# Direction fields and contour plots with MatLab



## 1. Direction fields.

To sketch a direction field for a first order differential equation with MATLAB, the equation must be in **normal form.** A first order differential equation is in normal form if it looks like this

$$y' = f(t, y).$$

To sketch a direction field, we use the MATLAB functions **meshgrid** and **quiver**. Briefly, **meshgrid** creates a grid of points in the $(t, y)$–plane, and **quiver** plots little vectors at each point, whose directions are determined by the right-hand side of the differential equation.

The **basic** syntax is something like

>> [T Y] = meshgrid(minT:step:maxT, minY:step:maxY);

%% Creates a uniformly spaced grid of points in the rectangle

$$\{(T, Y) : minT \leq T \leq maxT, \quad minY \leq Y \leq maxY\}.$$

%% The spacing is determined by the parameter 'step'.

>> dY = f(T,Y);

%% computes the slope of the vector at each point in the grid. Note that
%% you need to type in an actual function of $T$ and $Y$ here (not just write
%% 'f(T,Y)'). dY is a matrix of slopes.

>> dT = ones(size(dY));

%% This creates a matrix of the same dimension as dY of 1's.

>> quiver(T,Y, dT, dY);

%% The vector that the 'quiver' command plots at the point (T,Y) in the
%% grid will be parallel to the vector (dT,dY) = (1,dY), giving it the
%% correct slope. Quiver automatically scales the vectors so that they
%% do not overlap.

I'll use the syntax above to sketch a direction field for the differential equation $y' = \cos 2t - y/t$ in the rectangle $\{(t, y) : 0 < t < 8, \ -4 < y < 4\}$:

```
>> [T Y]=meshgrid(0:0.2:4,-2:0.2:2);
>> dY=cos(2*T)-Y./T;
>> dT=ones(size(dY));
>> quiver(T, Y, dT, dY);
```

The (unsatisfying) result is in Figure 1, below. The problem lies with the automatic scaling feature of quiver, and the fact that the vectors in the direction field above have vastly different lengths. To fix this problem, we can scale all the vectors to have unit length, by dividing each one by its length. Inserting the command

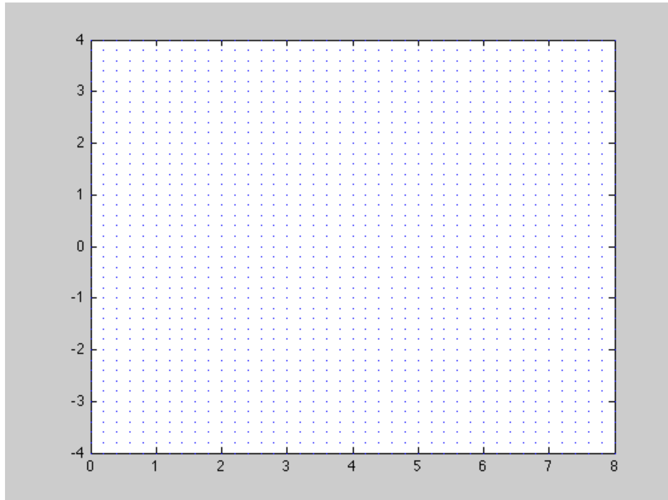

Figure 1: First try at direction field for $y' = \cos 2t - t/y$.

```
>> L=sqrt(1+dY.^2);
```

before the quiver command in the sequence above, and then changing the quiver command to

```
>> quiver(T, Y, dT./L, dY./L)
```

produces the output in Figure 2.

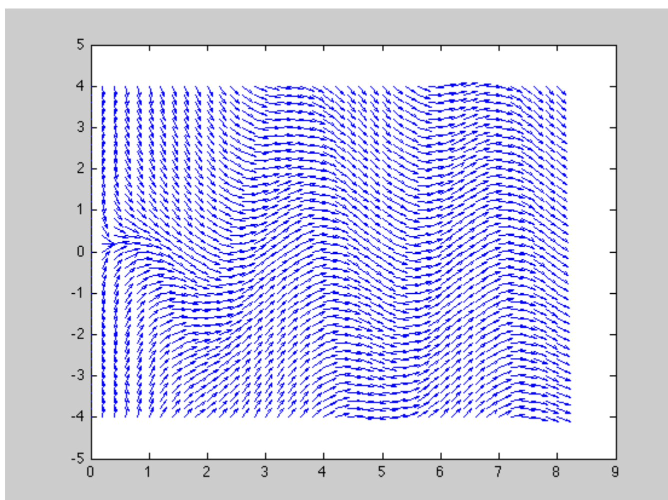

Figure 2: Second try at direction field for $y' = \cos 2t - t/y$.

This one looks like a direction field, but there are still things to fix. First of all, there is the unattractive white space around the direction field. This can be fixed with the command

```
>> axis tight
```

Second of all, the vectors are now overlapping in places. To fix this, we can add an optional argument to the quiver command that scales down the length of all the vectors. For example, the command

```
>> quiver(T, Y, dT./L, dY./L, 0.5), axis tight
```

produces the more pleasing direction field in Figure 3, where the vectors have been scaled to half their former length and the white space is gone.

Since quiver is a plotting command many (if not all) of the optional plotting arguments and commands can be used in conjunction with quiver. E.g., I added labels to the axes of the plot in Figure 3 with the command

```
>> xlabel 't', ylabel 'y';
```

and I added a title to the figure with the command

```
>> title 'Direction field for dy/dt=cos(2t)-y/t';
```

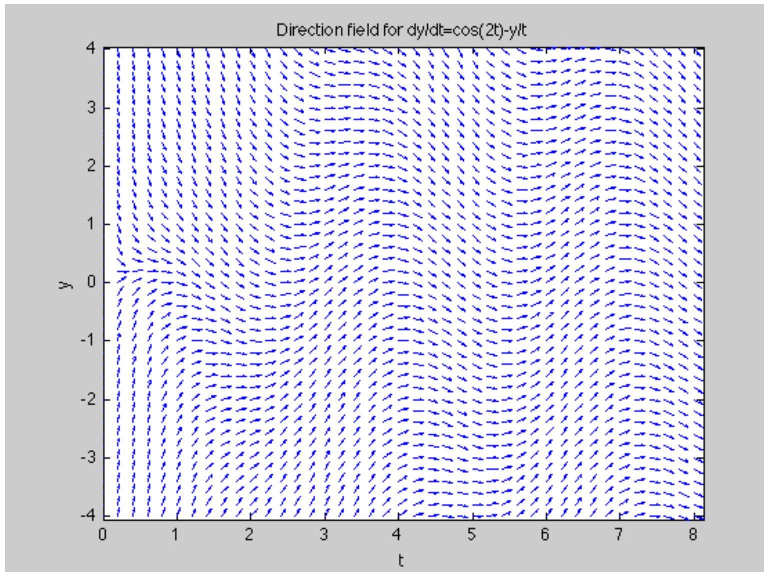


Figure 3: Third try at direction field for $y' = \cos 2t - t/y$.

The red direction field in Figure 4 was produced by adding the optional argument 'r' to the quiver command, i.e., typing

```
>> quiver(T, Y, dT./L, dY./L, 0.5,'r'), axis tight
```

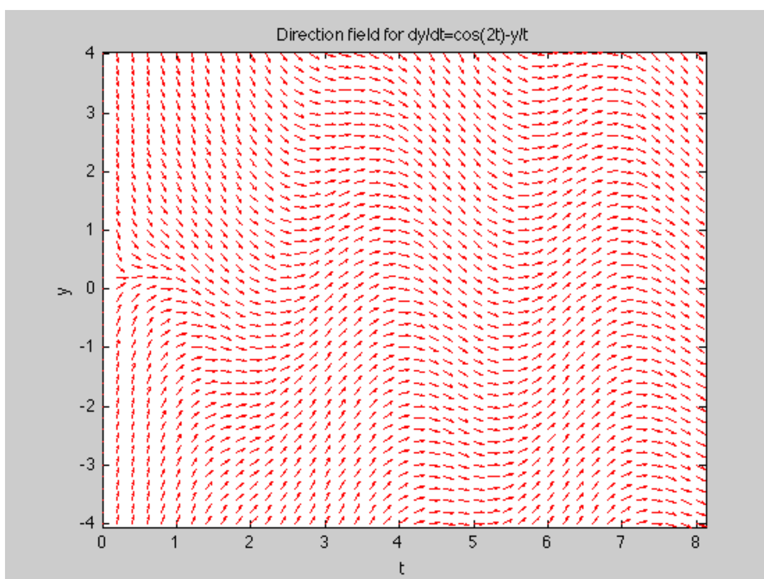


Figure 4: Red direction field for $y' = \cos 2t - t/y$.

## 2. Contour plots.

A *contour plot* is essentially the graph of an equation of the form $F(x,y) = C$. To graph such equations using MATLAB, we use the command **contour** together with **meshgrid**. The basic syntax is

```
>> [X Y] = meshgrid(minX:step:maxX, minY:step:maxY);
>> contour(X, Y, F(X,Y));
>> %% Remember: you have to type an actual function for F(X,Y).
```

This pair of commands will plot *several contours*, i.e., the result will be the graphs of several equations of the form F(X,Y)=C, for different values of C, which MATLAB chooses automatically, based on the grid and other considerations. E.g., the pair of commands

```
>> [X Y] = meshgrid(-5:0.2:5,-2:0.2:2);
>> contour(X,Y,Y.^4+X.^2);
```

produces the plots in Figure 5. If you want to choose the particular values of $C$ that MATLAB uses in the contour plot, you can do so with the optional argument $[\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_k}]$ in the **contour** command. This tells MATLAB to plot contours for the $k$ equations $F(X,Y) = C_j$, for $1 \le j \le k$. E.g., replacing the **contour** command above with

```
>> contour(X,Y,Y.^4+X.^2, [1,4,16]);
```

produces the contours in Figure 6.

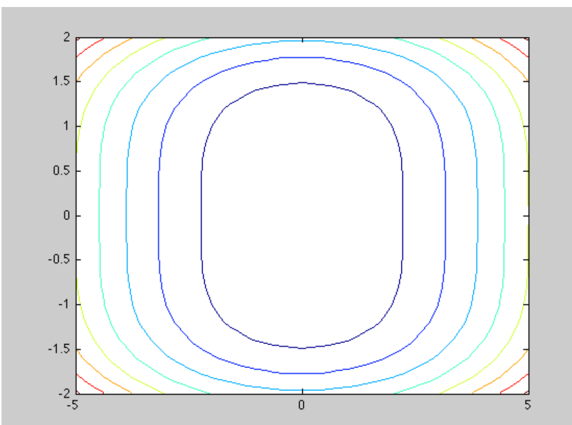

Figure 5: Contour plots for $y^4 + x^2 = C$.

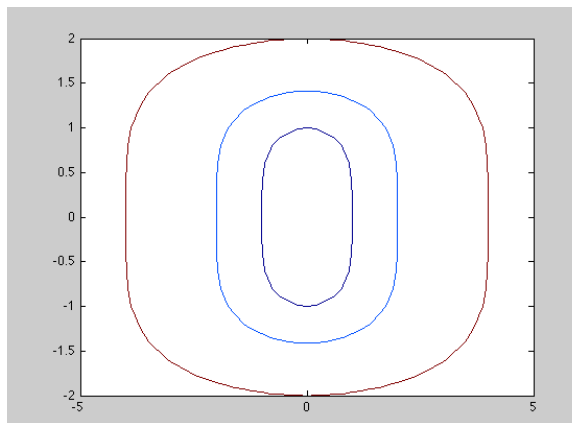

Figure 6: Contours for $C = 1$, 4 and 16.

## 3. Combining direction fields and contour plots.

Contour plots may be combined with direction fields to give a more complete picture of the solutions of differential equations. The contour plot is particularly useful in the case of separable differential equations, where implicit solutions are (relatively) easy to find, but explicit solutions are not.

Consider, for example the separable differential equation

$$\frac{dy}{dx} = \frac{4-x}{y^3+2}.$$

A direction field for this differential equation is given in Figure 7. While the direction field by itself does reveal certain patterns, it may not be completely clear what solutions look like. You should verify that the general solution of the differential equation above is given by $y^4 + 2x^2 - 16x + 8y = C$. Adding a contour plot of this equation to the direction field above, produces a much clearer picture of the nature of the solutions, as shown in Figure 8. The graph of such an equation is called an *integral curve* of the differential equation.

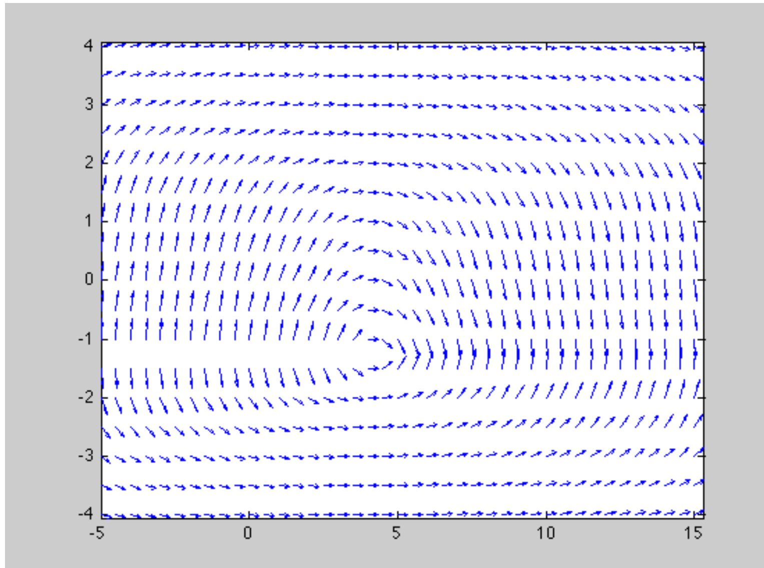


Figure 7: A direction field for $y' = (4 - x)/(y^3 + 2)$.

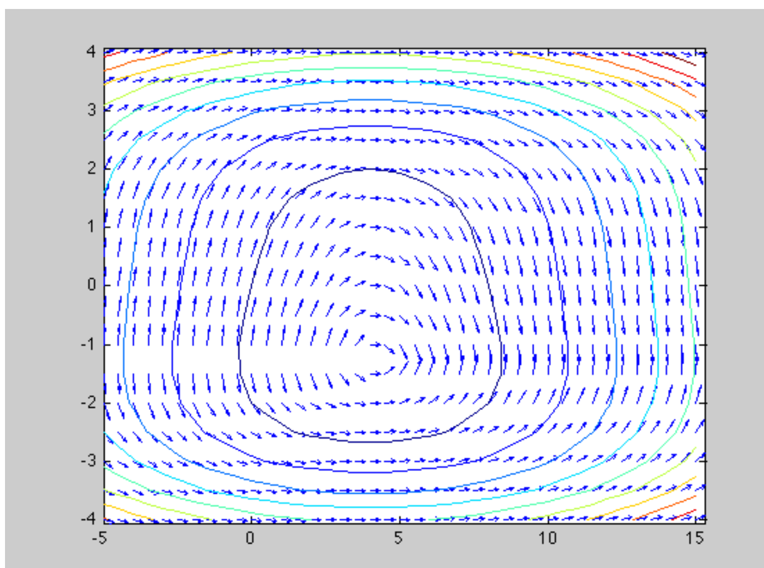


Figure 8: The direction field together with several *integral curves*.

Each curve corresponds to two solutions of the differential equation. The 'top' of the curve corresponds to an initial $y$-value that is greater than $-\sqrt[3]{2}$, and the 'bottom' of the curve corresponds to an initial $y$-value that is less than $-\sqrt[3]{2}$. Can you explain where the number $-\sqrt[3]{2}$ came from? Can you explain how this value is used to determine the *interval of definition* of a particular solution?