Math 231, Fri 12-Mar-2021 -- Fri 12-Mar-2021
Differential Equations and Linear Algebra
Spring 2020

Friday, March 12th 2021

Wk 6, Fr
Topic:: Existence and uniqueness wrapup
Read:: ODELA 2.2.1
HW::   WW Ch02Part2 due Mon 11 pm

Existence and uniqueness wrap-up
- IVP to solve: y' = y^2, y(0)=1
- Solve it outright, note the blowup
- Use Euler's method to solve it with h=0.3: steps right past the blowup
- Disturbing?

Revisit existence and uniqueness for y' = g(t,y) = a(t)y + f(t)
- Doesn't apply to problem above
- Gives assurance of no blowup in an interval

$A(t)$ = amt. of radioactive material at time t

$$A(t) = \boxed{Ce^{kt}}$$

At $t = 5730$ yrs, $\underline{A(5730) = \frac{1}{2}A(0)}$

$$A(0) = C$$

$$A(5730) = Ce^{5730k}$$

$$\cancel{C}e^{5730k} = \frac{1}{2}\cancel{C}$$

Ex.) $y' = y^2, \quad y(0) = 1$

$\frac{dy}{dt} = y^2$ after separating $\int y^{-2}\,dy = \int dt$

$$-y^{-1} = t + C$$

mult. by (-1)

$$y^{-1} = C - t$$

reciprocating

$$y = \frac{1}{C - t},$$ one-degree of freedom family of solns. to DE

IC: $y(0) = 1$;

$$1 = \frac{1}{C - 0} \Rightarrow C = 1$$

IVP has soln. $$y(t) = \frac{1}{1 - t}.$$ blows up at $t = 1$

If I applied Euler's method to this problem

Given $(t_0, y_0) = (0, 1)$

Choose h

Iterate $$y_{n+1} = y_n + h f(t_n, y_n)$$

Here, I'll take $h = 0.3$

$$f(t,y) = y^2$$

n	t_n	y_n
0	0	1
1	0.3	1.3
2	0.6	1.807
3	0.9	2.7866
4	1.2	5.116
5	1.5	12.698

$$y_1 = y_0 + h\,\underbrace{f(t_0, y_0)}$$
$$= 1 + (0.3)(1^2)$$
$$y_2 = y_1 + h\,f(t_1, y_1)$$
$$= 1.3 + (0.3)(1.3)^2$$
$$y_3 = 1.807 + (0.3)(1.807)^2$$
$$y_4 = 2.7866 + (0.3)(2.7866)^2$$
$$= 5.116$$

Existence & Uniqueness

If our problem is <u>linear</u>, then

$$y' = g(t,y), \qquad y(t_0) = y_0$$
$$= a(t)\,y + f(t)$$

So,

$$g(t,y) = a(t)\,y + f(t)$$

$$\frac{\partial g}{\partial y} = a(t)$$

and our criteria about sites where g, $\frac{\partial g}{\partial y}$ are continuous become criteria about where $a(t)$, $f(t)$ are continuous.

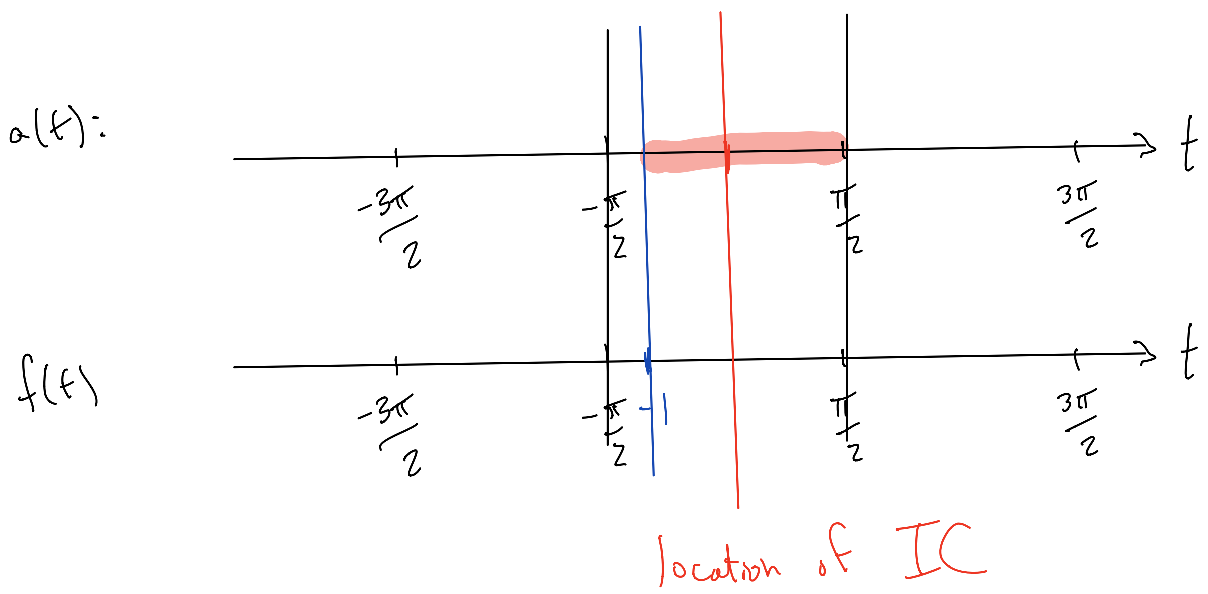
Thm 3 from 2 class periods ago takes this info account.

Applied to the problem

Ex. $$y' = (\tan t)\,y + \frac{1}{1+t}, \quad y(0) = 1$$

$a(t) = \tan t$ ~ discont. at $t = \ldots, -\frac{3\pi}{2}, \underbrace{-\frac{\pi}{2}, \frac{\pi}{2}}_{t=0 \text{ in between}}, \frac{3\pi}{2}, \ldots$

$f(t) = \dfrac{1}{1+t}$ ~ discont at $t = -1$



1st Order Systems of ODEs

We next consider the interaction between multiple dependent variables relying on the same independent variable, otherwise known as a **system of ODEs**. Generally, a system of equations has the same number of equations as unknowns, and this is true for the systems of ODEs we study as well. We start by looking at 1$^{\text{st}}$ order systems, ones in which each of the individual DEs is 1$^{\text{st}}$ order. Note that, while we will not state the theorems here, there are Existence and Uniqueness theorems for 1$^{\text{st}}$ order systems of DEs (both generally, and specifically to linear systems) which sound very much like the corresponding theorems for 1$^{\text{st}}$ order (scalar) DEs.

Example 1: Decoupled ODEs

$\vec{y} = \langle y_1, y_2, \ldots, y_n \rangle$, dep. vars.
are components

The simplest type of system of 1$^{\text{st}}$ order ODEs in n dependent variables is

$$\left.\begin{aligned} y_1' &= f_1(t, y_1),\\ y_2' &= f_2(t, y_2),\\ &\vdots\\ y_n' &= f_n(t, y_n). \end{aligned}\right\} \vec{f}$$

where no dependent variable y_i appears in any DE with a different dependent variable y_j. A specific instance with $n = 2$ might be

$$\begin{aligned} x' &= \sqrt{x},\\ y' &= t - y. \end{aligned}$$

Notice that the equation in x has no reference to y, while the equation in y has no reference to x. Such systems are said to be **decoupled**, which implies you may attack them separately and in either order, as if they were two completely separate problems.

To make an IVP out of this system, we would need ICs for both x and y—i.e., values $x(t_0)$, $y(t_0)$ specified (usually) at a common time t_0.

$$\begin{aligned} x' &= \sqrt{x}, & x(1) &= 1,\\ y' &= t - y, & y(1) &= 2. \end{aligned}$$

Solutions are

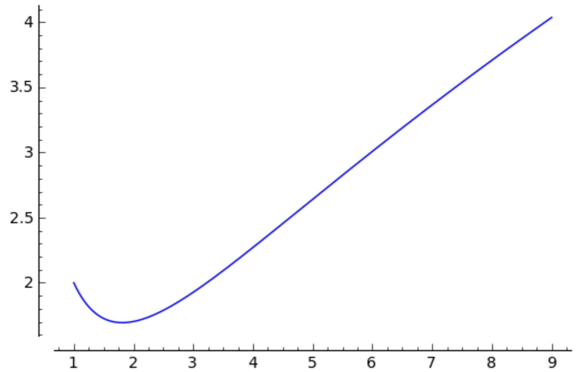
$$x(t) = \frac{1}{4}(t+1)^2, \qquad \text{and} \qquad y(t) = t - 1 + 2e^{1-t},$$

and are usually plotted as the single **vector function**

$$\mathbf{r}(t) = (x(t), y(t)) = \left(\frac{1}{4}(t+1)^2,\ t - 1 + 2e^{1-t}\right),$$

in the xy-plane, known as the **phase plane**. SAGE commands which achieve such a plot appear below.

```
var('t')
x(t) = (t+1)^2/4
y(t) = t - 1 + 2*exp(1-t)
parametric_plot( (x(t), y(t)), (t, 1, 5))
```



■

Example 2: Only slightly more complicated than a decoupled system

Consider the 1st order system

$$\begin{aligned} x' &= 2x, \qquad x(0) = 5, \\ y' &= xy, \qquad y(0) = 1. \end{aligned}$$

The system is nonlinear, and not truly decoupled. However, if we start by solving the first equation (the one in x, where y is absent) for $x(t)$, this system is no more difficult to solve than the decoupled system above.

In particular, having solved the first DE to get $x(t) = 5e^{2t}$, the second DE becomes linear, as we have

$$y' = xy = 5ye^{2t}, \quad \text{subject to} \quad y(0) = 1,$$

which has solution $y(t) = \exp\left(\frac{5}{2}e^{2t} - \frac{5}{2}\right)$.

■

Example 3: Fully coupled, linear system

The system of DEs

$$\begin{aligned} x' &= 2x + t^2 y, \qquad x(0) = 1, \\ y' &= x, \qquad y(0) = -1. \end{aligned}$$

is fully coupled (i.e., the expression for dx/dt relies on values of y and that for dy/dt relies on x), and is linear. [Note that the power to which the dependent variables x, y and their derivatives are raised is 1, the lack of *interactions* (terms like xx', xy, $x'y$, and the like), etc.]

■

Example 4: Affection between Romeo and Juliet (Strogatz, p. 138)

This is a disfunctional version of the story. Juliet grows cold toward Romeo when his feelings burn hot, but she becomes attracted to him if he doesn't seem to want her. Romeo, on the other hand, has feelings that lag behind Juliet's—If she shows interest, he starts warming up to her; but his interest wains when he notices it isn't reciprocated.

Let

$R(t)$ = Romeo's feelings for Juliet at time t.
$J(t)$ = Juliet's feelings for Romeo at time t.

When R, J are positive, this signifies feelings of love. When negative, this reflects hate. A simple model:

$$\left.\begin{aligned} dR/dt &= aJ \\ dJ/dt &= -bR \end{aligned}\right\} \quad \text{with } a, b \text{ positive constants.}$$

■

Example 5: Sharks caught in the Mediterranean Sea during WWI[1]

A tally of the types of fish brought by fishermen to the port of Fiume, Italy during the years 1914–1923 shows what percentage of the year's catch was classified as selachians (sharks, skates, rays, and other predators) which are undesirable for food:

1914	1915	1916	1917	1918	1919	1920	1921	1922	1923
11.9%	21.4%	22.1%	21.2%	36.4%	27.3%	16.0%	15.9%	14.8%	10.7%

Biologist Umberto D'Ancona was struck by the rise of this percentage during the years of WWI (1914-1919). D'Ancona's friend Vito Volterra proposed the following **nonlinear** model for the sizes $x(t)$ of prey and $y(t)$ of predator populations:

$$\begin{aligned} \frac{dx}{dt} &= ax - bxy, \\ \frac{dy}{dt} &= -cy + dxy, \end{aligned} \tag{1}$$

a, b, c, d > 0 constants

where a, b, c, d are positive contants. It is possible to find the solution curves (in the phase plane—that is, to find $\mathbf{r}(t) = (x(t), y(t))$) *analytically*, but we will not investigate how. Instead, we use the 4$^{\text{th}}$ order Runge-Kutta method to get a numerical solution.

[1]**Differential Equation Models, Vol. 1**, Braun, Martin, Courtney S. Coleman, Donald A. Drew, Eds., Springer-Verlag, 1978, p. 221 ff.

■

$$\vec{y} = \langle \underbrace{x, y}_{\text{dep. vars.}} \rangle \qquad \vec{y}\,' = \langle x', y' \rangle$$

$$\vec{y}\,' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} ax - bxy \\ -cy + dxy \end{bmatrix} = g(t, \vec{y})$$

1st order system of DEs in normal form

Numerical Solutions

My algorithm, implemented in SAGE, for the RK4 method on 1st order systems is

```
# vrkstep is not the full program, just a subroutine of it.
# See the main program vrk below.
def vrkstep(f, curr_y, t, h):
    y = vector(RR, curr_y)
    K1 = vector(RR, f(t, *y))
    nexty = y + h*K1/2
    K2 = vector(RR, f(t + h/2, *nexty))
    nexty = y + h*K2/2
    K3 = vector(RR, f(t + h/2, *nexty))
    nexty = y + h*K3
    K4 = vector(RR, f(t + h, *nexty))
    return (y + (K1 + 2*K2 + 2*K3 + K4) * h/6)

# This cell is the "program".  It must be evaluated, but will not actually
# do anything until the specifics of the problem are entered and the program
# is "called" (see the next cell).
def vrk(f, y0, t0, tFin, numSteps, keepCoords):
    h = (tFin - t0)/numSteps
    w = []
    t = t0
    y = vector(RR, y0)
    if (keepCoords.count(0) > 0):
        keepCoords.pop(keepCoords.index(0))
        keep_t = True
        nEntry = [n(t)]
        nEntry.extend([n(y0[j-1]) for j in keepCoords])
        w.append(tuple(nEntry))
    else:
        keep_t = False
        w.append(tuple([y0[j-1] for j in keepCoords]))
    keepIdxs = [j-1 for j in keepCoords]
    for i in range(1, numSteps+1):
        y = vrkstep(f, y, t, h)
        t = t0 + i*h
        if keep_t:
            nEntry = [n(t)]
            nEntry.extend([n(y[j]) for j in keepIdxs])
            w.append(tuple(nEntry))
        else:
            w.append(tuple([n(y[j]) for j in keepIdxs]))
    return w
```

Example 6: Numerical solution of Volterra Predator–Prey Eqns

Suppose $a = 1$, $b = 0.5$, $c = -0.75$ and $d = 0.25$ in the **predator–prey** equations (1). Use 4^{th} order RK to find the solution passing through the IC $x(0) = 1$, $y(0) = 2$.

Note that, while the lines of code for the algorithm above must be implemented, they do little until more code is introduced to initiate the specifics of the problem at hand and to call the vrk routine. In the case of our problem, we run commands such as these:

```
var('t x y')
f(t,x,y) = (x - x*y/2, -0.75*y + 0.25*x*y)
keepCoords = [1,2]
pts = vrk(f, [1,2], 0, 7, 200, keepCoords)
list_plot(pts, plotjoined=True)    # plots the soln (x(t),y(t)) in phase plane
```

■

Higher order systems

The study of 1^{st} order systems of DEs is far more important than it may appear at first glance. The reason is that both higher order scalar DEs and higher order systems of DEs can always be recast as 1^{st} order systems. Section 7.1 in the text discusses this conversion process at some length. We demonstrate the process here with a single example.

Example 7: The Two-Body Problem

Assume a "planet" (or other heavenly body) of mass M is fixed at point $(0, 0)$. There is a satellite of mass m orbiting this planet, whose position we label $(x, y) = (x(t), y(t))$. The gravitational force between planet and satellite as felt by the satellite has

$$\text{magnitude} = \frac{GMm}{x^2 + y^2}, \qquad \text{and direction vector} \qquad \text{direction} = \frac{(-x, -y)}{(x^2 + y^2)^{1/2}}.$$

By Newton's Law $F = ma$, we obtain the system of 2^{nd} order DEs

$$\begin{aligned} \frac{d^2x}{dt^2} &= \frac{-GMx}{(x^2 + y^2)^{3/2}}, \\ \frac{d^2y}{dt^2} &= \frac{-GMy}{(x^2 + y^2)^{3/2}}. \end{aligned} \tag{2}$$

To convert this to a 1^{st} order system, introduce new dependent variables $u = dx/dt$, $v = dy/dt$. The symbols d^2x/dt^2 and d^2y/dt^2 in system (2) may now be replaced by du/dt and dv/dt

respectively, yielding this 1st order system:

$$\begin{aligned}
\frac{dx}{dt} &= u, \\
\frac{du}{dt} &= \frac{-GMx}{(x^2+y^2)^{3/2}}, \\
\frac{dy}{dt} &= v, \\
\frac{dv}{dt} &= \frac{-GMy}{(x^2+y^2)^{3/2}}.
\end{aligned} \tag{3}$$

To solve this problem numerically using RK4, and graph the motion of the satellite in time, we carry out the following commands, in which we assume $G = M = 1$, as well as these ICs:

$$x(0) = 1, \quad u(0) = 0, \quad y(0) = 0, \quad v(0) = 0.75.$$

```
var('t x y')
f(t,x,dx,y,dy) = (dx, -x/(x^2+y^2)^(3/2), dy, -y/(x^2+y^2)^(3/2))
keepCoords = [1,3]
pts = vrk(f, [1,0,0,0.75], 0, 3, 100, keepCoords)
list_plot(pts, plotjoined=True)
```

■