Graphical Analysis of Ordinary Differential Equations

by AhmetSacan

Direction Fields.

Consider the general ODE:

$$\frac{dy}{dx} = f(x, y)$$

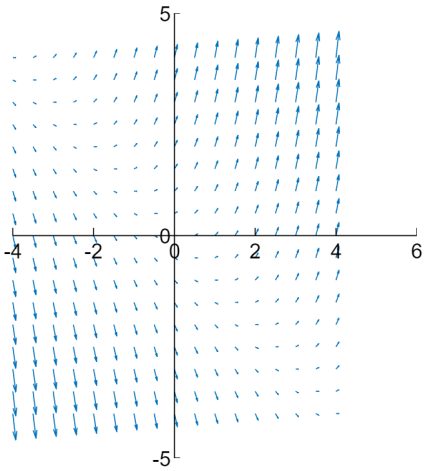
If f(x,y) is "complicated", there may not be an analytical solution for y(x). But we can certainly evaluate dy/dx to get a sense of the shape of solution of y(x).

As an example consider the following ODE:

$$y' = y + x$$

y' represents the slopes of the function y. Let's draw these slopes at a grid of x,y values.

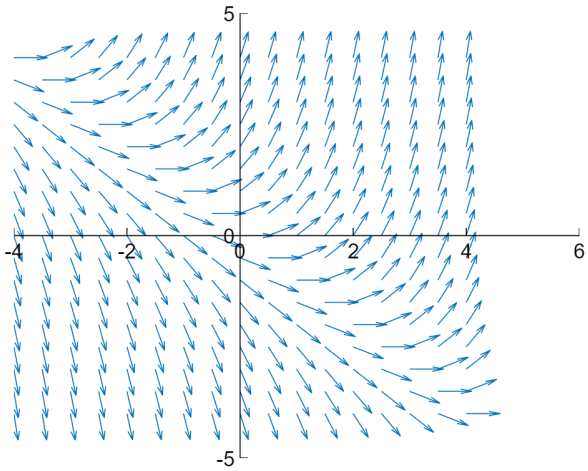
```
% meshgrid() is used to generate combinations of x/y values:
[xmesh,ymesh] = meshgrid(-4:.5:4, -4:.5:4);
dx = ones(size(xmesh)); %why is this??
dy = xmesh+ymesh;

% quiver() is the plotting function we use. It shows the direction and
% magnitude of the slope values y' at each x,y coordinate.
quiver(xmesh,ymesh,dx,dy); axis square;
fig_showaxisatorigin;
```



If you are only interested in the direction, but not the magnitude of the tangent line, you can normalize the vectors to make them each unit length.

```
lens=sqrt(dx.^2 + dy.^2);
dxu = dx./lens;
```

```
dyu = dy./lens;
quiver(xmesh,ymesh,dxu,dyu);
fig_showaxisatorigin;
```



Integral/Solution Curves

If you have an analytical solution for the differential equation, you can display examples of the solution curves on top of the direction fields.

The solution to y'=x+y can be found as follows

```
syms y(x)
dsolve([diff(y,x)==x+y(x)])
```

ans = $C_1 e^x - x - 1$

which gives: y = C*exp(x) - x -1. Writing this to put C on one side, we get:

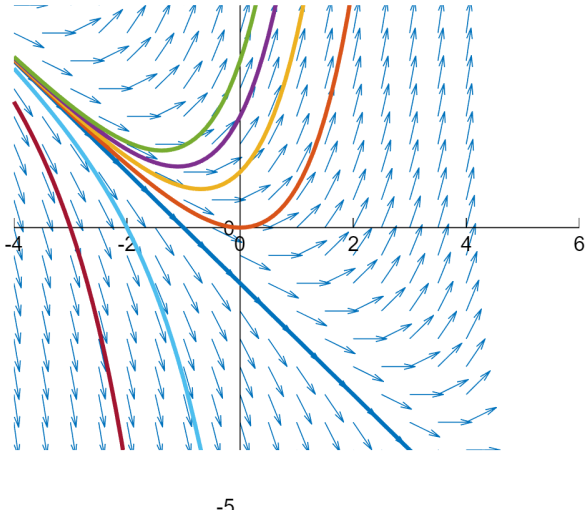
$$C = \frac{y + x + 1}{e^x}$$

```
% You can pick sample x,y points and calculate the corresponding C using
% the formula above. E.g., for x=0, y=1, we get C=2.
```

for several values of C, let's draw the integral curves:

```
xylims=[xlim; ylim];
hold on;
%I set up example C values so they cut x/y axes at integer values.
for C=[-exp(2) -2*exp(3) 0 1 2 3 4]
    x=xylims(1,1):.1:xylims(1,2);
    y = C*exp(x) - x - 1;
    plot(x,y,'Linewidth',2);
```

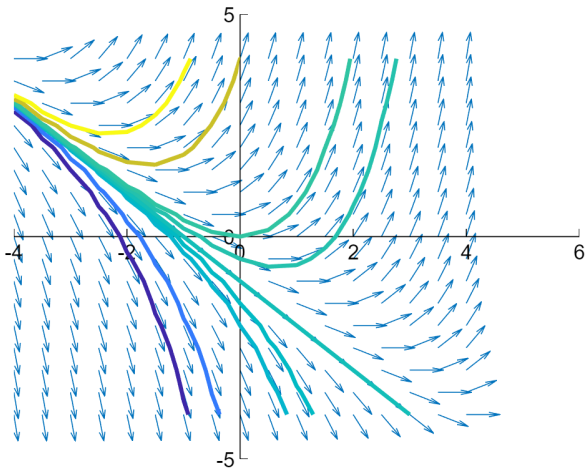
```
end
ylim([-4 4]);
hold off;
```



Contour Lines

Integral curves can also be shown using the contour() function. For contour(), we provide values of C at each mesh point and let matlab show the lines that correspond to specific values of C.

```
% Redraw the direction fields.
quiver(xmesh,ymesh,dxu,dyu);
fig_showaxisatorigin;

C=(ymesh+xmesh+1)./exp(xmesh);
hold on;
% if you don't provide contour values, it decides automatically.
%contour(xmesh,ymesh, C, 'LineWidth',2);
%contour(xmesh,ymesh, C, [-10 2], 'LineWidth',2);
contour(xmesh,ymesh, C, [-10 -5 -1 -0.5 0 0.5 1 5 10], 'LineWidth',2);
hold off;
```



Phase Diagrams

When two dependent variables depend on a third independent variable (typically the third variable is time), we typically show the phase diagram, where at each mesh point, the arrows indicate the direction of change in the two variables (i.e., where they would go next).

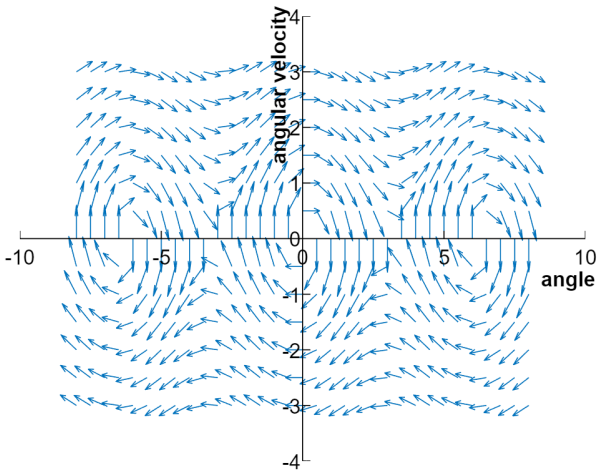
Consider the angle x(t) and angular velocity y(t) of a swinging pendulum of length L. To simplify, let's use a pendulum whose length is equal to the value of gravitation g (L=g). See: https://en.wikipedia.org/wiki/Pendulum_(mathematics)

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} = -\frac{g}{L} * sin(x) = -sin(x)$$

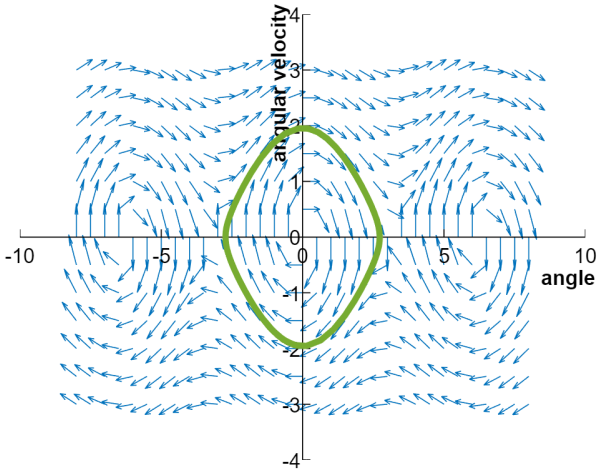
Let's show the phase plot of angle (x) vs. angular velocity (y)

```
[xmesh,ymesh] = meshgrid(-8:.5:8, -3:.5:3);
dx = ymesh;
dy = -sin(xmesh);
lens=sqrt(dx.^2 + dy.^2);
dyu = dy./lens;
dxu = dx./lens;
quiver(xmesh,ymesh,dxu,dyu);
xlabel angle; ylabel('angular velocity')
fig_showaxisatorigin;
```

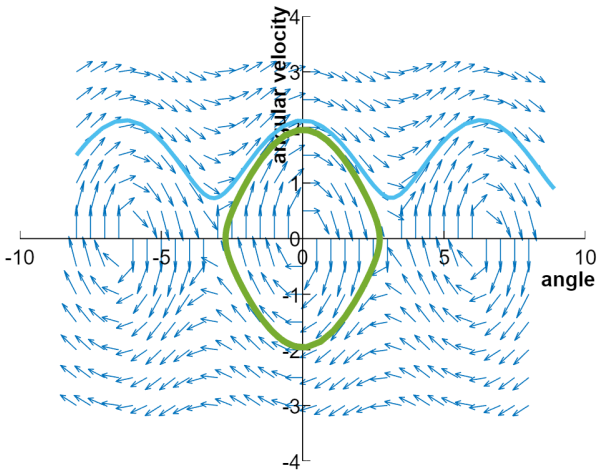


Let's generate solution curves and show it on top of the phase plot.

```
dxyfun=@(t,xy) [xy(2); -sin(xy(1))]; %function handle
[T,XY] = ode45(dxyfun, [0 20],[-2 1]);
hold on;
plot(XY(:,1),XY(:,2),'LineWidth',3);
hold off
```




```
%Let's try another initial condition.
[T,XY] = ode45(dxyfun, [0 12],[-8 1.5]);
hold on;
plot(XY(:,1),XY(:,2),'LineWidth',2);
hold off
```




```
% ALSO SEE phaseplot_animate.m
```