MATH 2A RECITATION 10/6/11

ALDEN WALKER

1. Remember

You are allowed to use a computer to draw direction fields, but you may not use a computer to do symbolic manipulation for you (like, symbolically solve differential equations).

2. Joining Points

Suppose that I give you n red points and n blue points in the plane. Is it possible to draw straight lines joining every blue point to a red point (pair up red and blue) so that none of the lines cross?

If you try to think up a solution to this problem, you will find that it is tricky. Here is a very clever argument. This sort of thing shows up a lot in math. First, there are lots of ways of pairing red and blue points with lines that *do* (or might) cross. In fact, there are $n!$ ways. This is large, but finite. Now think about the total length of all the lines in the pairing—because there are a finite number of pairings, there is some pairing with minimum total length. Call this pairing P, and suppose that P has lines that cross. It is easy to convince yourself by drawing a simple picture that you can reduce the total length of P by uncrossing the lines. This is a contradiction, however, because the length of P was supposed to be minimal! Therefore, P must not have any crossing lines. This proves the existence of a pairing with no crossings.

3. Slope Fields

If you know that $y' = f(x, y)$, you can just plot what y' is at many points in the plane to get an idea of the behavior of a solution to the differential equation. Clearly, you want to do this on a computer.

I am going to attempt to give all (or most) of my computer examples using Sage. Sage is a free, open source alternative to Mathematica, Matlab, etc. In some ways, it is not as polished as the non-free programs, but it is very nice in some other ways. One cool feature is the fact that its notebook interface runs using a web browser. The people who develop Sage have set up a website (`www.sagenb.org`) on which anyone can set up an account and use Sage from any computer with access to the internet. That is, no installing of anything or worrying about keyservers.

3.1. Slope Field Commands.

Go to your Sage notebook (or sagenb.org). You must first tell Sage that x and y are variables by entering the command:

```
var('x y')
```

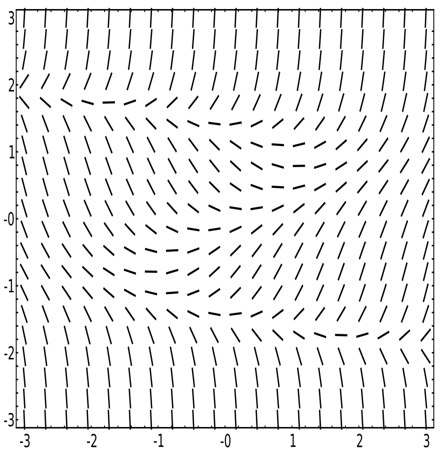
Press shift-enter to execute the command. Then, you simply type your function $f(x, y)$ (here I am using $f(x, y) = y^3 - 2y + x$), followed by the bounds you want for the plot, into the function `plot_slope_field` as in:

```
plot_slope_field(y^3 - 2*y + x, (x,-3, 3), (y,-3, 3))
```

The aspect ratio isn't quite right, since you probably want a square plot, so you can give it that graphics option using the command:

```
plot_slope_field(y^3 - 2*y + x, (x,-3, 3), (y,-3, 3)).show(aspect_ratio=1)
```

And bam, you have your plot (right click and save image to save the image — you can also run the command `plot_slope_field(y^3 - 2*y + x, (x,-3, 3), (y,-3, 3)).save("slopeField.eps")` to save as, for example, an eps file):



Sage slope field

.

3.2. **With Mathematica.** Here is a command to plot the slope field in Mathematica:
`VectorPlot[{1,y^3-2*y+x}/Norm[{1,y^3-2*y+x}], {x,-3,3},{y,-3,3}, VectorStyle -> Arrowheads[0]]`
Actually, I think Sage makes it easier, which is a little surprising. I hear it's also quite easy in Matlab.

4. Approximation of Solutions

4.1. **Euler's Method.** The basic idea:

(1) Start at your initial point, get dx/dt there by evaluating $f(x,t)$.
(2) Go some small distance h along a line with that slope
(3) Get the actual dx/dt there and go to step (2)

Repeat this until you reach the t-value t_1 for which you want to know $x(t)$. You have "approximated the value of $x(t_1)$ with Euler's method with a step size of h." The recurrence can be stated:

$$t_n = nh$$
$$x_{n+1} = x_n + hf(x_n, t_n)$$

4.1.1. *Example.* Let $dx/dt = x - t$, and $x(0) = 0$. Approximate the value of $x(1)$ using Euler's method with a step size of $h = 1/4$:
We just set the initial values $x_0 = 0$, $t_0 = 0$, so:
$t_1 = 1/4 \quad x_1 = 0 + (1/4)0 = 0$
$t_2 = 1/2 \quad x_2 = 0 + (1/4)(-1/4) = -1/16$
$t_3 = 3/4 \quad x_3 = -1/16 + (1/4)(-1/16 - 1/2) = -13/64$
$t_4 = 1 \quad x_4 = -13/64 + (1/4)(-13/64 - 3/4) = -113/256 \approx -0.44$
A step size of $h = 1/200$ gives $x(1) = -0.711517$. The actual answer is -0.7182818 (yes, that's $2 - e$), so that's not too bad.
Page 202 has some nice pictures.

4.2. **Picard's Method.** Using Picard's method, we get successive approximating *functions* for x, rather than approximations for specific values $x(t)$ as with Euler's method. The motivation for Picard's method is that if x is a solution to $x' = f(x,t)$, then

$$x(t) = x(0) + \int_{t_0}^{t} x'(\tilde{t})\, d\tilde{t} = x(0) + \int_{t_0}^{t} f(x(\tilde{t}), \tilde{t})\, d\tilde{t}$$

If the problem were "trivial," then f wouldn't depend on x, so let's just assume the dependence is very simple, setting $x_0(t) = x(0)$, to get the approximation:

$$x_1(t) = x_0(0) + \int_{t_0}^{t} f(x_0(\tilde{t}), \tilde{t})\, d\tilde{t}$$

We can then repeat this process using x_1 in place of x to get x_2, and so on. If f is $\mathcal{C}^1$, then there is some small interval on which Picard's successive approximations are guaranteed to agree in the limit with the true solution (see p.13 of the notes).

4.2.1. *Example.* As before, let $dx/dt = x - t$ and $x(0) = 0$. Then we approximate

$$x_1(t) = 0 + \int_0^t (0 - \tilde{t})\, d\tilde{t} = -\frac{t^2}{2}$$

and

$$x_2(t) = \int_0^t (-\frac{\tilde{t}^2}{2} - \tilde{t})\, d\tilde{t} = -\frac{t^3}{6} - \frac{t^2}{2}$$

We can see that the pattern will give

$$x_n(t) = -\sum_{i=2}^{n+1} \frac{t^i}{i!}$$

And if we let $n \to \infty$, it's clear that we will get $x_\infty(t) = -e^t + t + 1$. Actually, we have solved the differential equation, since we can check that $dx_\infty/dt = -e^t + 1 = x_\infty - t$, and $x_\infty(0) = 0$. Pretty cool.

5. Difference Equations

These are discrete analogs of autonomous differential equations. In general, we are interested in the long-term behavior of recursions like

$$x_{n+1} = f(x_n)$$

For example, $x_{n+1} = x_n^2 + x_n + 1$. A good way to qualitatively study equations like these is to draw a **cobweb** diagram, an example of which can be found on p.226.

Warning!: There are a couple problems on your homework about difference equations. One is simply to plot a cobweb diagram and describe some behavior. This isn't too hard. The other one is a proof, which you will probably find harder. Just like with the autonomous equations, your goal should be to draw a picture, formally show that your picture is correct (if you zoom in), and then explain (formally) why that yields what you want to prove.

6. Separable and Exact Equations

6.1. **Separable.** If you need to solve a differential equation of the form $\frac{dx}{dt} = \frac{g(t)}{h(x)}$, then you can write $h(x)dx = g(t)dt$. If you integrate this, you will obtain an (implicit) solution to the DE.
Really you obtain a function in x and t for which the level sets are solutions to the DE with different initial conditions. Notice that I've "multiplied by dt, which of course doesn't make any sense. However, see p.64 for the justification of this trick.

6.1.1. *Example.* Solve $\frac{dx}{dt} = \frac{2t}{3x^2-1}$ with $x(0) = 0$: We separate to get $(3x^2 - 1)dx = (2t)dt$, and integrating,

$$x^3 - x = t^2$$

In this case, when $t = 0$, we have $x = 0$, so $C = 0$. You can plot the implicit solution using Sage with the command (first make sure to run `var('x t')` to tell Sage that x and t are variables):[1]

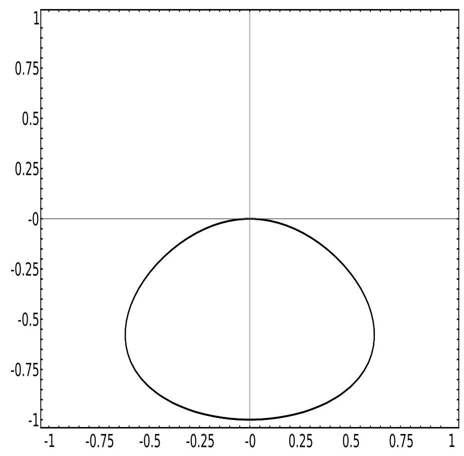
```
implicit_plot(x^3-x-t^2, (t, -1, 1), (x, -1, 1))
```

If you want lines for the axes, you can say:

```
implicit_plot(x^3-x-t^2, (t, -1, 1), (x, -1, 1)).show(gridlines=[[0],[0]])
```

Anyway, you get something like:

[1]In Mathematica, the command is `ContourPlot`



Note that the solution to the IVP is *n*ot the whole plot: since the graph doubles back, the solution is only defined between the vertical tangents. We can determine this algebraically, since if we look at the differential equation, we see that the solution will only be defined where $3x^2 - 1 \neq 0$, that is, since we know $x(0) = 0$ (so $3x^2 - 1 < 0$), we see we must have $-1/\sqrt{3} < x < 1/\sqrt{3}$, or rather, $-\sqrt{3^{-1/2} - 3^{-3/2}} < t < \sqrt{3^{-1/2} - 3^{-3/2}} \approx 0.62$.

6.2. **Exact Equations.** We can generalize from separable equations a little bit to **exact** equations. These are equations of the form:

$$M(x,t)dt + N(x,t)dx = 0$$

Subject to the constraint that there is a $\mathcal{C}^1$ function $E(x,t)$ such that $E_t = M$ and $E_x = N$. Separable equations are an example of this with M being a function only of t and N being a function only of x. Notice that if such an E exists, then when we implicitly differentiate something of the form $E = C$, we get $E_t + E_x \frac{dx}{dt}$, which is the equation that we started with. That is, if we find E, it gives us an implicit solution for x.

6.2.1. *Testing for Exactness.* If M and N are $\mathcal{C}^1$ and defined in a rectangle, and $M_x = N_t$, then the equation is exact (see p.19 of the notes).

6.2.2. *Using Exactness to Find a Solution.* Since $E_t = M$, we have

$$E(x,t) = \int_{t_0}^{t} M(x,s)\,ds + h(x)$$

We then differentiate with respect to x, and the fact that $E_x = N$ gives us

$$N(x,t) = \int_{t_0}^{t} M_x(x,s)\,ds + h'(x)$$

Which will allow us to find h, assuming we can antidifferentiate.

6.2.3. *Example.* We want to solve the equation:

$$(2x^2 + t)dt + (4xt + 1)dx = 0$$

We check that it's exact by first noting that it's $\mathcal{C}^1$ and defined everywhere (a rectangle), and that $\frac{\partial}{\partial x}(2x^2 + t) = 4x = \frac{\partial}{\partial t}(4xt + 1)$. Therefore it is exact, and we can hope to recover a solution E (a **first integral**). Following the method,

$$\begin{aligned} E(x,t) &= \int_{t_0}^{t} (2x^2 + s)\,ds + h(x) \\ &= 2x^2 t + t^2/2 + h(x) \end{aligned}$$

If you are distressed by the apparent disappearance of the t_0 term in the integral, have no fear, it is safely thought of as part of $h(x)$ since it is constant with respect to t.
Then we differentiate, since we know $E_x = N$, ie

$$4xt + 1 = N(x,t) = E_x = 4xt + h'(x)$$

So $h'(x) = 1$, so $h(x) = x + C$. Therefore our solution is $E(x,t) = 2x^2 t + t^2/2 + x = C$.
If we want to get a specific solution, we need to plug in given values to solve for C.