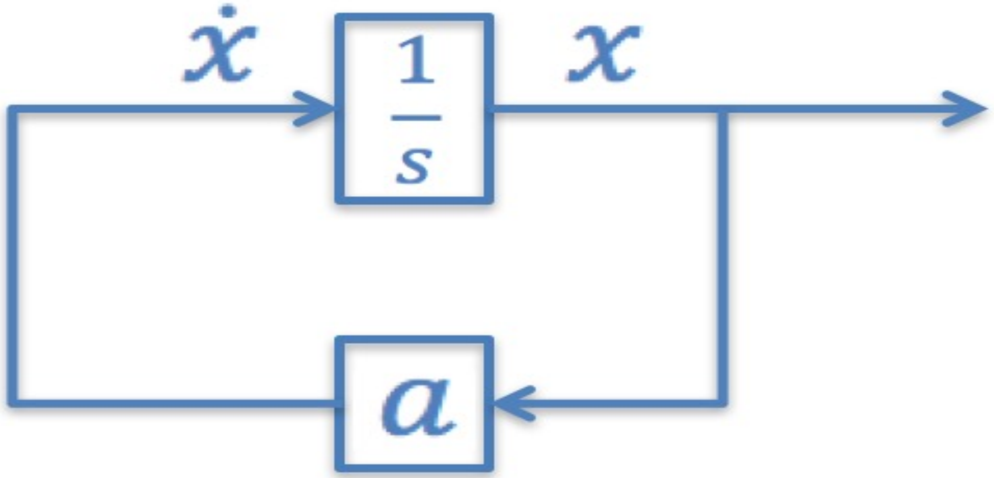
We will use the following:

$$T = 5$$

$$x(0) = 1$$

$$0 \le t \le 25$$

We start by drawing a simple Block Diagram for the model like this (“Pen & paper”):



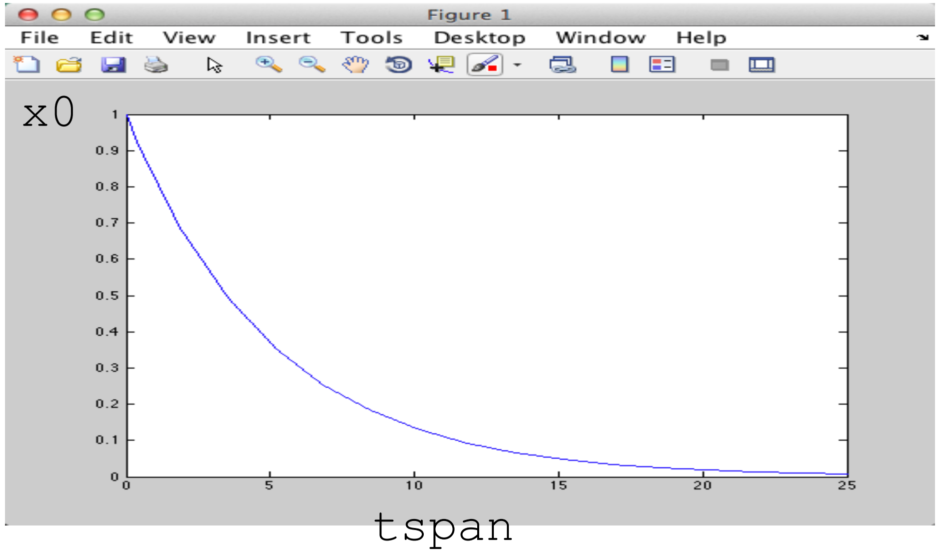

We will create and simulate this block diagram with Simulink

# Using ODE Solvers in MATLAB

$$\dot{x} = ax$$

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

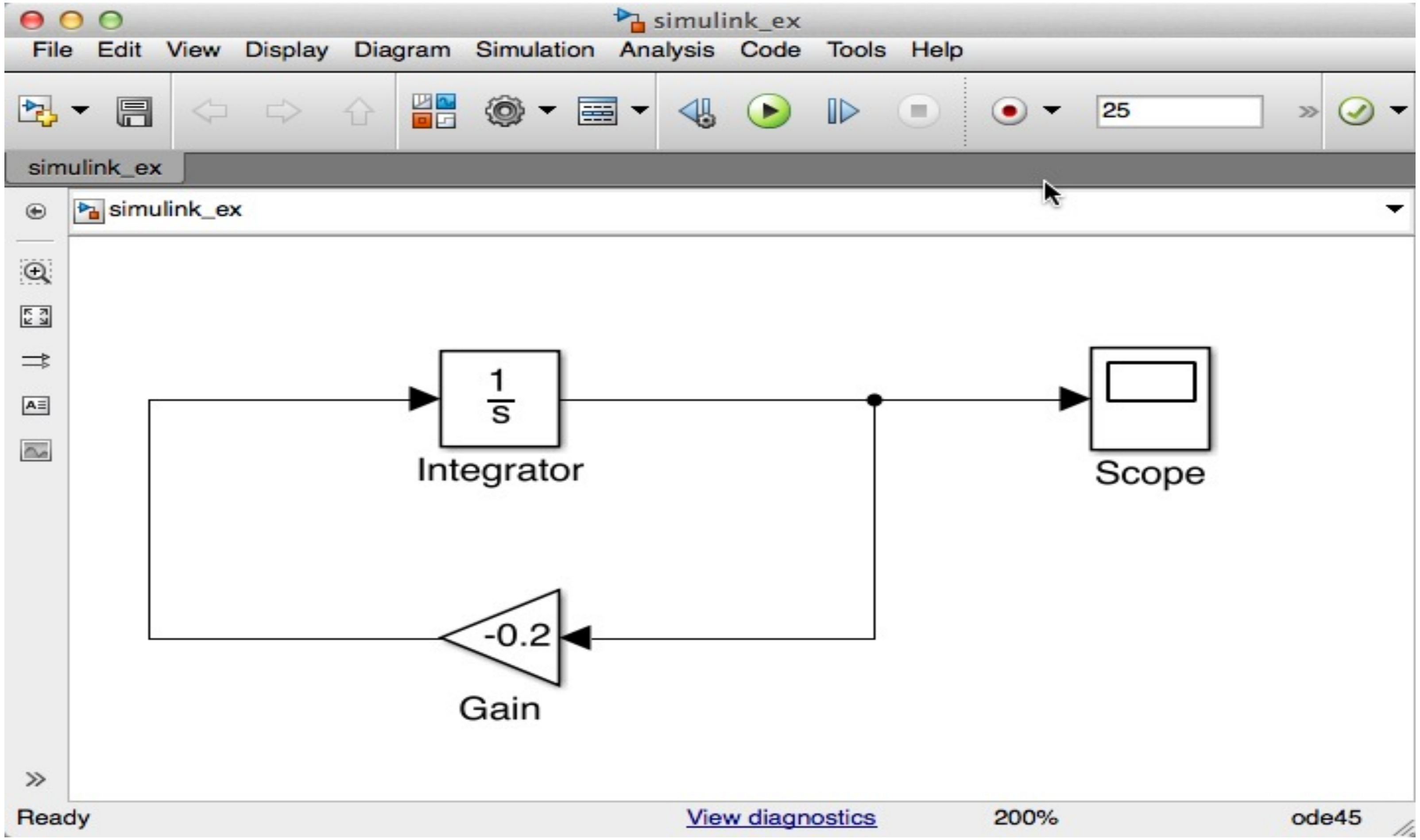
```
function dx = mydiff(t,x)
T = 5;
a = -1/T;
dx = a*x;
```

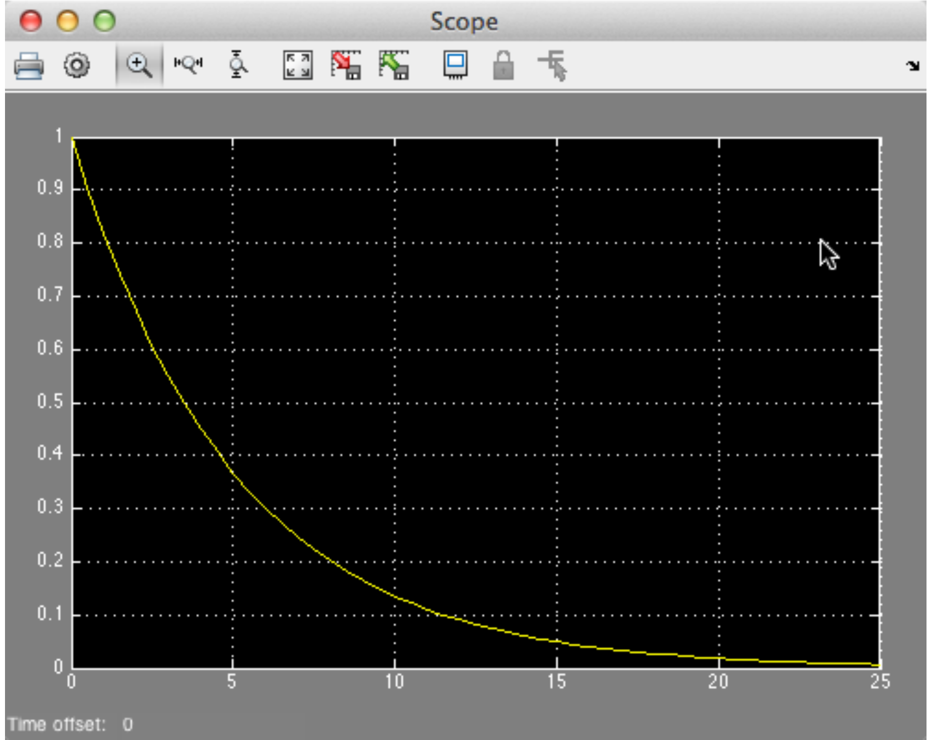



**Step 2:** Use one of the built-in ODE solver (ode23, ode45, ...) in a Script.

```
clear
clc

tspan = [0 25];
x0 = 1;

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

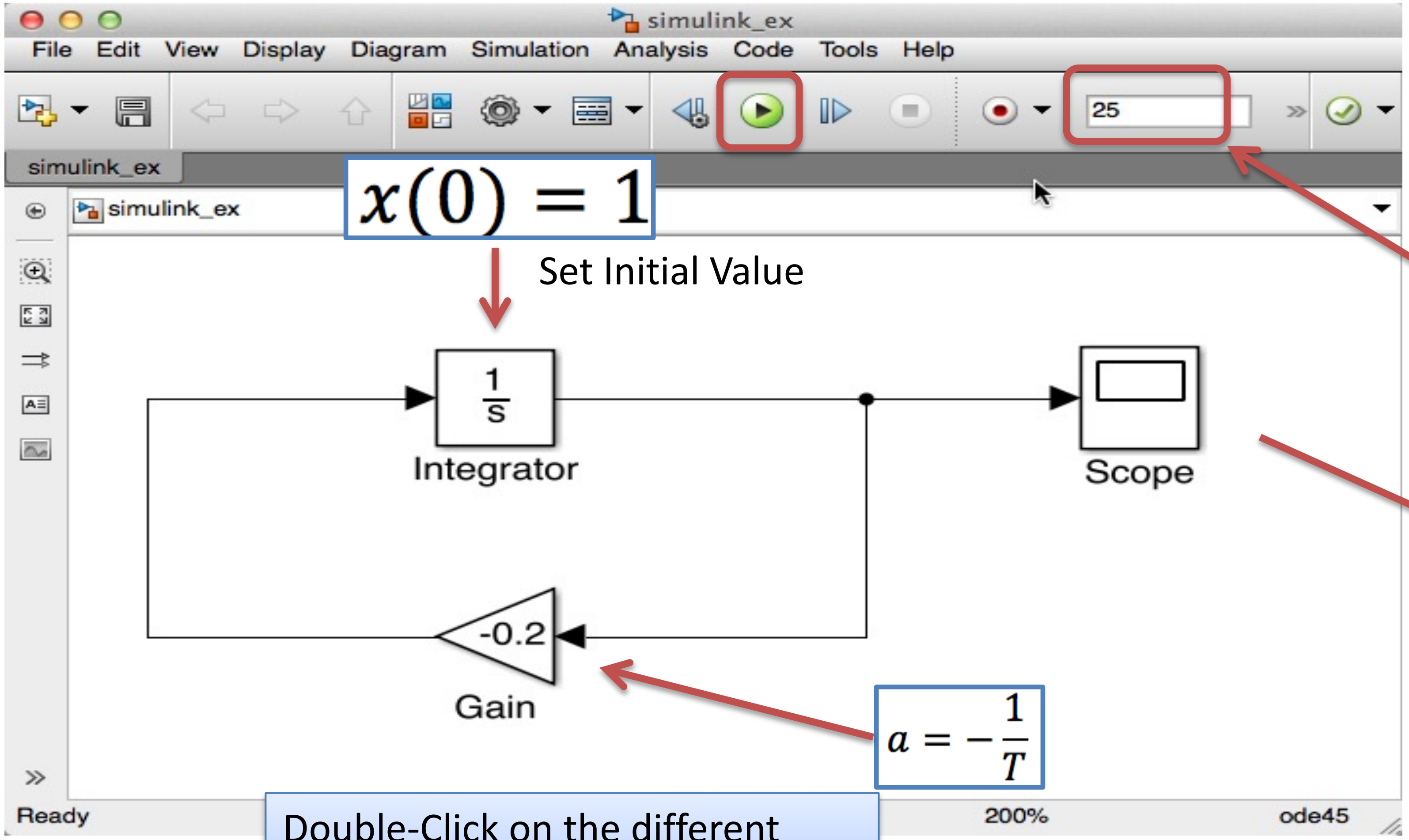
# My First Simulink Model



Solution

# My First Simulink Model




$x(0) = 1$

Set Initial Value

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

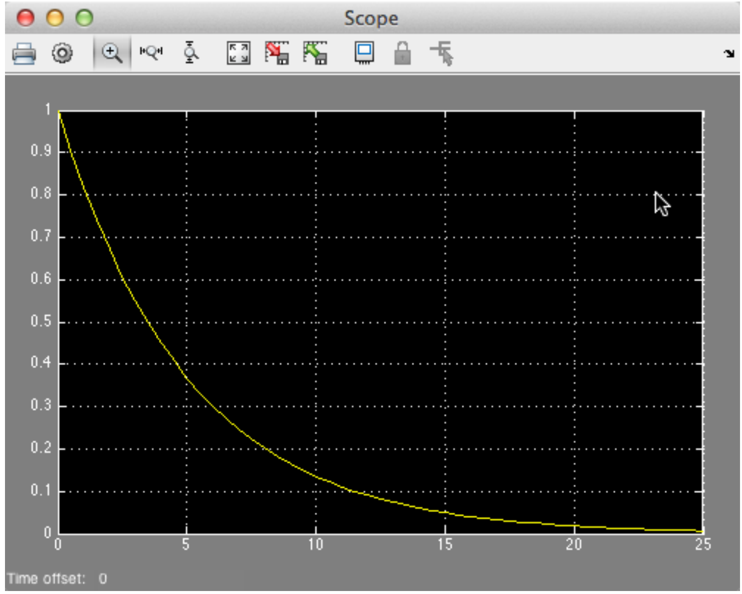
Double-Click on the different Blocks to enter values

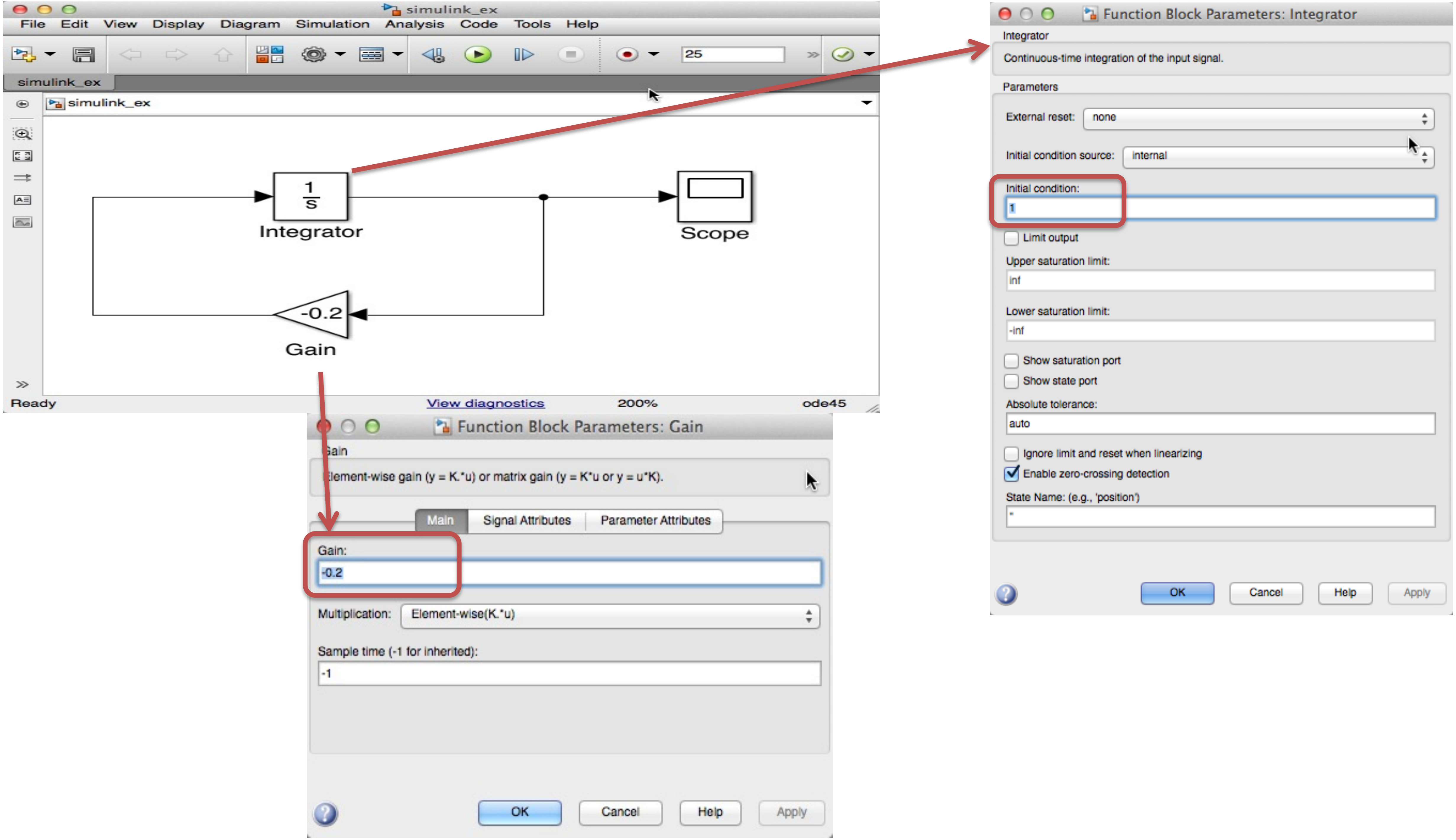
Start the Simulation by clicking this icon

Simulation Time

$0 \leq t \leq 25$

See the Simulation Results




simulink_ex
File Edit View Display Diagram Simulation Analysis Code Tools Help
25
simulink_ex
simulink_ex
1
s
Integrator
Scope
-0.2
Gain
Ready
View diagnostics
200%
ode45
Function Block Parameters: Gain
Gain
Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Main
Signal Attributes
Parameter Attributes
Gain:
-0.2
Multiplication:
Element-wise(K.*u)
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply
Function Block Parameters: Integrator
Integrator
Continuous-time integration of the input signal.
Parameters
External reset:
none
Initial condition source:
internal
Initial condition:
1
Limit output
Upper saturation limit:
inf
Lower saturation limit:
-inf
Show saturation port
Show state port
Absolute tolerance:
auto
Ignore limit and reset when linearizing
Enable zero-crossing detection
State Name: (e.g., 'position')
''
OK
Cancel
Help
Apply

# Bacteria Population

Here we will simulate a simple model of a bacteria population in a jar.

The model is as follows:

$$\textit{birth rate} = bx$$

$$\textit{death rate} = px^2$$

Then the total rate of change of bacteria population is:

$$\dot{x} = bx - px^2$$

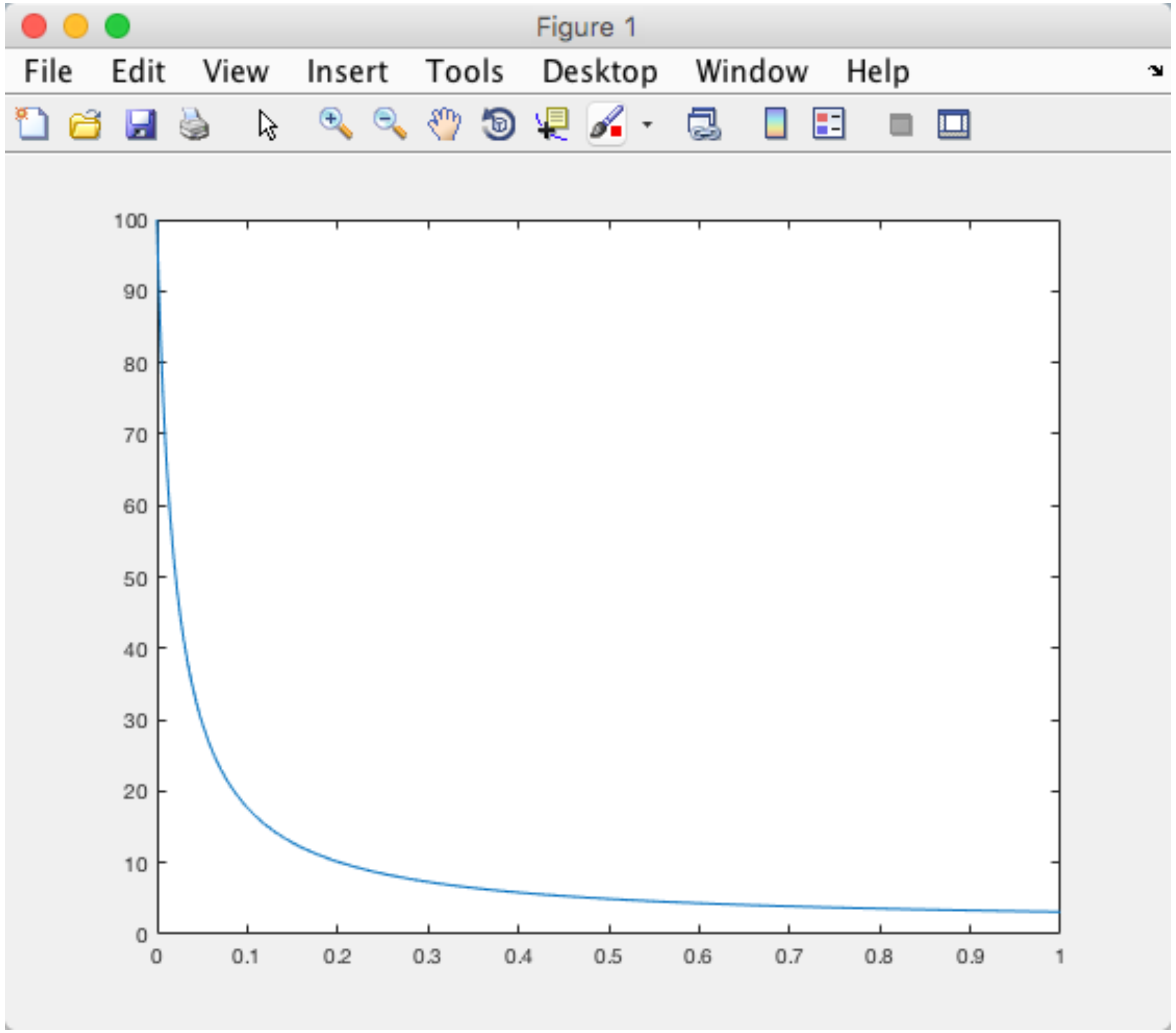
Set ***b=1***/hour and ***p=0.5*** bacteria-hour

→ We will simulate the number of bacteria in the jar after **1 hour**, assuming that initially there are **100 bacteria** present.

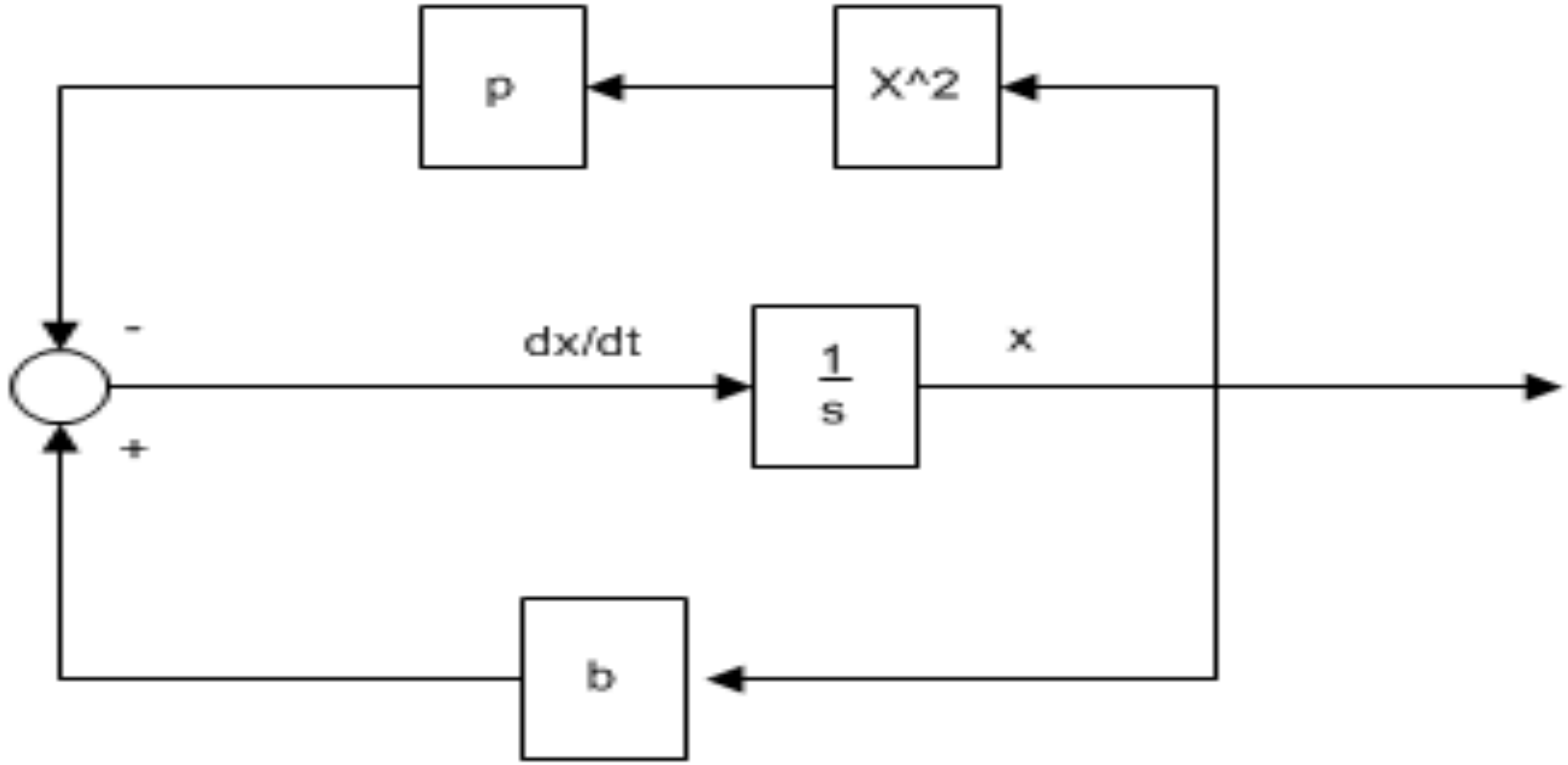
# In MATLAB We would do like this

```
function dx = bacteriadiff(t,x)
% My Simple Differential Equation

b=1;
p=0.5;

dx = b*x - p*x^2;
```

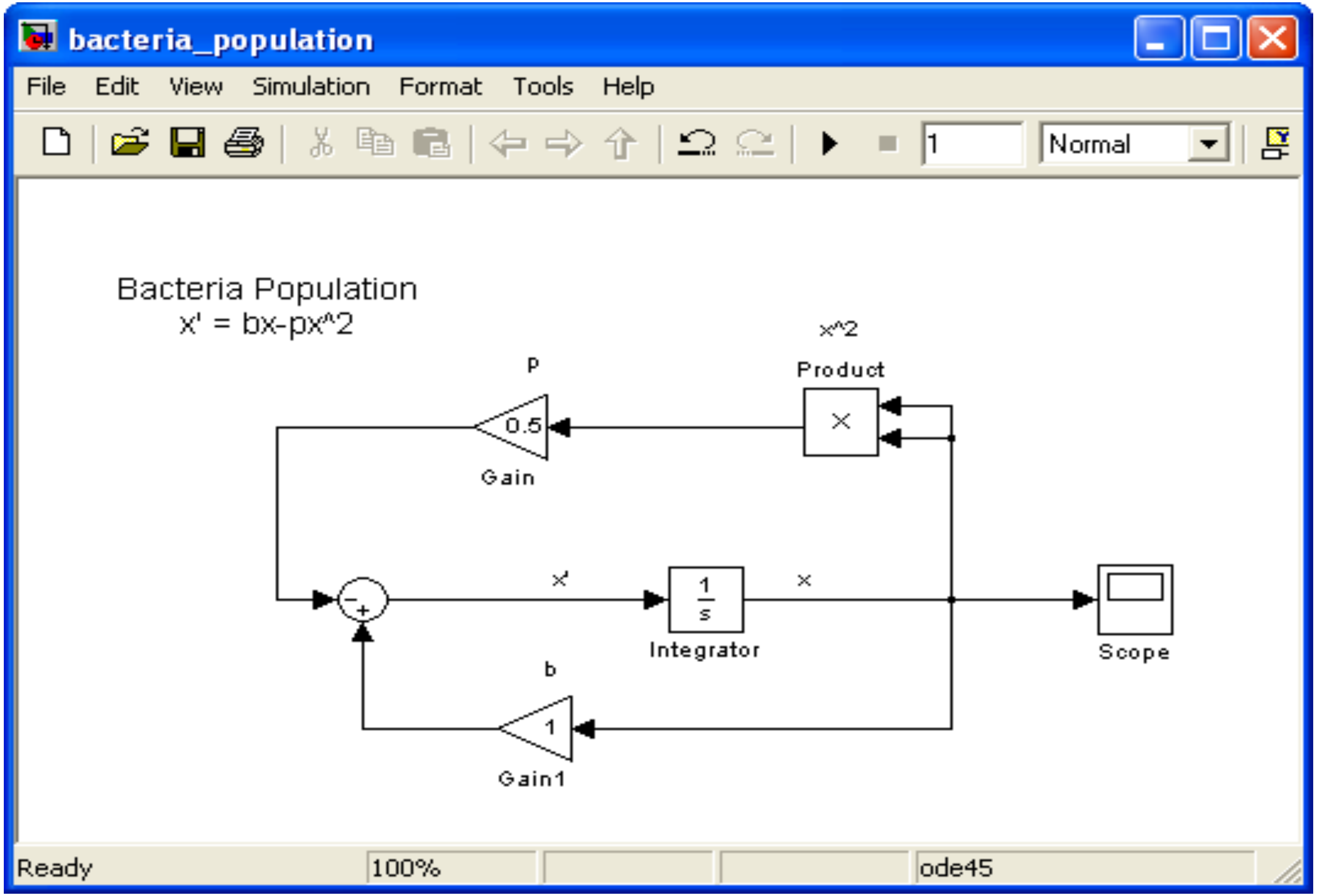
```
clear
clc

tspan=[0 1];
x0=100;

[t,y]=ode45(@bacteriadiff, tspan,x0);
plot(t,y)

[t,y]
```



# Block Diagram for the Model ("Pen and Paper")



# Simulink Block Diagram for the Model



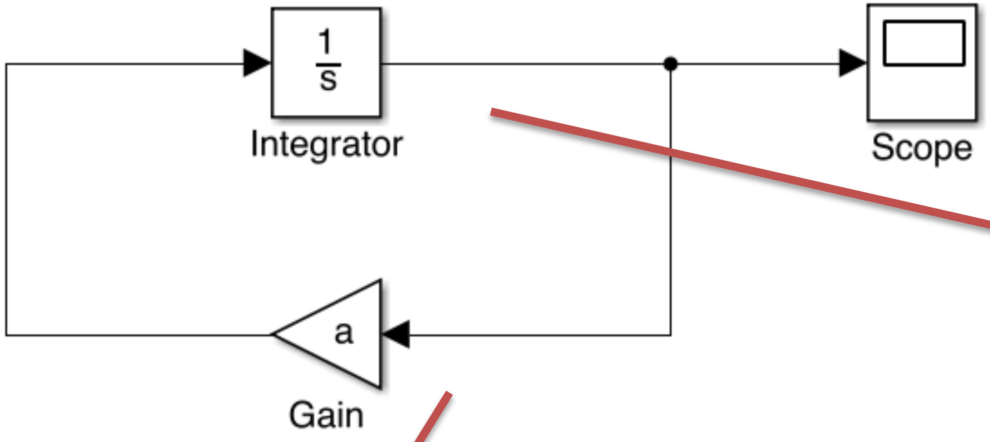
# Data-driven Modelling

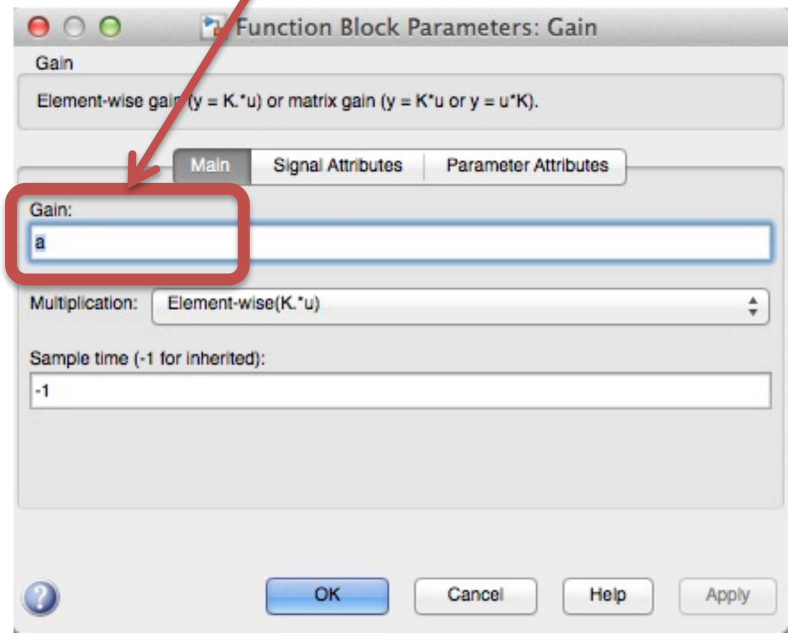
- You may use Simulink together with MATLAB in order to specify data and parameters to your Simulink model.
- You may specify commands in the MATLAB Command Window or as commands in an m-file (Script).
- This is called data-driven modeling
- Instead of using values directly we use variables instead - This is more flexible because we can easily change Simulation Parameters without touching the Simulink Model

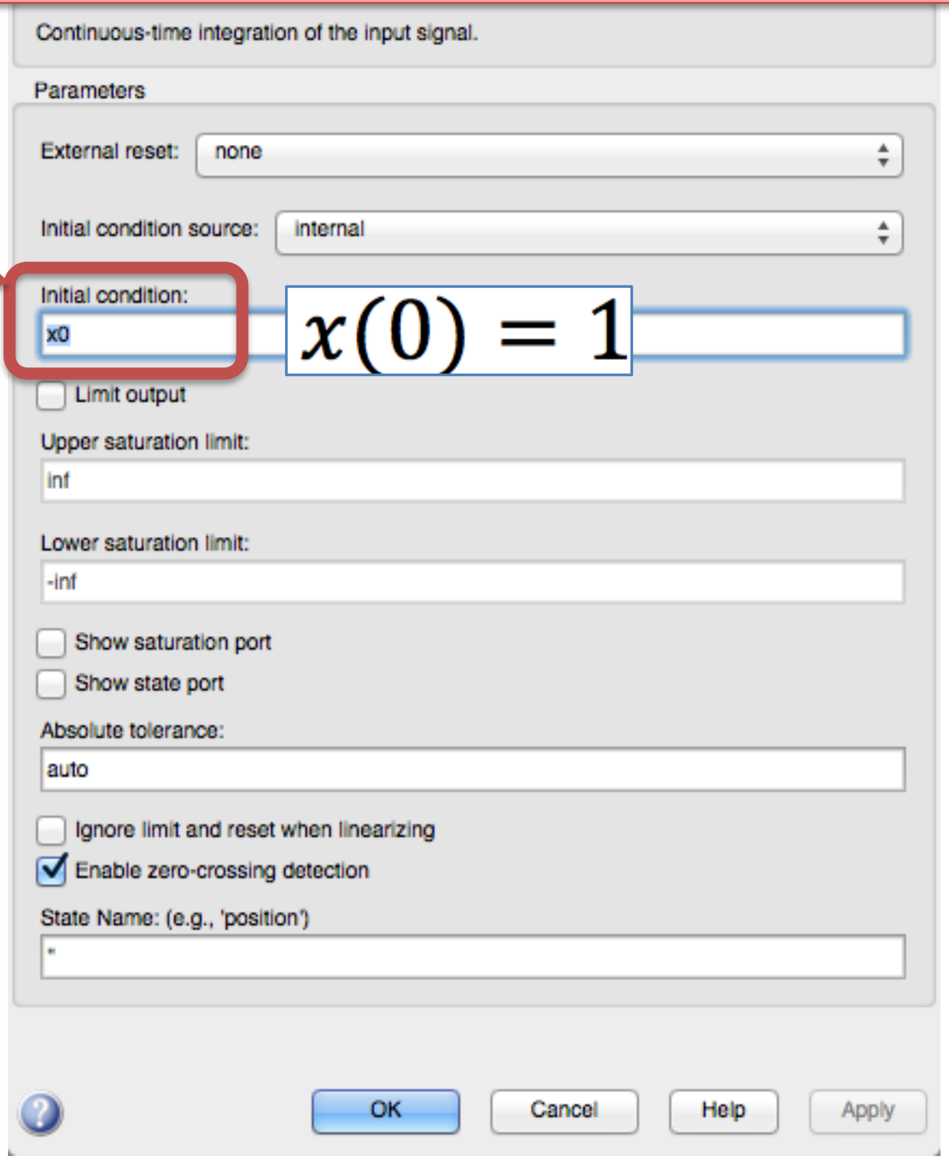
# Example

$$\dot{x} = ax$$

Instead of using values directly we use variables instead – This is more flexible because we can easily change Simulation Parameters without changing the Simulink Model



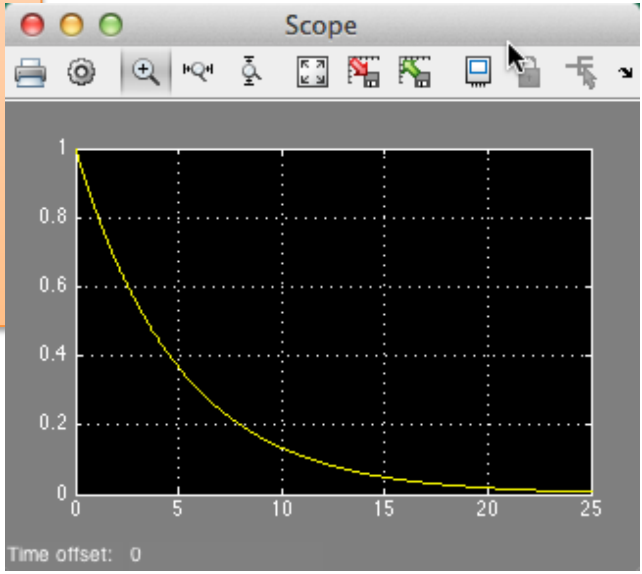
# Data-driven Modelling

MATLAB Script for running the Simulink Simulation:

```
clear
clc

%Set Simulator Settings
x0=1;
T=5;
a=-1/T;
t_stop=25; %[s]
T_s=t_stop/1000; %[s]
options=simset('solver', 'ode5', 'fixedstep', T_s);

%Starting Simulation
sim('simulink_ex2', t_stop, options);
```

This is the Name for our Simulink Model

We get the same results:

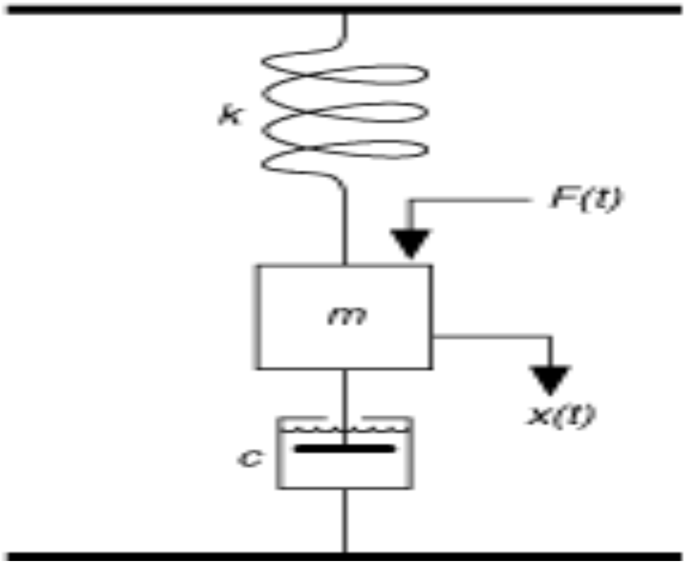


# Mass-Spring-Damper System

In this example we will create a mass-spring-damper model in Simulink and configure and run the simulation from a MATLAB m-file.

The differential equation for the system is as follows:

$$\ddot{x} = \frac{1}{m}(F - c\dot{x} - kx)$$



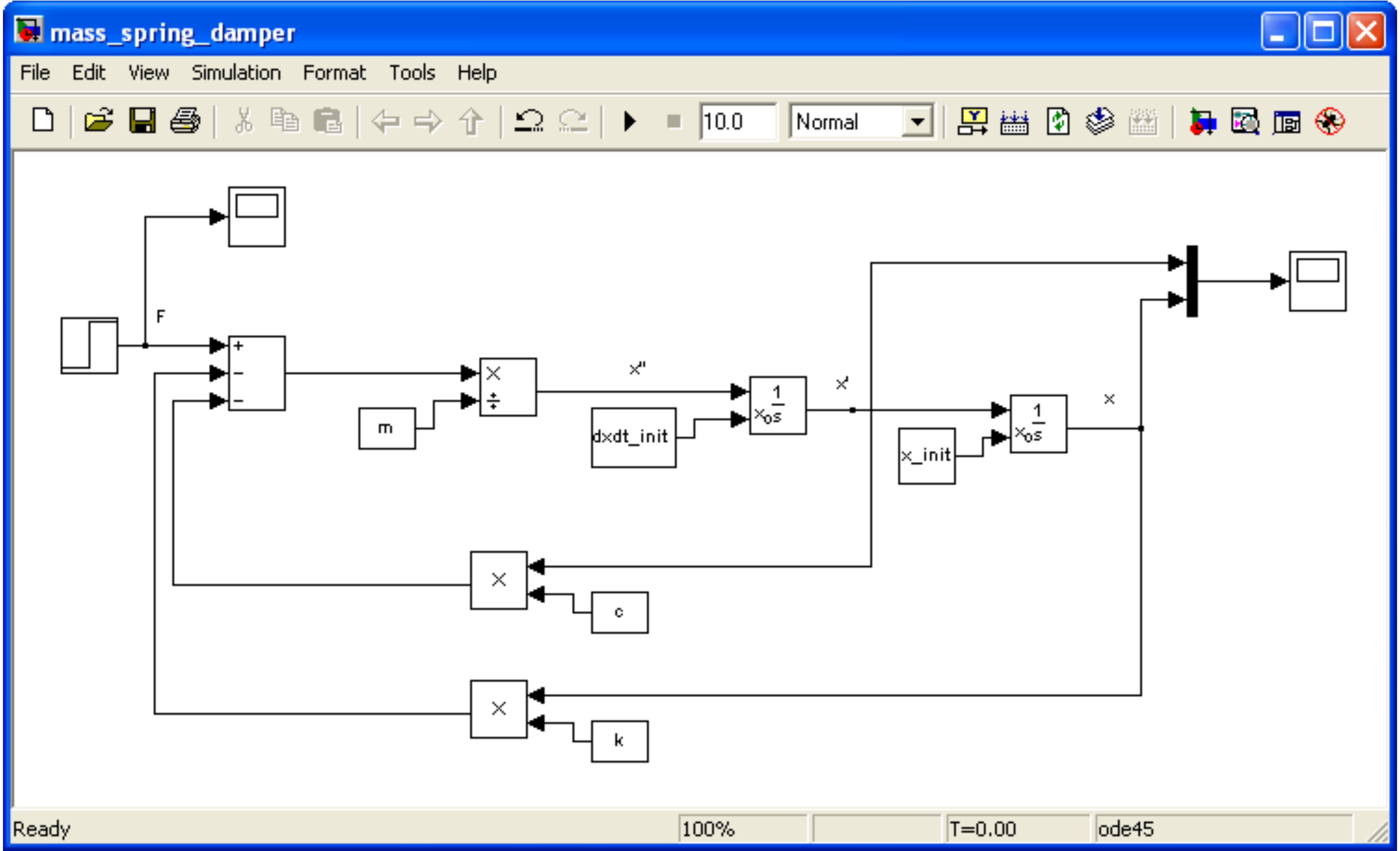

Where:

$x$ - position

$\dot{x}$ - speed

$\ddot{x}$ - acceleration

Instead of hard-coding the model parameters in the blocks you should refer to them as variables set in an m-file.



mass_spring_damper
File Edit View Simulation Format Tools Help
10.0
Normal
F
x''
x'
x
m
dxdt_init
x_init
1
s
c
k
Ready
100%
T=0.00
ode45

The following variables should then be set in the m-file:

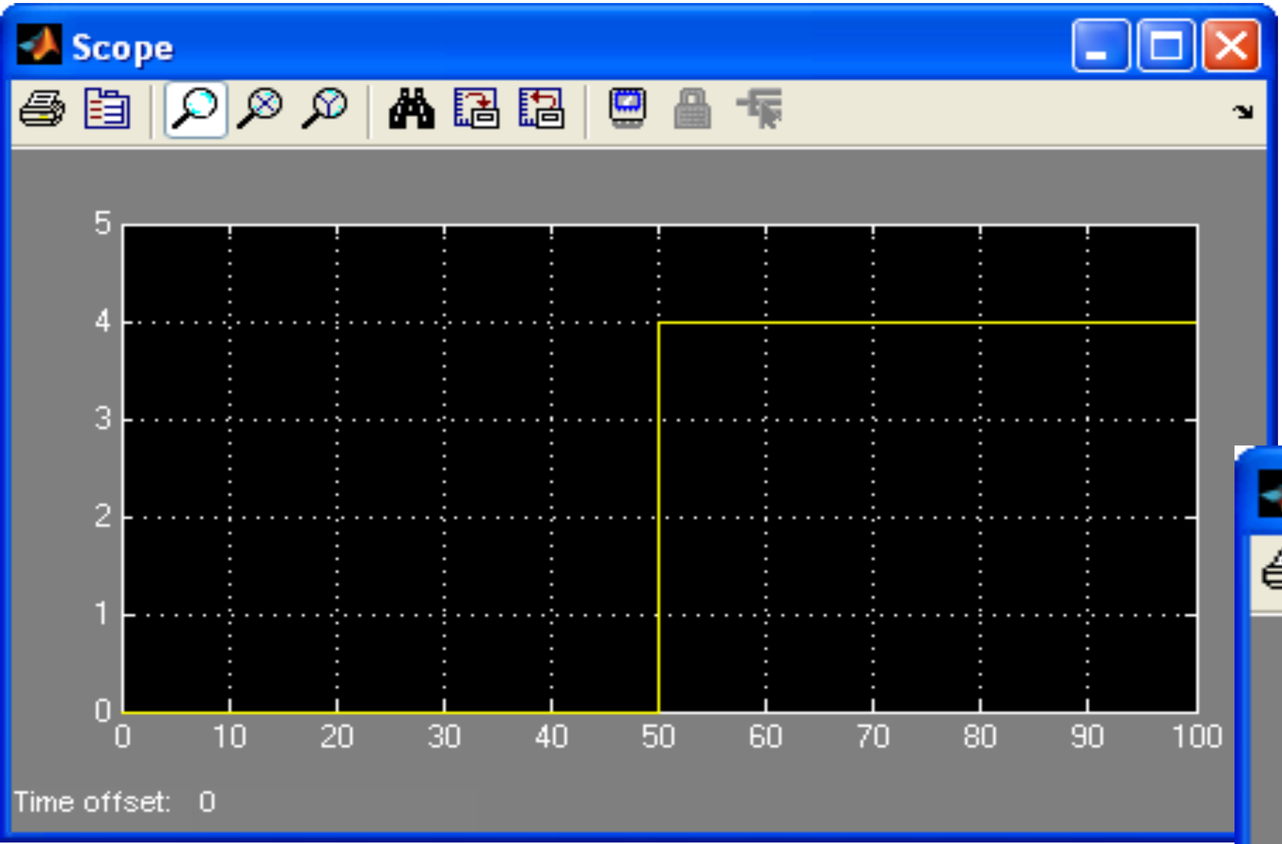
```
x_init = 4; %[m]. Initial position.
dxdt_init = 0; %[m/s]. Initial Speed.
m = 20; %[kg]
c = 4; %[N/(m/s)]
k = 2; %[N/m]
t_step_F = 50; %[s]
F_O = 0; %[N]
F_1 = 4; %[N]
```

Editor - M:\Work\Training\MATLAB\An Introduction to Simulink\Code\Mass-Spring-Dam...

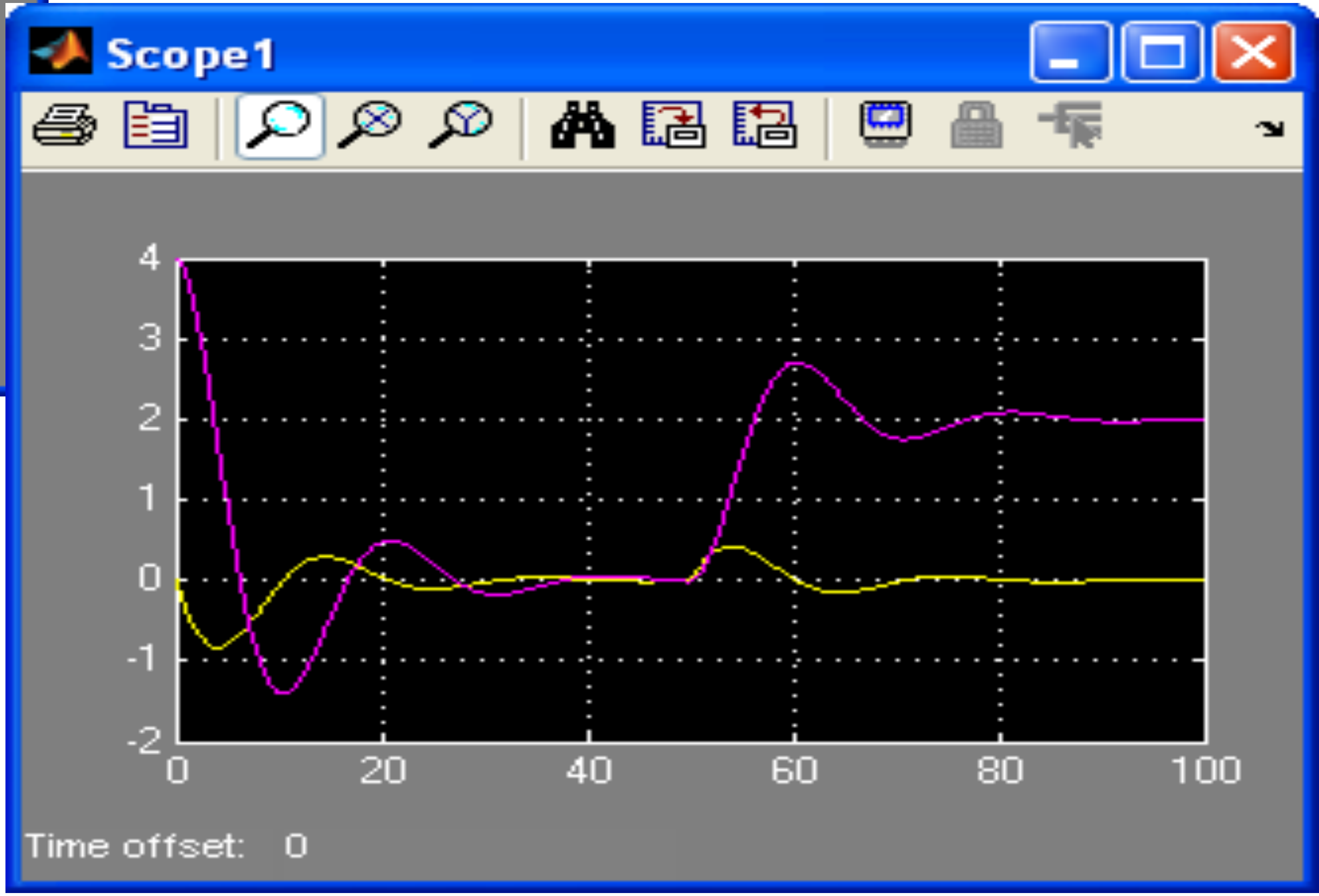
```matlab
%Script of mass-spring-damper simulator.
%Hans-Petter Halvorsen. 20.11.2009

%Modell Parameters
x_init=4; %[m]. Initial position.
dxdt_init=0; %[m/s]. Initial Speed.
m=20; %[kg]
c=4; %[N/(m/s)]
k=2; %[N/m]
t_step_F=50; %[s]
F_0=0; %[N]
F_1=4; %[N]

%Simulator Settings
t_stop=100; %[s]
T_s=t_stop/1000; %[s]
options=simset('solver', 'ode5', 'fixedstep', T_s);

%Starting simulation
sim('mass_spring_damper', t_stop, options);
```

script | Ln 16 | Col 22 | OVR

Force F:




Position $x$ and speed $\dot{x}$:



# Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog