To apply Euler's method to our system of first order equations

$$\frac{dx}{dt} = v(t), \qquad \frac{dv}{dt} = -10\,kx(t), \qquad v(0) = 0, \qquad x(0) = 0.1, \quad k = 50$$

we must use a very small step size because of the numerical instability due to the steepness of the graph. We will use a step size of
h = 0.001 in order to give a numerical solution.

To execute the definitions and code in red below, click at the end of the code line and press return. When Maple is running a computation, you will see a stop sign in the menu bar at the top of the page. You can stop the computation by clicking the stop sign. Please wor through this page and
print out the solutions. Then copy the worksheet and change the values of h out what happens for smaller values of h. You will see the numerical
instability caused by steep slopes and many oscillations.

First, define the functions that give the value of the slope:

```
> restart;
> y := (x, v, t) → v;
```

$$y := (x, v, t) \rightarrow v \tag{1}$$

```
> a := (x, v, t) → -500*x;
```

$$a := (x, v, t) \rightarrow -500\,x \tag{2}$$

Then, define the step size and the initial values for each variable and for t.

```
> h := 0.001;
```

$$h := 0.001 \tag{3}$$

```
> t[0] := 0;  x[0] := 0.1;  v[0] := 0;
```

$$t_0 := 0$$

$$x_0 := 0.1$$

$$v_0 := 0 \tag{4}$$

Then for each variable, the new value is ``old value + step size times slope."
So after one step, the values are calculated as follows:

```
> t[1] := t[0] + h:
  x[1] := x[0] + h*y(x[0],v[0],t[0]):
  v[1] := v[0] + h*a(x[0],v[0],t[0]):
  [x[1],v[1],t[1]];
```

$$[0.1, -0.0500, 0.001] \tag{5}$$

To perform a large number of iterations (5000), we
write a for-loop to do all the calculations at once.

The for loop creates a sequence of points in the t-x plane.

We start by defining the the step size and number of steps:

```
> N := 5000; h := 0.001;
```

$$N := 5000$$

$$h := 0.001 \tag{6}$$

Next define the initial values:

```
> t[0] := 0; x[0] := 0.1; v[0] := 0;
```

Then define the first point in each of the three sequences.

```
> euler_ptsx := [t[0],x[0]];
> for n from 0 to N-1 do
>    t[n+1] :=  t[n] + h;
>    x[n+1] := x[n] + h*y(x[n],v[n],t[n]);
>    v[n+1] := v[n] + h*a(x[n],v[n],t[n]);
```

new value = old value + stepsize * slope

```
>    new_ptx := [ t[n+1], x[n+1] ];
```

define the newest points in the three sequences of points.

```
>    euler_ptsx := euler_ptsx, new_ptx;
```

modify each of the three sequences by adding the new point at the end.

```
> od:
```

$$t_0 := 0$$

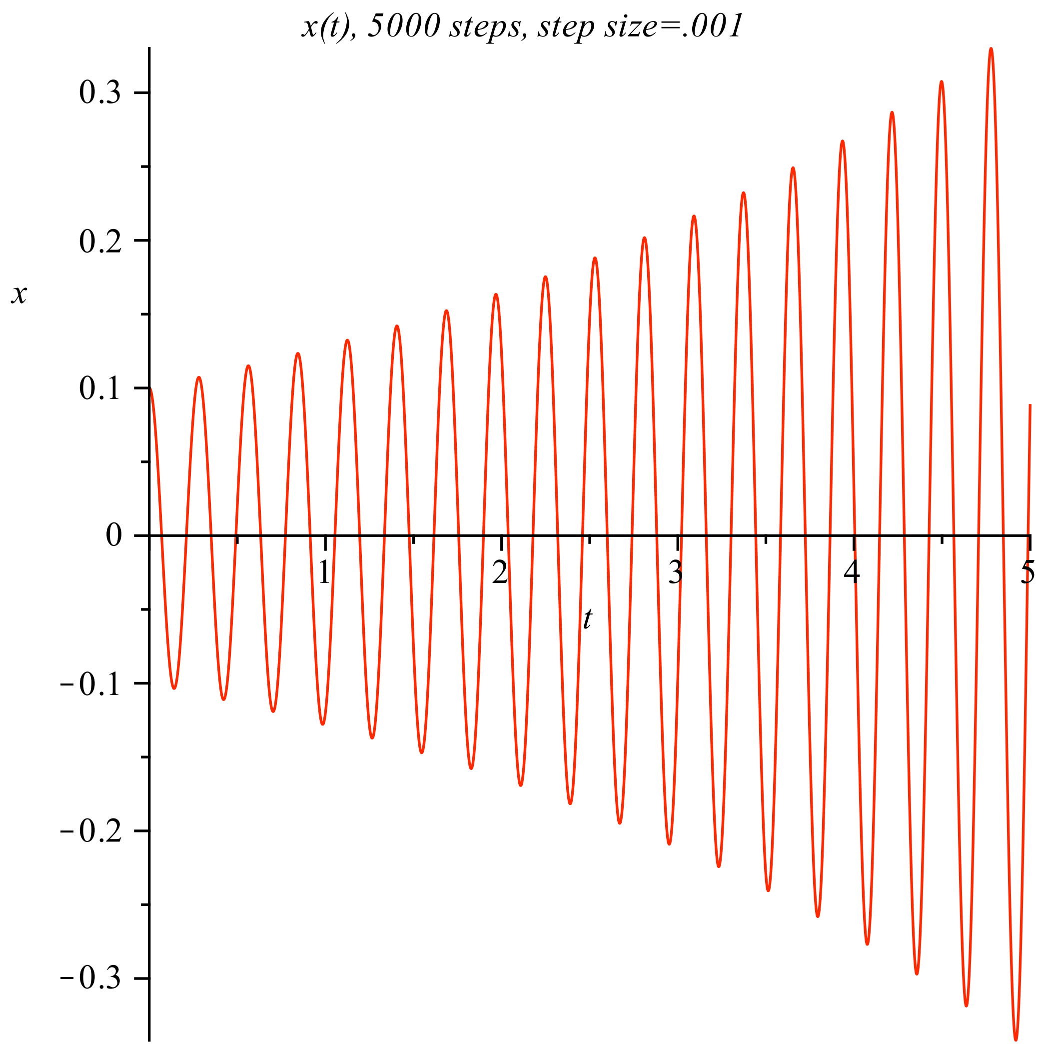
$$x_0 := 0.1$$

$$v_0 := 0$$

$$euler\_ptsx := [0, 0.1] \tag{7}$$

The colon at the end of the for loop suppresses all output from calculations within the for-loop. We can see the list of points by executing, for example, ``euler_ptsx;" to see the list of points in the t-x plane. The code below generates the graph of our numerical solution.

```
> plot([euler_ptsx],labels = [`t`,`x`],
  title = `x(t), 5000 steps, step size=.001`);
```


x(t), 5000 steps, step size=.001
0.3
0.2
x
0.1
0
1
2
3
4
5
t
-0.1
-0.2
-0.3

>