Solving Linear Ordinary Differential Equations with Matlab's symbolic toolbox: spring-mass-damping example

This code executed in Matlab `R2022a`

Video-presentations:

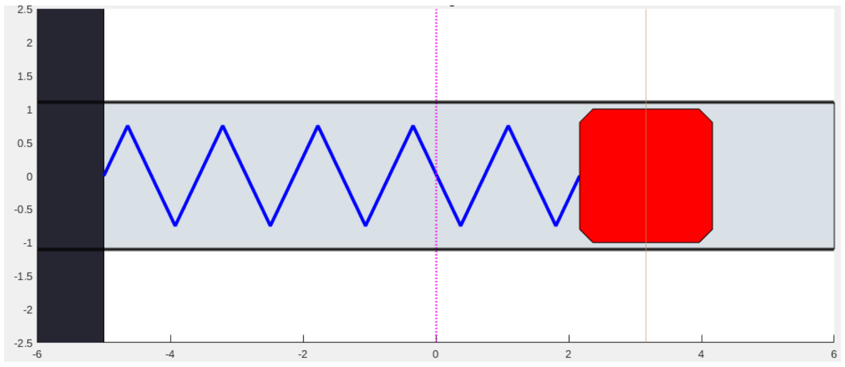
http://personales.upv.es/asala/YT/V/masmusym1EN.html , http://personales.upv.es/asala/YT/V/masmusym2EN.html

Objectives: analyzing the solution of linear ODEs provided by Matlab (symbolic toolbox), understanding its meaning, and representing it graphically in a spring-damper (free response) model, and understanding a very simple animation code.

Table of Contents

EDO in non-normalized (second derivatives) form

Consider the following spring (with some damping, too, by friction with walls and spring losses):



Let us solve the equation of a mass-spring-damper system (linearized, zero equilibrium; linear viscous friction assumed), starting at initial conditions out of equilibrium, written as:

$$M\frac{d^2y}{dt^2} = -ky - b\frac{dy}{dt}$$

```
syms y(t) %symbolic function of time
syms M k b real %constant parameters
vel=diff(y); % notation for velocity
accel=diff(y,2); % notation for acceleration

ODE_SYM= M*accel==-k*y-b*vel
```

```
ODE_SYM(t) =
```

$$M\,\frac{\partial^2}{\partial t^2}\,y(t) = -b\,\frac{\partial}{\partial t}\,y(t) - k\,y(t)$$

```
%if it were a nonlinear ODE, a solution could not, in general, be found in the form of
M_num=0.5;k_num=1;b_num=0.2; %numerical values for constant parameters
%let's replace symbols by actual numbers
ODE_mass_spring=subs(ODE_SYM, {M,k,b}, {M_num,k_num,b_num})
```

```
ODE_mass_spring(t) =
```

$$\frac{\frac{\partial^2}{\partial t^2}\,y(t)}{2} = -\frac{\frac{\partial}{\partial t}\,y(t)}{5} - y(t)$$

General solution

With no fixed initial conditions, there are infinitely many solutions, formally related to two integration constants:

```
sol=simplify(dsolve(ODE_mass_spring))
```

```
sol =
```

$$e^{-\frac{t}{5}}\left(C_1\cos\left(\frac{7\,t}{5}\right) - C_2\sin\left(\frac{7\,t}{5}\right)\right)$$

This is the so-called general solution, where C_1 and C_2 influence initial conditions.

Decay rate: 1/5 s^{-1}, free oscillation frequency: 7/5 rad/s.

```
sol_velocity=simplify(diff(sol))
```

```
sol_velocity =
```

$$-\frac{e^{-\frac{t}{5}}\left(C_1\cos\left(\frac{7\,t}{5}\right) - C_2\sin\left(\frac{7\,t}{5}\right)\right)}{5} - e^{-\frac{t}{5}}\left(\frac{7\,C_2\cos\left(\frac{7\,t}{5}\right)}{5} + \frac{7\,C_1\sin\left(\frac{7\,t}{5}\right)}{5}\right)$$

Let's test that it verifies the ODE:

```
simplify( M_num*diff(sol,2) == -k_num*sol-b_num*diff(sol) )
```

ans = symtrue

```
subs(sol,t,0)%initial position
```

ans = C_1

```
subs(sol_velocity,t,0)%initial velocity
```

ans =

$$-\frac{C_1}{5}-\frac{7\,C_2}{5}$$

Particular solution for given intial conditions (position and speed)

```
position=dsolve(ODE_mass_spring, y(0)==3, vel(0)==2)
```

position =

$$\frac{\mathrm{e}^{-\frac{t}{5}}\left(21\cos\left(\frac{7\,t}{5}\right)+13\sin\left(\frac{7\,t}{5}\right)\right)}{7}$$

Let's check that the prescribed initial conditions are indeed met:

```
subs(position,t,0)
```

ans = 3

```
vel=simplify(diff(position))
```

vel =

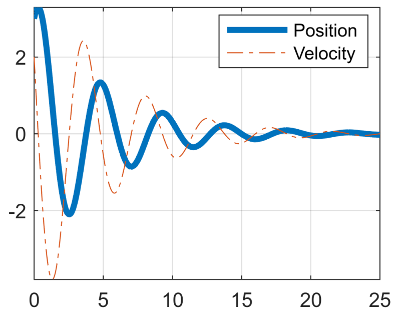
$$\frac{2\,\mathrm{e}^{-\frac{t}{5}}\left(7\cos\left(\frac{7\,t}{5}\right)-16\sin\left(\frac{7\,t}{5}\right)\right)}{7}$$

```
subs(vel,t,0)
```

ans = 2

If we draw position (thick blue) and speed (dashed red), we get:

```
Tf=25; %the final time at which we wish to stop plotting
fplot(position,[0 Tf],LineWidth=3), hold on
fplot(vel,[0 Tf],LineStyle='-.'), hold off, grid on,
legend(["Position","Velocity"], location='best')
```



ODE in normalised form (dx/dt=Ax)

We pursue a model in the form $\frac{dx}{dt} = Ax$ (no input, so no need of $\frac{dx}{dt} = Ax + Bu$ for the moment being).

We check that we get coincident results.

```
syms pos(t) vel(t) %position and velocity
```

We express the mass-spring-damping model as a pair of first-order differential equations:

```
ODE_SYM= [ diff(pos) == vel;   diff(vel) == -k/M*pos-b/M*vel ]
```

```
ODE_SYM(t) =
```

$$\begin{pmatrix} \frac{\partial}{\partial t}\ \mathrm{pos}(t) = \mathrm{vel}(t) \\ \frac{\partial}{\partial t}\ \mathrm{vel}(t) = -\frac{b\ \mathrm{vel}(t)}{M} - \frac{k\ \mathrm{pos}(t)}{M} \end{pmatrix}$$

We can express the same thing in matrix form (because the equations are linear):

```
x=[pos; vel]; %state fector
A=[0 1;-k/M -b/M];
ODE_SYM=    diff(x) == A*x %normalised matrix form representation of the ODE
```

```
ODE_SYM(t) =
```

$$\begin{pmatrix} \frac{\partial}{\partial t}\ \mathrm{pos}(t) = \mathrm{vel}(t) \\ \frac{\partial}{\partial t}\ \mathrm{vel}(t) = -\frac{b\ \mathrm{vel}(t)}{M} - \frac{k\ \mathrm{pos}(t)}{M} \end{pmatrix}$$

Matrix form is convenient for subsequent theoretical analysis (not in the scope of this material).

General solution x=[pos; vel];

We will solve with `dsolve` command, once the numerical values of the constant parameters have been replaced:

```
ODE_mass_spring=subs(ODE_SYM, {M,k,b}, {M_num,k_num,b_num})
```

```
ODE_mass_spring(t) =
```

$$\begin{pmatrix} \frac{\partial}{\partial t}\ \mathrm{pos}(t) = \mathrm{vel}(t) \\ \frac{\partial}{\partial t}\ \mathrm{vel}(t) = -2\ \mathrm{pos}(t) - \frac{2\ \mathrm{vel}(t)}{5} \end{pmatrix}$$

```
A_numeric=subs(A, {M,k,b}, {M_num,k_num,b_num}) %this is the matrix A whose eigenvalues
```

```
A_numeric =
```

$$\begin{pmatrix} 0 & 1 \\ -2 & -\frac{2}{5} \end{pmatrix}$$

```
sol=dsolve(ODE_mass_spring)
```

```
sol = struct with fields:
    vel: C1*cos((7*t)/5)*exp(-t/5) - C2*sin((7*t)/5)*exp(-t/5)
    pos: C2*((7*cos((7*t)/5)*exp(-t/5))/10 + (sin((7*t)/5)*exp(-t/5))/10) - C1*((cos((7*t)/5)*exp(-t/5))/1
```

```
simplify(sol.pos)
```

```
ans =
```

$$\frac{e^{-\frac{t}{5}}\left(7\,C_2\cos\left(\frac{7\,t}{5}\right) - C_1\cos\left(\frac{7\,t}{5}\right) + 7\,C_1\sin\left(\frac{7\,t}{5}\right) + C_2\sin\left(\frac{7\,t}{5}\right)\right)}{10}$$

```
simplify(sol.vel)
```

```
ans =
```

$$e^{-\frac{t}{5}}\left(C_1\cos\left(\frac{7\,t}{5}\right) - C_2\sin\left(\frac{7\,t}{5}\right)\right)$$

It's the general solution. Initial conditions would be

```
subs(sol,t,0)
```

```
ans = struct with fields:
    vel: C1
    pos: (7*C2)/10 - C1/10
```

Particular solution for given initial conditions

```
spring_state=dsolve(ODE_mass_spring, pos(0)==3, vel(0)==2);
simplify(spring_state.pos)
```

```
ans =
```

$$\frac{e^{-\frac{t}{5}}\left(21\cos\left(\frac{7\,t}{5}\right)+13\sin\left(\frac{7\,t}{5}\right)\right)}{7}$$

```
simplify(spring_state.vel)
```

```
ans =
```

$$\frac{2\,e^{-\frac{t}{5}}\left(7\cos\left(\frac{7\,t}{5}\right)-16\sin\left(\frac{7\,t}{5}\right)\right)}{7}$$

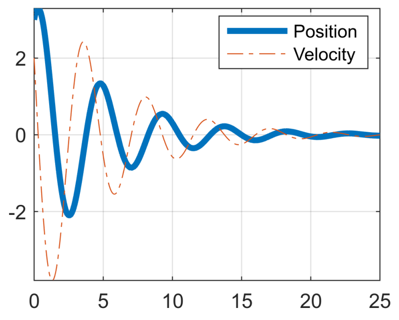
Prescribed initial conditions do hold:

```
subs(spring_state,t,0)
```

```
ans = struct with fields:
    vel: 2
    pos: 3
```

Let us plot the resulting time response (of course, identical to the one at first section):

```
fplot(spring_state.pos,[0 Tf],LineWidth=3), hold on
fplot(spring_state.vel,[0 Tf],LineStyle='-.'), hold off, grid on, legend(["Position","V
```


Position
Velocity