Discrete Systems with MATLAB

Hans-Petter Halvorsen

Discrete Systems

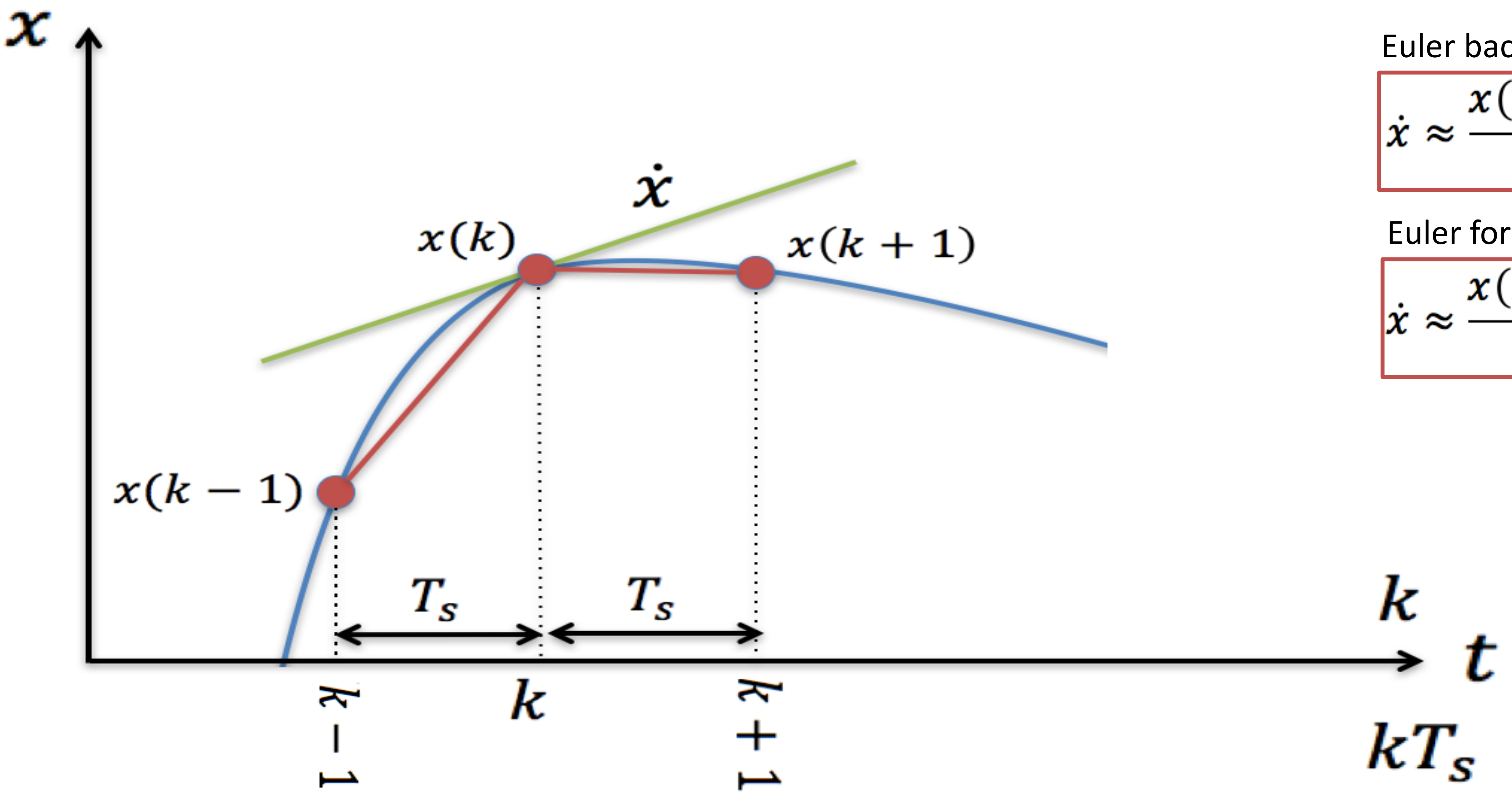
- MATLAB has built-in powerful features for simulation of continuous differential equations and dynamic systems.
- Sometimes we want to or need to discretize a continuous system and then simulate it in MATLAB.
- This means we need to make a discrete version of our continuous differential equations.
- Actually, the built-in ODE solvers in MATLAB use different discretization methods

Discretization Methods

- Euler;
  - Euler forward method,
  - Euler backward method
- Zero Order Hold (ZOH)
- Tustin
- …

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:

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

More Accurate!

Euler forward method:

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

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 literature!

Discrete Simulation

Given the following differential equation:

$$\dot{x} = ax$$

where $a = -\frac{1}{T}$, where T is the time constant

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

Find the discrete differential equation and plot the solution for this system using MATLAB.

Set $T = 5$ and the initial condition $x(0) = 1$.

Create a script in MATLAB (.m file) where we plot the solution $x(t)$ in the time interval $0 \le t \le 30$

Discrete Simulation

We can use e.g., the Euler Approximation:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

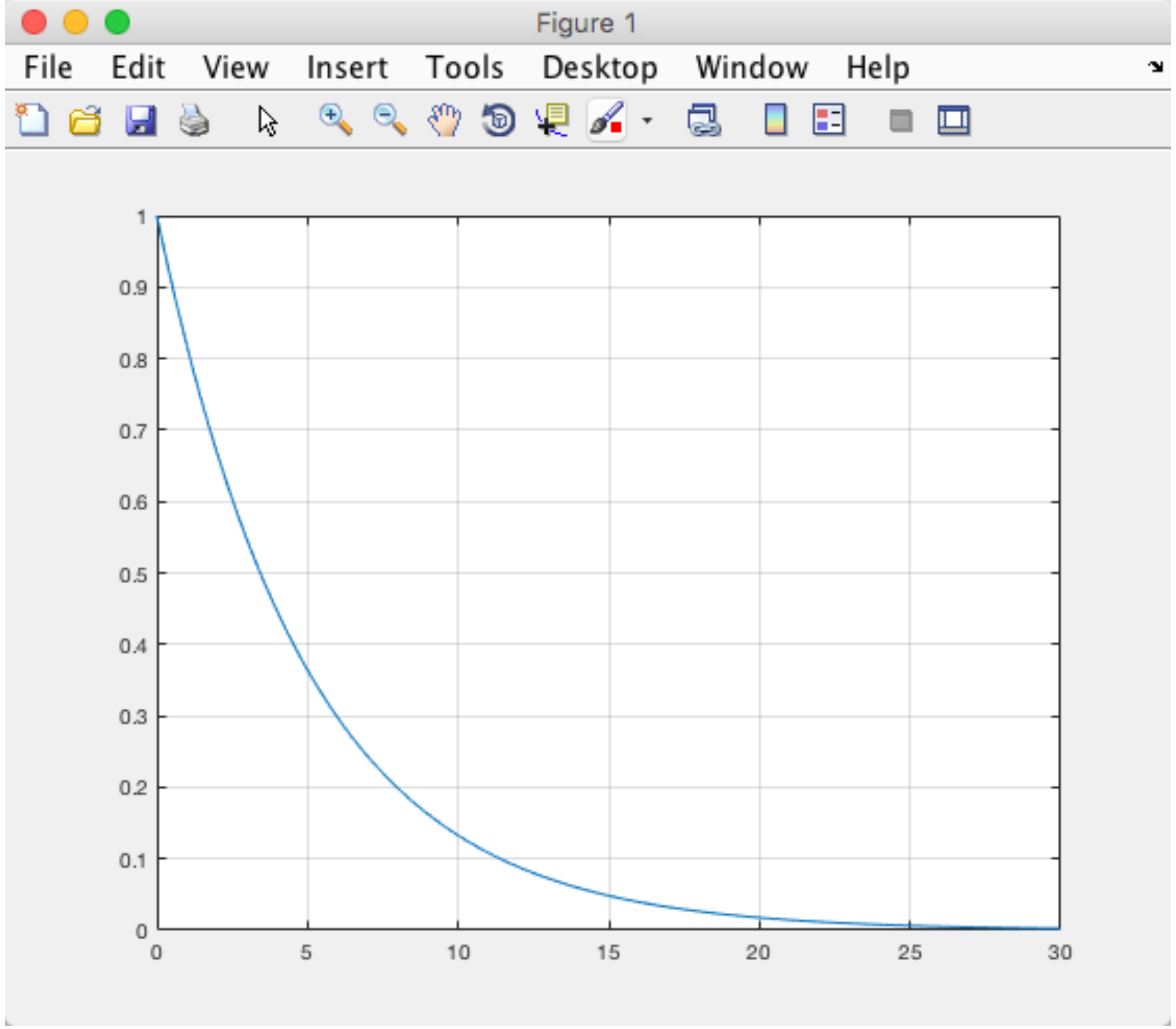
Then we get:

$$\frac{x_{k+1} - x_k}{T_s} = a x_k$$

Which gives:

$$x_{k+1} = (1 + aT_s) x_k$$

```
clear, clc

% Model Parameters
T = 5;
a = -1/T;

% Simulation Parameters
Ts = 0.1; %s
Tstop = 30; %s
x(1) = 1;

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

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



Bacteria Simulation

In this task 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.

→ Find the discrete model using the Euler Forward method by hand and implement and simulate the system in MATLAB using a For Loop.

Bacteria Simulation

We create a discrete model. and use Euler Forward differentiation method:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

Where T_s is the Sampling Time.

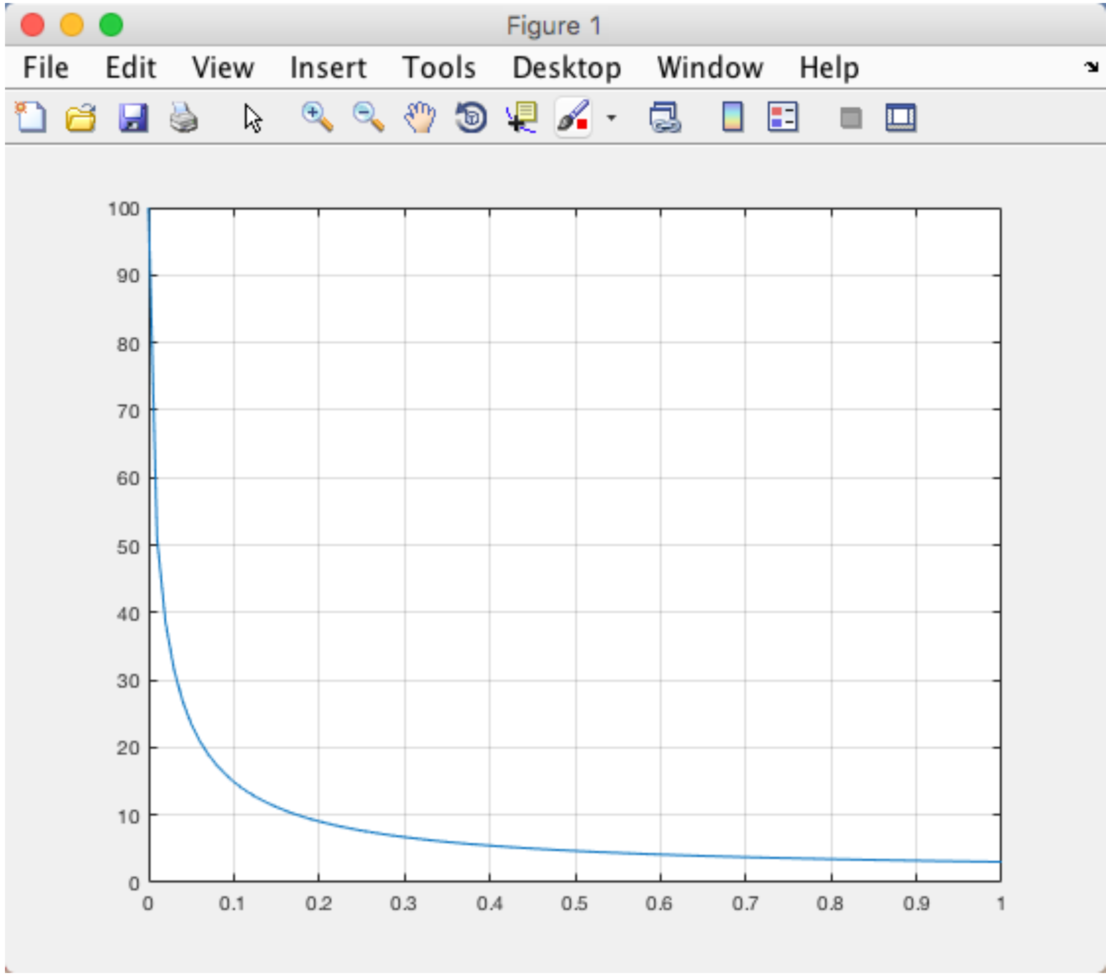
We get:

$$\frac{x_{k+1} - x_k}{T_s} = bx_k - px_k^2$$

This gives:

$$x_{k+1} = x_k + T_s(bx_k - px_k^2)$$

```
clear, clc
% Model Parameters
b = 1;
p = 0.5;

% Simulation Parameters
Ts=0.01;
Tstop = 1; %s
x(1)=100;

% Simulation Loop
for k=1:(Tstop/Ts)
    % Discrete model
    x(k+1) = x(k) + Ts*(b*x(k) - p*x(k)^2);
end

% Plot the simulation results
t=0:Ts:Tstop;
plot(t,x)
grid on
```



Simulation with 2 variables

Given the following system:

$$\frac{dx_1}{dt} = -x_2$$

$$\frac{dx_2}{dt} = x_1$$

Find the discrete system and simulate the discrete system in MATLAB.

Simulation with 2 variables

Using Euler:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

Then we get:

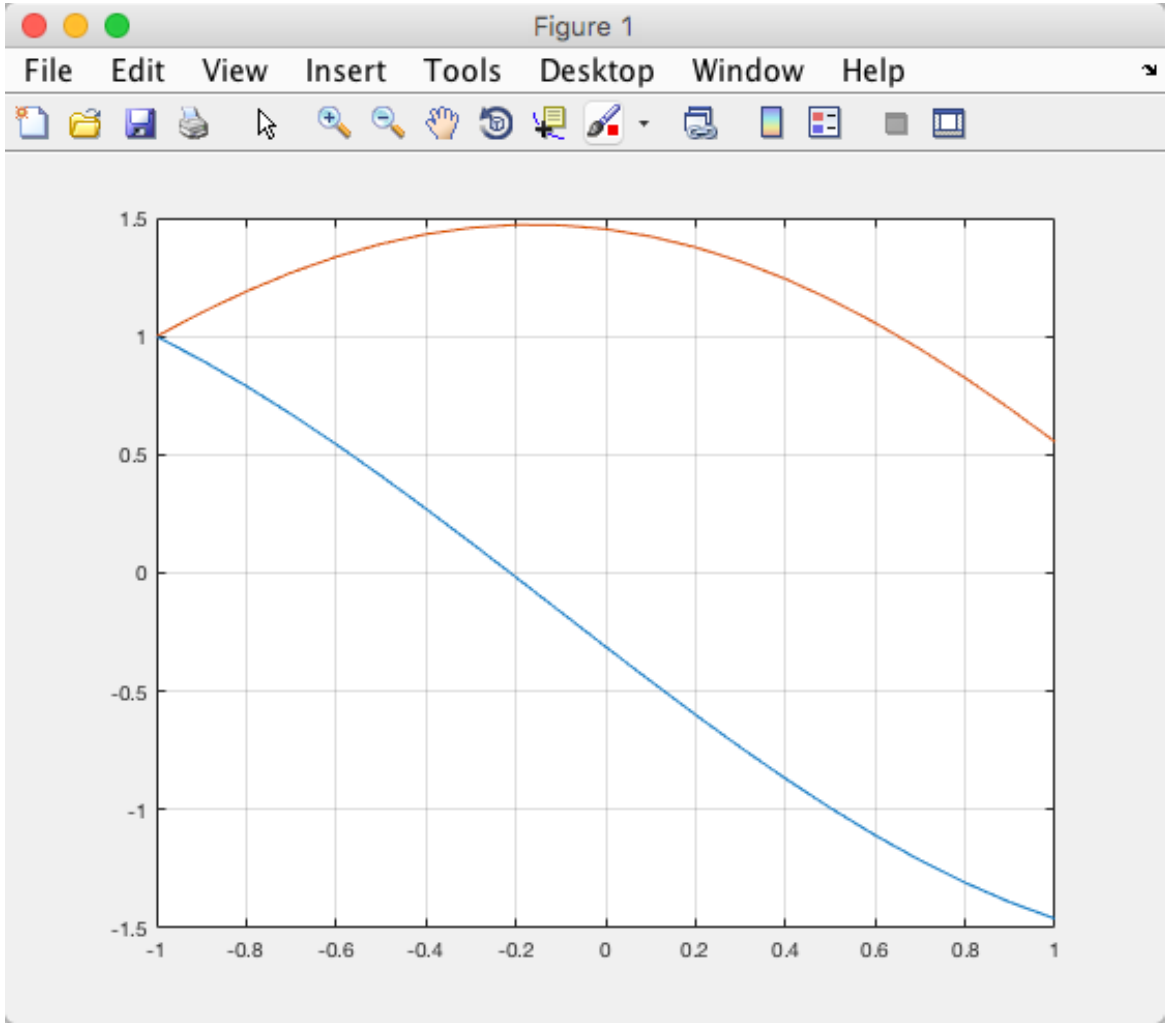
$$x_1(k+1) = x_1(k) - T_s x_2(k)$$

$$x_2(k+1) = x_2(k) + T_s x_1(k)$$

Which we can implement in MATLAB.

The equations will be solved in the time span $[-1\ 1]$ with initial values $[1, 1]$.

```
clear, clc

Ts = 0.1;

Tstart = -1;
Tstop = 1;

x1(1) = 1;
x2(1) = 1;

for k=1:((Tstop-Tstart)/Ts)
   x1(k+1) = x1(k) - Ts.*x2(k);
   x2(k+1) = x2(k) + Ts.*x1(k);
end

t = Tstart : Ts : Tstop;
plot(t,x1,t,x2)
grid on
```



Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog