How to Plot a Direction Field with `dirfield.m`

- Download `dirfield.m` from Canvas. Save it in a directory that Matlab can find. (Wherever you save it, make sure that the directory is in your Matlab path.)

- `dirfield.m` contains a function called `dirfield`. This function takes 3 arguments:

  1. The first function takes in another function $f(t,x)$ where $x' = f(t,x)$. Notice that in our model x' is actually a function of only x and not t. That's okay. The easiest way to create this function is to use an *anonymous function*.

     For example, if $f(t,x) = x^2$ we could type

     ```
     f = @(t,x) x.^2;
     ```

     Of course, you will need to change this to contain the actual function you want instead of just x^2. In Matlab, you can define the anonymous function f either in the Command Window or in a script.

  2. The second argument is the t-values, given in the vector form

     ```
     t_start : t_step : t_end
     ```

     For example, if you needed to define a t-vector that starts at 0, ends at 30, and has a step-size of 1, you could type

     ```
     0 : 1 : 30
     ```

  3. The third argument the x-values, given in the vector form

     ```
     x_start : x_step : x_end
     ```

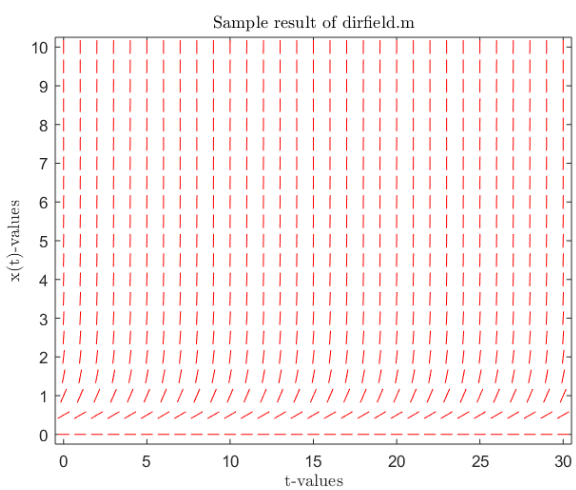
     For example, if you needed to define an x-vector that starts at 0, ends at 10, and has a step-size of 0.5, you could type

     ```
     0 : 0.5 : 10
     ```

- To run this code for the example problem given in steps 1-3 above, you could type in the Command Window or a script

  ```
  f = @(t,x) x.^2;
  dirfield( f, 0:1:30, 0:0.5:10)
  ```

  The resulting direction field would be



Adjust the input function, t-values, and x-values until you hae the direction field that you want.

How to Plot a Vector Field with `flow.m`

- Download `flow.m` from Canvas. Save it in a directory that Matlab can find. (Wherever you save it, make sure that the directory is in your Matlab path.)

- `flow.m` is a script that generates the vector field for a system of equations of the form

$$\frac{dx_1}{dt} = f(x_1, x_2), \qquad \frac{dx_2}{dt} = g(x_1, x_2)$$

  For example, we might want to find the vector field of the system

$$\frac{dx_1}{dt} = ax_2, \qquad \frac{dx_2}{dt} = -x_1$$

  Since `flow.m` is not a function, it does not take any arguments. However, there are 5 lines of code that must be adjusted to match the specific problem you want to visualize:

  1. Step 1 defines that ranges of x_1 and x_2 values that will be used in the vector field. The ranges of values are defined with the variables `x1min`, `x1max`, `x2min`, and `x2max`.

     For example, if we want to plot the vector field over the ranges $-5 < x_1 < 5$ and $-5 < x_2 < 5$ we change the values in this line of code to

     ```
     x1min = -5; x1max = 5; x2min = -5; x2max = 5;
     ```

  2. Step 2 defines the step size between points in the x_1 and x_2 intervals. These values are defined with the variables `x1step` and `x2step`.

     For example, if we want to use a step size of 1 for both the x_1 and x_2 values, we would type

     ```
     x1step = 1; x2step = 1;
     ```

  3. Step 3 defines all of the parameters used in the system of equations:

     ```
     a = 1;
     ```

  4. Step 4 defines the differential equations $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ that we want to find the vector field for:

     ```
     dx1 = a*x2;
     dx2 =  -x1;
     ```

  5. Step 5 labels the x_1 and x_2 axes and titles the plot.

     ```
     xlabel('$x1$','Interpreter','latex')
     ylabel('$x2$','Interpreter','latex')
     title('Vector field example','Interpreter','latex')
     ```

  The vector field that would result from this example is