Altair Compose Exercise – Implementation of Mass-Spring-Damper system

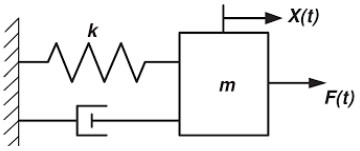
The student is asked to implement the Euler's method for numerical integration through oml language.

Theoretical References

The Euler's method for a problem $\ddot{x} = f(\dot{x}, x, t)$ requires to reply an iteration at following intervals of time. This iteration can be briefly expressed in this way:

$$
\begin{aligned}
\dot{x}_{t+\Delta t} &= \dot{x}_t + f(\dot{x}_t, x_t, t)\Delta t \\
x_{t+\Delta t} &= x_t + \dot{x}_{t+\Delta t}\Delta t \\
t &:= t + \Delta t
\end{aligned}
$$

In the case of a spring-mass-damper system, we can write $f(\dot{x}_t, x_t, t) = \frac{F\sin(\omega t) - c\dot{x}_t - kx_t}{m}$ if the model is: $m\ddot{x} + c\dot{x} + kx = F\sin(\omega t)$



COMPOSE Implementation

The implementation is not that hard. We just have to initialize some variables and constants and save the solutions (speed and displacement) and time of each iteration in dynamic vectors, which will be used to generate plots.
The iteration previously shown can be written in oml language as follows:

```
clear all
close all
clc
% variables initialization
t=0;            %s
x = 1.0;        %m
v= 0.0;         %m/s
% Constants
dt = 0.01;      %s
Tmax= 10.0;     %s
Fo = 50;         %N
m = 10;         %kg
f = 1.6;         %s^(-1)
c = 3.6;          %N/(m/s)
k= 100;          %N/m

% vectors initialization
```

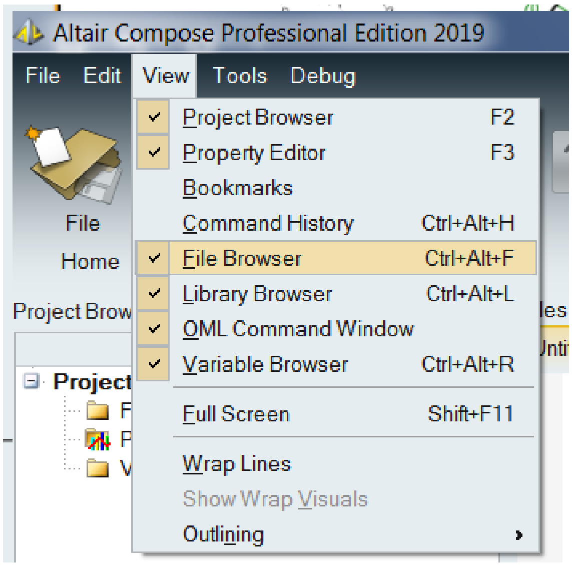
```
v_vec = v;
x_vec = x;
t_vec = t;

while t<=Tmax
    F = Fo*cos(2*pi*f*t)-k*x-c*v;
    a = F/m;
    v = v+a*dt;
    x = x+v*dt;
    t = t+ dt;
    v_vec = [v_vec;v];
    x_vec = [x_vec;x];
    t_vec = [t_vec;t];
end

plot(t_vec, x_vec, t_vec, v_vec)
xlabel('time [s]');
ylabel('displacement [m] , velocity [m/s]');
title(‘Mass-spring-damper’)
legend('X','V')
grid on;
```

Some notes:

- The line `v_vec = [v_vec;v];` is used to add an element at the end of a vector. In this way the vector gets longer at each cycle of the integration. We do that to save all the speed values over time in a vector v_vec which will be ploted at the end. The same is done for x_vec.
- It’s also useful to save all the time values t in a vector t_vec, which will be used as x-axis in the plots

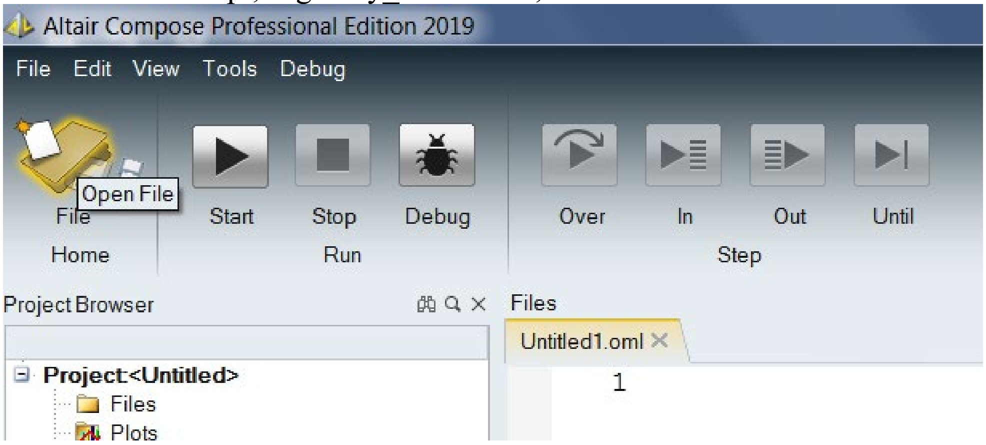
To use Compose:

1. run Compose
2. under tab “View” tick the features we want to enable. Here the most common ones:



3. move to the desired directory through the file browser of Compose

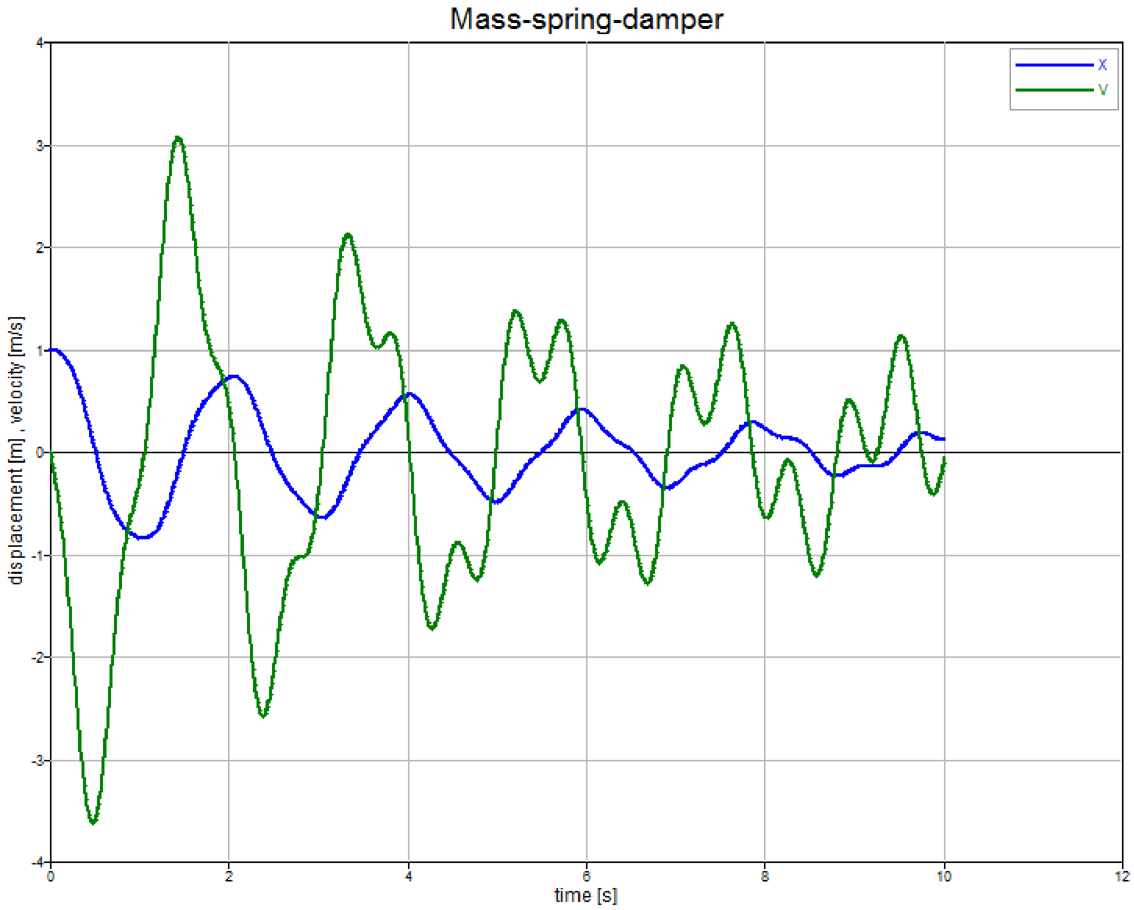
4. write an .oml script, E.g. “my_euler.oml”, in the Editor.



5. save the script .oml in the working directory

6. run my_euler.oml (clicking on the Start button in the Editor toolbar)
7. we should obtain the plots of speed and displacement (as we are considering a forced system the homogeneous response evanishes in a certain time and the solution is at this point only determined by the particular integral. Try to change some parameters value to better see it)

Result:



Exercise 1

Modify the script in order to obtain the displacement of the system with these properties:

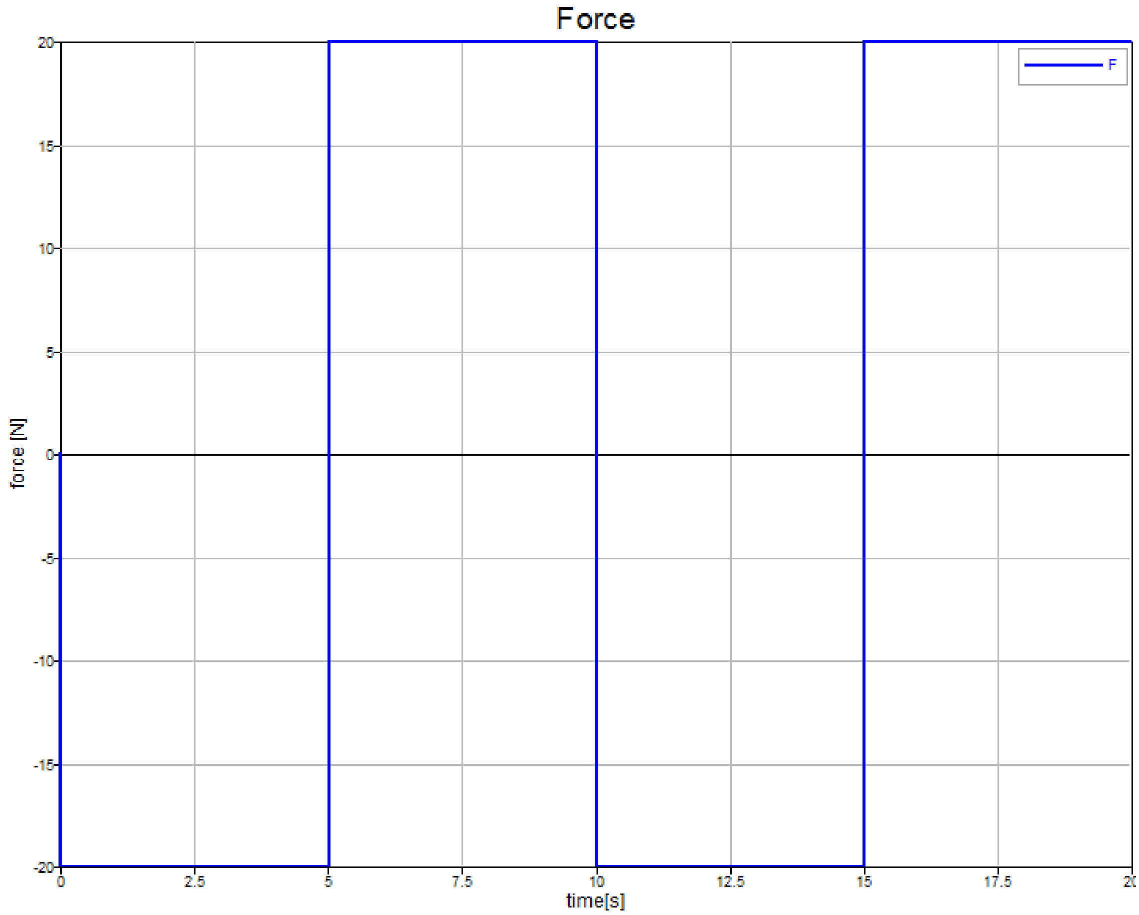
Initial conditions:
$x_0 = 0.0$ [m]
$v_0 = 2.0$ [m/s]
Harmonic force:
$F_0 = 3$ [N],
frequency = 4 [Hz]
System:
damping = 4 [N/(m/s)]
stiffness = 80 [N/m]
mass = 10 [kg]

Try to decrease time-step dt to improve accuracy; try to increase time-step dt until divergence condition.
Optional: Try to overlap plots we get for different time-steps.

Exercise 2

Same data of exercise 1 (solved) but this time the external force is a square-wave with amplitude $F_0 = 20$ [N] and frequency = 0.1 [Hz], and the stiffness is 180 [N/m].

Set the simulation time to 20 seconds.



Hint: use function mod() to obtain a periodic function, E.g. insert in the while cycle, in place of line Fo = ..., something similar to:

```
if mod(t,1.0/f) < (0.5*1.0/f)
        …
else
        …
end
```