Math 2250-004
Fri Jan 27

2.3 Improved velocity models: velocity-dependent drag forces

For particle motion along a line, with
$$\text{position } x(t) \text{ (or } y(t)),$$
$$\text{velocity } x'(t) = v(t), \text{ and}$$
$$\text{acceleration } x''(t) = v'(t) = a(t)$$

We have Newton's 2nd law
$$m\, v'(t) = F$$
where F is the net force.

- We're very familiar with constant force $F = m\, \alpha$, where α is a constant:
$$v'(t) = \alpha \quad \bullet$$
$$v(t) = \alpha\, t + v_0 \quad \bullet$$
$$x(t) = \frac{1}{2}\alpha\, t^2 + v_0\, t + x_0 . \quad \bullet$$

Examples we've seen a lot of:
- $\alpha = -g$ near the surface of the earth, if up is the positive direction, or $\alpha = g$ if down is the positive direction.
- boats or cars or "particles" subject to constant acceleration or deceleration.

New today !!! Combine a constant force with a velocity-dependent drag force, at the same time. The text calls this a "resistance" force:
$$m\, v'(t) = m\, \alpha + \boxed{F_R} \qquad \text{today.}$$
Empirically/mathematically the resistance forces F_R depend on velocity, in such a way that their magnitude is
$$|F_R| \approx k\,|v|^p , 1 \leq p \leq 2 .$$
$$F_R = -kv$$
- $p = 1$ (linear model, drag proportional to velocity):
$$m\, v'(t) = m\, \alpha - k\, v$$
   This linear model makes sense for "slow" velocities, as a linearization of the frictional force function, assuming that the force function is differentiable with respect to velocity...recall Taylor series for how the velocity resistance force might depend on velocity:
$$F_R(v) = F_R(0) + F_R'(0)\, v + \frac{1}{2!}F_R''(0)v^2 + \frac{1}{3!}F_R'''(0)v^3 + \dots$$
$F_R(0) = 0$ and for small enough v the higher order terms might be negligable compared to the linear term,
so
$$F_R'(0)v$$
$$F_R(v) \approx F_R'(0)\, v \approx -k\, v .$$
We write $-k\, v$ with $k > 0$, since the frictional force opposes the direction of motion, so sign opposite of the velocity's.
   http://en.wikipedia.org/wiki/Drag_(physics)#Very_low_Reynolds_numbers:_Stokes.27_drag
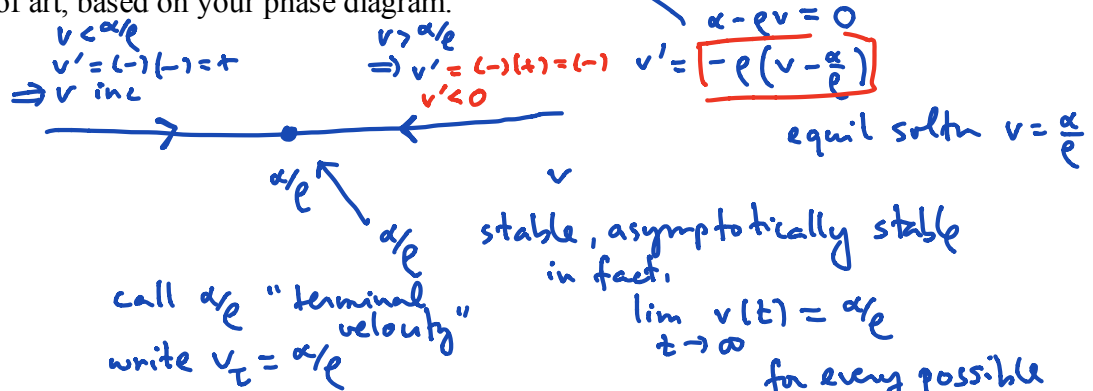
$$m v'(t) = m\alpha - \frac{kv}{m}$$

<u>Exercise 1a</u>: Rewrite the linear drag model as

$$v'(t) = \alpha - \rho v$$

$$\rho = \frac{k}{m}$$

where the $\rho = \dfrac{k}{m}$. Construct the phase diagram for v. Notice that $v(t)$ has exactly one constant

constant velocity

(equilibrium) solution, and find it. Its value is called the *terminal velocity*. Explain why *terminal velocity* is an appropriate term of art, based on your phase diagram.

$v < \alpha/\rho$
$v' = (-)(-) = +$
$\Rightarrow v$ inc

$v > \alpha/\rho$
$\Rightarrow v' = (-)(+) = (-)$
$v' < 0$

$\boxed{\alpha - \rho v = 0}$
$v' = \boxed{-\rho(v - \frac{\alpha}{\rho})}$

equil soltn $v = \frac{\alpha}{\rho}$

$\alpha/\rho \nwarrow$

α/ρ

call α/ρ "terminal velocity"

write $v_\tau = \alpha/\rho$

stable, asymptotically stable
in fact,
$$\lim_{t \to \infty} v(t) = \alpha/\rho$$
for every possible
$v(0) = v_0$

<u>1b)</u> Solve the IVP

$$v'(t) = \alpha - \rho v$$
$$v(0) = v_0$$

and verify your phase diagram analysis. (This is, once again, our friend the first order constant coefficient linear differential equation.)

linear, constant coefficients:

$v' + \rho v = \alpha$

$e^{\rho t}[v' + \rho v] = \alpha e^{\rho t}$

$\frac{d}{dt}(e^{\rho t} v) = \alpha e^{\rho t}$

$e^{\rho t} v = \int \alpha e^{\rho t} dt = \frac{\alpha}{\rho} e^{\rho t} + C$

$\div e^{\rho t}$

$v = \frac{\alpha}{\rho} + C e^{-\rho t}$

I.F. $= e^{\int \rho \, dt} = e^{\int \rho \, dt} = e^{\rho t}$

: IVP, $v(0) = v_0 = \frac{\alpha}{\rho} + C \Rightarrow C = v_0 - \frac{\alpha}{\rho}$

<u>1c)</u> integrate the velocity function above to find a formula for the position function $y(t)$.

$\boxed{v(t) = \frac{\alpha}{\rho} + (v_0 - \frac{\alpha}{\rho})e^{-\rho t}}$

$\boxed{v(t) = v_\tau + (v_0 - v_\tau)e^{-\rho t}}$

$y(t) = \int v(t) dt$

$y(t) = v_\tau t + \dfrac{(v_0 - v_\tau)e^{-\rho t}}{-\rho} + \boxed{C}$

@ $t=0$: $y_0 = 0 - \dfrac{(v_0 - v_\tau)}{\rho} + C \Rightarrow C = \boxed{y_0 + \dfrac{(v_0 - v_\tau)}{\rho}}$

consistent with phase portrait.

$y(t) = \underbrace{y_0}_{\substack{\text{initial} \\ \text{value}}} + \underbrace{v_\tau t}_{\substack{\text{moving} \\ \text{with} \\ \text{constant velocity } v_\tau}} + \underbrace{\dfrac{v_0 - v_\tau}{\rho}(1 - e^{-\rho t})}_{\substack{\text{correction term in case} \\ v_0 \neq v_\tau}}$

- $p = 2$, for the power in the resistance force. This can be an appropriate model for velocities which are not "near" zero....described in terms of "Reynolds number". Accounting for the fact that the resistance opposes direction of motion we get

F_R

$$m v'(t) = m \alpha - k v^2 \quad \text{if } v > 0 \quad, \quad F_R < 0$$
$$m v'(t) = m \alpha + k v^2 \quad \text{if } v < 0. \quad F_R > 0$$

http://en.wikipedia.org/wiki/Drag_(physics)#Drag_at_high_velocity

Exercise 2) Once again letting $\rho = \dfrac{k}{m}$ we can rewrite the DE's as

$$v'(t) = \alpha - \rho v^2 \quad \text{if } v > 0$$
$$v'(t) = \alpha + \rho v^2 \quad \text{if } v < 0.$$

2a) Consider the case in which $\boxed{\alpha = -g}$ so we are considering vertical motion, with up being the positive direction. Draw the phase diagrams. Note that each diagram contains a half line of v-values. Make conclusions about velocity behavior in case $v_0 > 0$ and $v_0 \leq 0$. Is there a terminal velocity?

$v' < 0$

$v > 0$ $\dfrac{dv}{dt} = -g - \rho v^2 = -(g + \rho v^2)$

$\displaystyle\int \dfrac{dv}{g + \rho v^2} = \int -dt$

$v' > 0$ $v' = (+)(-) < 0$

o ←———

u-sub
then arctan

$-\sqrt{g/\rho}$ →•← o "v_T"
asymptotically stable

$v < 0 :$ $\dfrac{dv}{dt} = -g + \rho v^2 = \rho\left(v^2 - \dfrac{g}{\rho}\right)$ equil soltn: $v^2 = \dfrac{g}{\rho}$

$v = \pm \sqrt{g/\rho}$

2b) Set up the two separable differential equation IVPs for the cases above, so that you will be able to complete finding the solution formulas (in your homework)....Of course, once you find the velocity function you'll still need to integrate that, if you want to find the position function!

$$\dfrac{dv}{v^2 - \frac{g}{\rho}} = \rho \, dt$$

$$\dfrac{dv}{(v - \sqrt{g/\rho})(v + \sqrt{g/\rho})} = \rho \, dt$$

partial fractions!

arctanh ☺

Application: We consider the bow and deadbolt example from the text, page 102-104. It's shot vertically into the air (watch out below!), with an initial velocity of 49 $\frac{m}{s}$. In the no-drag case, this could just be the vertical component of a deadbolt shot at an angle. With drag, one would need to study a more complicated system of DE's for the horizontal and vertical motions, if you didn't shoot the bolt straight up.

"5g" numerically

Exercise 3: First consider the case of no drag, so the governing equations are

$$v'(t) = -g \approx -9.8 \; \frac{m}{s^2}$$

$$v(t) = -g\,t + v_0 = -g\,t + 5\,g \;\; = g(-t+5)$$

$$x(t) = -\frac{1}{2}g\,t^2 + v_0\,t + x_0 = -\frac{1}{2}g\,t^2 + 5\,g\,t. \;\; = -\tfrac{1}{2}g(t^2 - 10t) = -\tfrac{1}{2}gt(t-10)$$

Find when $v = 0$ and deduce how long the object rises, how long it falls, and its maximum height.

5 sec.   5 sec.   x(5)
                                ||
                                122.5 meters

Maple check:

```
> restart :
  Digits := 5 :
```

```
> g := 9.8;
  v0 := 49.0;
  v1 := t → -g·t + v0;
  y1 := t → -1/2 ·g·t² + v0·t;
```

$$g := 9.8$$
$$v0 := 49.0$$
$$v1 := t \rightarrow -g\,t + v0$$
$$y1 := t \rightarrow -\frac{1}{2}\,g\,t^2 + v0\,t \tag{7}$$

Tuesday start here.

$$\text{Recall:} \quad mv' = -mg - kv$$
$$\div m \quad v' = -g - \rho v = -\rho\left(v + \frac{g}{\rho}\right); \quad \boxed{v_\tau = -\frac{g}{\rho}}$$

Exercise 4: Now consider the linear drag model for the same deadbolt, with the same initial velocity of $5\,g = 49\ \frac{m}{s}$. We'll assume that our deadbolt has a measured terminal velocity of $v_\tau = -245\ \frac{m}{s} = -25\,g$,

so $|v_\tau| = 25\,g = \frac{g}{\rho} \Rightarrow \boxed{\rho = .04}$ (convenient). So, from our earlier work:

$$|v_\tau| = \frac{g}{\rho}$$
$$25\,g = \frac{g}{\rho}$$
$$\rho = \frac{1}{25}$$

$$* \quad v = v_\tau + \left(v_0 - v_\tau\right)e^{-\rho t}$$

$$y = y_0 + t\,v_\tau + \frac{\left(v_0 - v_\tau\right)\left(1 - e^{-\rho t}\right)}{\rho}$$

So,

$$* \quad v = -\frac{g}{\rho} + \left(v_0 + \frac{g}{\rho}\right)e^{-\rho t} = -245 + 294\,e^{-.04\,t}.$$

$$** \quad y = 0 - 245\,t + \frac{294}{.04}\left(1 - e^{-.04\,t}\right).$$

set v(t)=0 solve for t

$y(t)$ where $v(t_1) = 0$.

When does the object reach its maximum height, what is this height, and how long does the object fall? Compare to the no-drag case with the same initial velocity, in Exercise 3.

$$-245 + 294\,e^{-.04\,t} = 0 \ ...$$

solve $y(t) = 0$
for t, say $t = t_2$
object fell for
$t_2 - t_1$ sec.

Maple check, and then work:

```
>
> with(DEtools):
> g := 9.8; ρ := .04; v0 := 49;
```

$$g := 9.8$$
$$\rho := 0.04$$
$$v0 := 49 \tag{8}$$

```
> dsolve( {v'(t) = -g - ρ·v(t), v(0) = v0}, v(t) );
```

$$v(t) = -245 + 294\,e^{-\frac{1}{25}t} \quad * \tag{9}$$

```
> v2 := t → -245.0 + 294 e^(-1/25 t);
```

$$v2 := t \to -245.0 + 294\,e^{-\frac{1}{25}t} \tag{10}$$

```
> solve(v2(t) = 0, t);
```

$$\boxed{4.5580} \tag{11}$$

how long deadbolt is going up → w/o drag it went up for 5 sec. makes sense

```
> y2 := t → y0 + ∫₀ᵗ -g/ρ + e^(-ρ s) (v0 + g/ρ) ds;
```

$$y2 := t \to y0 + \int_0^t \left(-\frac{g}{\rho} + e^{-\rho s}\left(v0 + \frac{g}{\rho}\right)\right)ds \tag{12}$$

> $v2(t)$;
  $\dfrac{294}{.04}$;
  $y0 := 0$;
  $y2(t)$;

$$-245.0 + 294\, e^{-\frac{1}{25} t}$$

$$7350.0$$

$$y0 := 0$$

✗✗ $\quad 7350. - 245.\, t - 7350.\, e^{-0.040000\, t}$ *(handwritten: t is in 2 very diffret places!)* *(handwritten: y(t) = 0)* **(13)**

> $solve(v2(t) = 0, t)$;
  $solve(y2(t) = 0, t)$;
  $y2(4.558)$;

(handwritten: $t_1 =$) $\quad 4.5580$

(handwritten: $t_2 =$) $9.4110, 0.$ *(handwritten: when deadbolt lands (vs. 10 sec w/o drag))*

(handwritten: falling for) $\quad 108.28$ *(handwritten: max height (compared to 122.5 m w/o drag --- makes sense))* **(14)**

(handwritten: $t_2 - t_1 = 9.41 \; -4.56$)

(handwritten: 4.85 sec. | longer to fall down)

picture:

> $with(plots)$:
  $plot1 := plot(y1(t), t = 0..10, color = green)$:
  $plot2 := plot(y2(t), t = 0..9.4110, color = blue)$:
  $display(\{plot1, plot2\}, title = $ `comparison of linear drag vs no drag models`$)$;

comparison of linear drag vs no drag models

(handwritten annotations on plot: no drag; a little linear drag.)
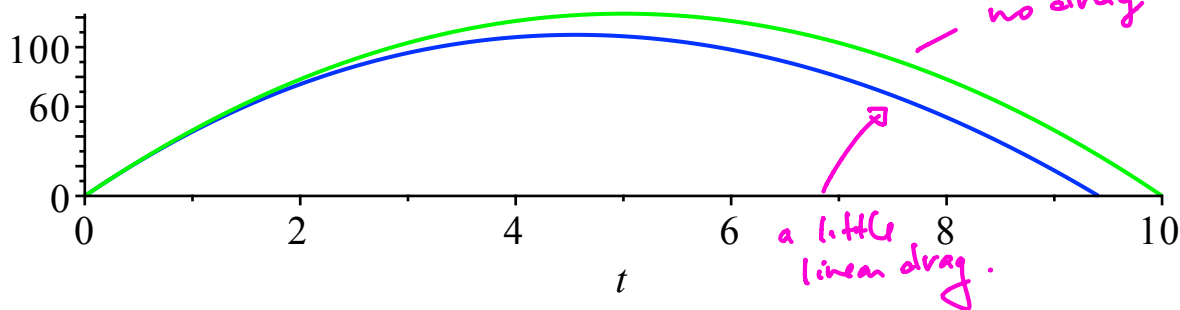
Math 2250-004
Week 4 notes, January 30-February 3. Sections 2.3, 2.4-2.6, 3.1

*Tuesday: finish §2.3 example.
Then*

Mon Jan 30: Use notes from last Friday to discuss improved velocity models, section 2.3

Tues Jan 31 and Wed Feb 1: Sections 2.4-2.6 numerical methods for differential equations.

Wednesday
• finish Maple notes
• show Matlab set-up (lab)
• quiz.

You may wish to download this file in Maple from our class homework or lecture page, or by directly opening the URL

http://www.math.utah.edu/~korevaar/2250spring17/week4.mw

from Maple. It contains discussion and Maple commands which may help you with your Maple/Matlab/Mathematica work later this week, in addition to our in-class discussions.

    In these notes we will study numerical methods for approximating solutions to first order differential equations. Later in the course we will see how higher order differential equations can be converted into first order systems of differential equations.   It turns out that there is a natural way to generalize what we do now in the context of a single first order differential equations, to systems of first order differential equations. So understanding this material will be an important step in understanding numerical solutions to higher order differential equations and to systems of differential equations later on.

....

If you decide to use these notes in the Math Department Computer lab (and for printing out class notes for free), Math Department login for students is as follows:

login name: Suppose your name is Public, John Q. Then your login name is
                              c-pcjq
(If several UU math students have the same initials as you, then you may actually be signed up as one of c-pcjq1, c-pcjq2, c-pcjq3 or c-pcjq4.)

password: If your UID is u***4986 then your password is
                              pcjq4986
(unless you've changed it). Even if you had to add a number to your login name, your password will only have the four letters part of your login name.

To open this document from Maple, go to our Math 2280 lecture page, then to the link above. Use the File/Open URL option inside Maple, and copy and paste the URL into the dialog box.

save a copy of this file to your home directory (using whatever name seems appropriate).
....

    In this handout we will study numerical methods for approximating solutions to first order differential equations. Later in the course we will see how higher order differential equations can be converted into first order systems of differential equations.   It turns out that there is a natural way to generalize what we do now in the context of a single first order differential equations, to systems of first order differential equations. So understanding this material will be an important step in understanding numerical solutions to higher order differential equations and to systems of differential equations.
    We will be working through material from sections 2.4-2.6 of the text.
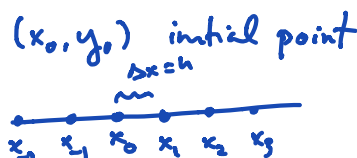
Euler's Method:

The most basic method of approximating solutions to differential equations is called Euler's method, after the 1700's mathematician who first formulated it. Consider the initial value problem

$$\frac{dy}{dx} = f(x,y) \quad \text{— slope function.}$$

$$y(x_0) = y_0.$$

We make repeated use of the familiar approximation

(x_0, y_0) initial point
$\Delta x = h$

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}$$

with fixed "step size" $\Delta x := h$ at our discretion (the smaller the better, probably, if we want to be accurate). We consider approximating the graph of the solution to the IVP. Begin at the initial point (x_0, y_0). Let

$$x_1 = x_0 + h$$

To get the Euler approximation y_1 to the exact value $y(x_1)$ use the rate of change $f(x_0, y_0)$ at your initial point (x_0, y_0), i.e. $\Delta y \approx f(x_0, y_0)\Delta x$, so

(x_0, y_0)

$\frac{\Delta y}{\Delta x} \approx \frac{dy}{dx} = f(x,y)$ $y_1 = y_0 + f(x_0, y_0)h.$

(x_1, y_1)
$\parallel \quad \searrow$

In general, if we've approximated (x_j, y_j) we set

$x_0 + h \quad y_0 + \Delta y$

$$x_{j+1} = x_j + h$$
$$y_{j+1} = y_j + f(x_j, y_j)h.$$

$y_1 = y_0 + f(x_0, y_0)\frac{\Delta x}{h}$

We could also approximate in the $-x$ direction from the initial point (x_0, y_0), by defining e.g.

$$x_{-1} = x_0 - h$$
$$y_{-1} = y_0 - hf(x_0, y_0)$$

and more generally via

$$x_{-j-1} = x_{-j} - h$$
$$y_{-j-1} = y_{-j} - hf(x_{-j}, y_{-j})$$

...etc.

.

Below is a graphical representation of the Euler method, illustrated for the IVP

IVP $\begin{cases} y'(x) = \boxed{1 - 3x + y} \quad f(x,y) \\ y(0) = 0 \end{cases}$
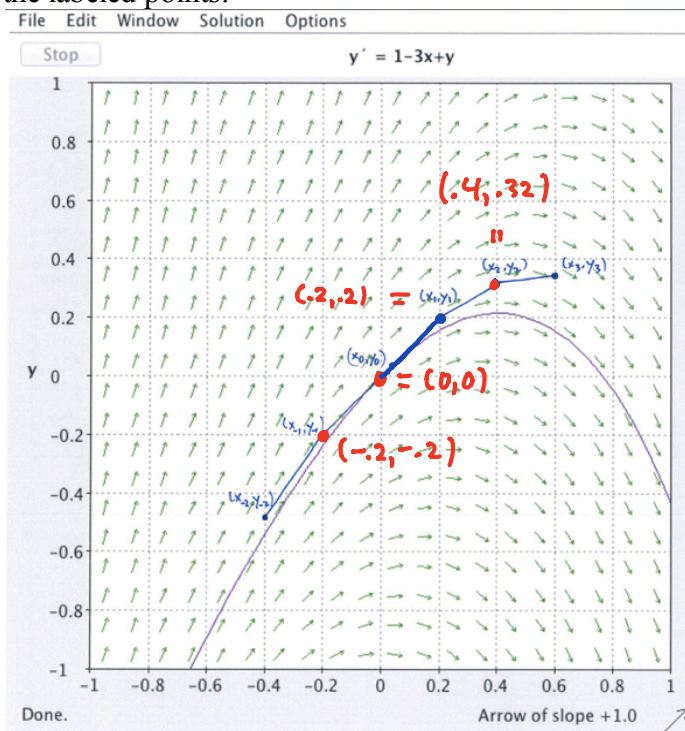
Δx

$(x_0, y_0) = (0, 0)$

with step size $h = 0.2$.

Exercise 1 Find the numerical values for several of the labeled points.

$x_0 = 0$ $y_0 = 0$ $f(x_0, y_0) = 1$

$x_1 = .2$ $y_1 = y_0 + 1(.2)$ $f(.2, .2)$
$ = 0 + .2 = 1 - .6 + .2$
$ = .2 = .6$

$x_2 = .4$ $y_2 = y_1 + f(x_1, y_1) \cdot (.2)$
$ = .2 + (.6)(.2)$
$ = .32$

$x_{-1} = -.2$ $y_{-1} = y_0 - .2 (f(x_0, y_0))$
$\phantom{x_{-1} = -.2 \quad} = 0 - .2 f(0,0)$
$\phantom{x_{-1} = -.2 \quad} = -.2$

File   Edit   Window   Solution   Options

Stop $y' = 1-3x+y$

Done.   Arrow of slope +1.0

As a running example in the remainder of these notes we will use one of our favorite IVP's from the time of Calculus, namely the initial value problem for $y(x)$,

$$y'(x) = \boxed{y = f(x,y)} \qquad y(x) = e^x$$
$$y(0) = 1.$$
$$(x_0, y_0) = (0, 1)$$

We know that $y = e^x$ is the solution.

Exercise 2 Work out by hand the approximate solution to the IVP above on the interval $[0, 1]$, with $n = 5$ subdivisions and $h = 0.2$, using Euler's method. This table might help organize the arithmetic. In this differential equation the slope function is $f(x, y) = y$ so is especially easy to compute.

step i	x_i	y_i	$k = f(x_i, y_i)$ $= y_i$	$\Delta x = h$	$\Delta y = h \cdot k$ $.2 \cdot 1$	$x_i + \Delta x$ $0 + .2$	$y_i + \Delta y$ $1 + .2$
0	0	1	1	0.2	0.2	0.2	1.2
1	0.2	1.2	1.2	.2	$(1.2)(.2) =$ .24	.4	$1.2 + (.2)(1.2) = (1.2)(1+.2) = (1.2)^2$ 1.44 1.44 .288
2	.4	1.44	1.44 $(1.2)^2$	.2	$(1.2)(.2)$ .288	.6	$(1.2)^2 + (1.2).2 = (1.2)^2(1.2)$ 1.728 $= (1.2)^3$ 1.728
3						.8	$(1.2)^4$
4						1	$(1.2)^5 = 2.488$
5	1	$(1.2)^5$					

Euler Table $\underset{\uparrow\uparrow}{2.488}$

Your work should be consistent with the picture below.
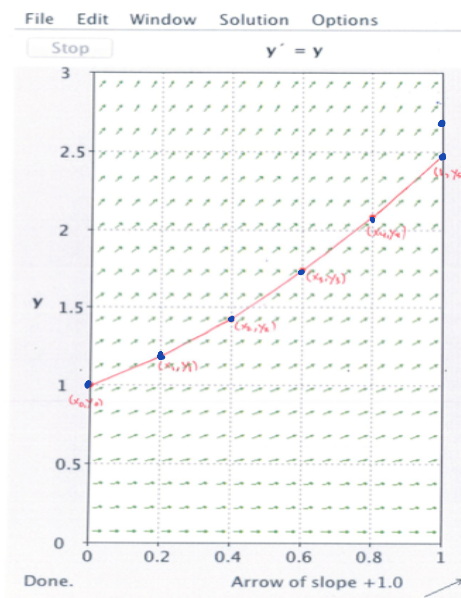
Here is the automated computation for the previous exercise, and a graph comparing the approximate solution points to the actual solution graph:

Exercise 3 Enter the following commands (by putting your cursor in the command fields and hitting enter). Discuss what each command is doing, and ask for any needed clarification. The first collection of commands is initializing the data. The second set is running the Euler loop. If you forget to initialize the data first, you might run into trouble trying to run the second set. No matter what is written in the document, commands are not executed until the cursor is put into a command field, and the <enter> or <return> key is pressed.

>

Initialize:

> $restart$: #clear all memory, if you wish

> $Digits := 6$: #use floating point arithmetic with 6 digits. could use more if desired

> $unassign(\,`x`,\,`y`);$ # in case you used the letters elsewhere

> $f := (x,y) \rightarrow y;$ # slope field function for the DE $y'(x)=y$
> # change for different DE's!!!

$$f := (x,y) \rightarrow y \tag{1}$$

> $x[0] := 0; y[0] := 1;$ #initial point
> $h := 0.2; n := 5;$ #step size and number of steps

$$x_0 := 0$$

$$y_0 := 1$$

$$h := 0.2$$

$$n := 5 \tag{2}$$

> $exactsol := x \rightarrow e^x;$ #exact solution for the IVP $y'=y$, $y(0)=1$
> #change (or omit) for different IVP's!!!

$$exactsol := x \rightarrow e^x \tag{3}$$

>

Euler Loop

> **for** i **from** 0 **to** n **do** #this is an iteration loop, with index "i" running from 0 to n
> $print(i, x[i], y[i], exactsol(x[i]));$
>   #print iteration step, current x,y, values, and exact solution value
> $k := f(x[i], y[i]);$ #current slope function value
> $x[i+1] := x[i] + h;$
> $y[i+1] := y[i] + h \cdot k;$
> **end do**: #how to end a do loop in Maple

$$0, 0, 1, 1$$
$$1, 0.2, 1.2, 1.22140$$
$$2, 0.4, 1.44, 1.49182$$
$$3, 0.6, 1.728, 1.82212$$
$$4, 0.8, 2.0736, 2.22554$$

↑ ↑ ↑ ↑x_i

i x_i y_i e

(4)

5, 1.0, 2.48832, 2.71828 *(circled: 2.48832, 2.71828)*

what we got by hand!

(4)

Plot results:

> with (plots) :
>
> *Eulerapprox* := *pointplot* ({ *seq* ([*x*[*i*], *y*[*i*]], *i* = 0 ..*n*) }) : #*approximate soln points*
> *exactsolgraph* := *plot* (*exactsol* (*t*), *t* = 0 ..1, `color` = `black`) : #*exact soln graph*
>   #*used t because x has been defined to be something else*
> *display* ({ *Eulerapprox, exactsolgraph* }, *title* = `approximate and exact solution graphs`);

approximate and exact solution graphs
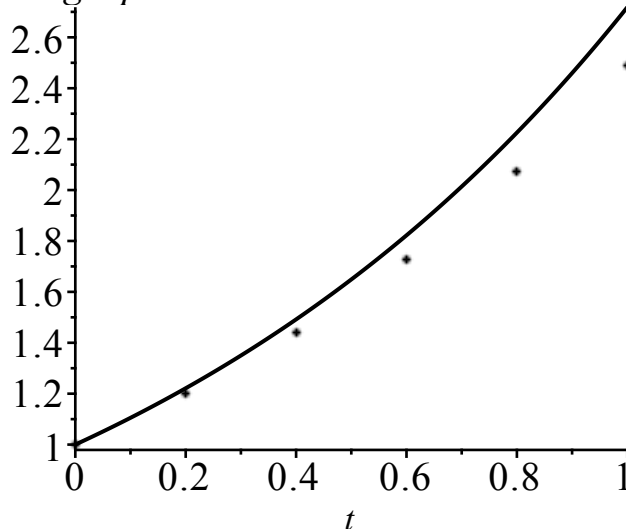
Exercise 4: Why are your approximations too small in this case, compared to the exact solution?

because we used constant slope on each subinterval (from left endpoints, but slope function "y" gets larger as y gets larger, so we were underestimating actual slopes

It should be that as your step size h gets smaller, your approximations to the actual solution get better. This is true if your computer can do exact math (which it can't), but in practice you don't want to make the computer do too many computations because of problems with round-off error and computation time, so for example, choosing $h = .0000001$ would not be practical. But, trying $h = 0.01$ in our previous initial value problem should be instructive.

If we change the n-value to 100 and keep the other data the same we can rerun our experiment, copying, pasting, modifying previous work.

Initialize
```
> restart : #clear all memory, if you wish
> unassign(`x`, `y`); # in case you used the letters elsewhere
> Digits := 6 :
> f := (x, y) →y; # slope field function for the DE  y'(x)=y
                  # change for different DE's!!!
```
$$f := (x, y) \to y \tag{5}$$

```
> x[0] := 0; y[0] := 1; #initial point
  h := 0.01; n := 100;   #step size and number of steps
```
$$x_0 := 0$$
$$y_0 := 1$$
$$h := 0.01$$
$$n := 100 \tag{6}$$

```
> exactsol := x→e^x; #exact solution for the IVP  y'=y, y(0)=1
                     #change (or omit) for different IVP's!!!
```
$$exactsol := x \to e^x \tag{7}$$
```
>
```

Euler Loop (modified using an "if then" conditional clause to only print every 10 steps)

```
> for i from 0 to n do  #this is an iteration loop, with index "i" running from 0 to n
      if frac( i/10 ) = 0
             then print(i, x[i], y[i], exactsol(x[i]));
        end if: #only print every tenth time
      k := f(x[i], y[i]); #current slope function value
      x[i + 1] := x[i] + h;
      y[i + 1] := y[i] + h·k;
  end do: #how to end a for loop in Maple
```
$$0, 0, 1, 1$$
$$10, 0.10, 1.10463, 1.10517$$
$$20, 0.20, 1.22020, 1.22140$$
$$30, 0.30, 1.34784, 1.34986$$
$$40, 0.40, 1.48887, 1.49182$$
$$50, 0.50, 1.64463, 1.64872$$
$$60, 0.60, 1.81670, 1.82212$$
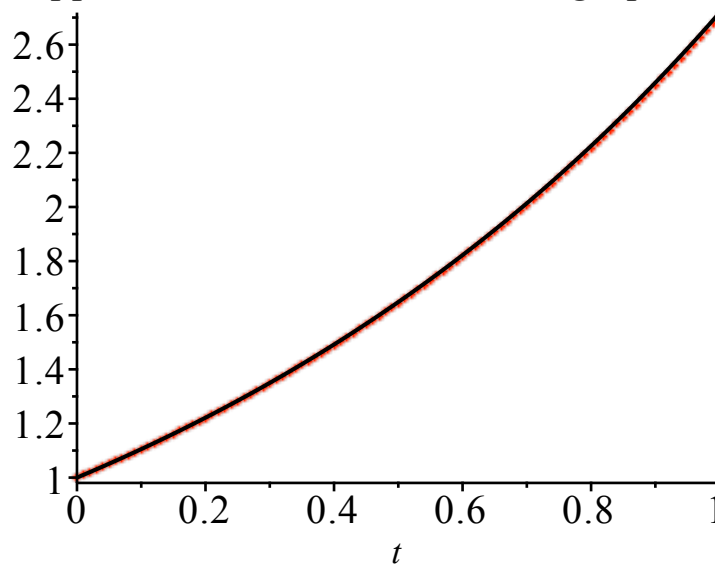$$70, 0.70, 2.00676, 2.01375$$

plot results:

> with(plots) :
Eulerapprox := pointplot({seq([x[i], y[i]], i = 0 .. n)}, color = red) :
exactsolgraph := plot(exactsol(t), t = 0 .. 1, `color` = `black`) :
    #used t because x was already used has been defined to be something else
display({Eulerapprox, exactsolgraph}, title = `approximate and exact solution graphs`);

approximate and exact solution graphs

Exercise 5: For this very special initial value problem

$$y'(x) = y = f(x,y), \text{ slope fun.}$$
$$y(0) = 1$$

which has $y(x) = e^x$ as the solution, set up Euler on the x-interval $[0, 1]$, with n subdivisions, and step size $h = \dfrac{1}{n}$. Write down the resulting Euler estimate for $\exp(1) = e$. What is the limit of this estimate as $n \to \infty$? You learned this special limit in Calculus!

Skip this (I'll fill it in after class)

$x_0 = 0, \ y_0 = 1; \ f(0,1) = 1$

$x_1 = \frac{1}{n}, \ y_1 = y_0 + 1 \cdot \frac{1}{n} = 1 + \frac{1}{n}; \quad f(x_1, y_1) = f\left(\frac{1}{n}, 1 + \frac{1}{n}\right) = 1 + \frac{1}{n}.$

(with underbrace $f(x_0, y_0)$ and h)

$x_2 = \frac{2}{n}; \ y_2 = y_1 + \frac{1}{n} f(x_1, y_1) = \left(1 + \frac{1}{n}\right) + \frac{1}{n}\left(1 + \frac{1}{n}\right) = \left(1 + \frac{1}{n}\right)\left(1 + \frac{1}{n}\right) = \left(1 + \frac{1}{n}\right)^2$

$x_3 = \frac{3}{n}, \ y_3 = \left(1 + \frac{1}{n}\right)^3 \quad \text{(after some work)}$

$\vdots$

$x_n = \frac{n}{n} = 1; \quad \boxed{y_n = \left(1 + \frac{1}{n}\right)^n} \quad \leftarrow$ e.g. with $n = 5$ & $h = .2$, this $= (1.2)^5 \cong 2.488$ *that we did by hand*

Start here Wednesday $\quad \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n = e \ !!$

(L'Hôpital! go look at your Calculus text)

In more complicated differential equations it is a very serious issue to find relatively efficient ways of approximating solutions. An entire field of mathematics, ``numerical analysis'' deals with such issues for a variety of mathematical problems. Our text explores improvements to Euler in sections 2.5 and 2.6, in particular it discusses improved Euler, and Runge Kutta. Runge Kutta-type codes are actually used in commerical numerical packages, e.g. in Maple and Matlab.

Let's summarize some highlights from 2.5-2.6.

Suppose we already knew the solution $y(x)$ to the initial value problem

$$\begin{cases} y'(x) = f(x,y) \\ y(x_0) = y_0. \end{cases} \qquad y'(t) = f(t, y(t))$$

If we integrate the DE from x to $x + h$ and apply the Fundamental Theorem of Calculus, we get

$$y(x + h) - y(x) = \int_x^{x+h} f(t, y(t))\, dt, \text{ i.e.} \qquad y(x+h) - y(x) = \int_x^{x+h} y'(t)\, dt$$

$$\left. y(x + h) = y(x) + \int_x^{x+h} f(t, y(t))\, dt. \right\} \cong f(x, y(x)) \cdot h$$

One problem with Euler is that we approximate this integral above by $h \cdot f(x, y(x))$, i.e. we use the value at the left-hand endpoint as our approximation of the integrand, on the entire interval from x to $x + h$. This causes errors that are larger than they need to be, and these errors accumulate as we move from subinterval to subinterval and as our approximate solution diverges from the actual solution. The improvements to Euler depend on better approximations to the integral. These are subtle, because we don't yet have an approximation for $y(t)$ when t is greater than x, so also not for the integrand $f(t, y(t))$ on the interval $[x, x + h]$.
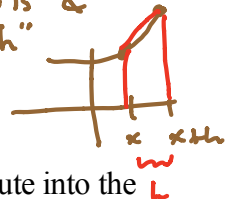
<u>Improved Euler</u> uses an approximation to the Trapezoid Rule to compute the integral in the formula

$$y(x + h) = y(x) + \int_x^{x+h} f(t, y(t)) \, dt.$$

Recall, the trapezoid rule to approximate the integral

$$\int_x^{x+h} f(t, y(t)) \, dt$$

would be

average of integrand @ two endpts & mult by "h"

$$\frac{1}{2} h \cdot (f(x, y(x)) + f(x + h, y(x + h))).$$

Since we don't know $y(x + h)$ we approximate its value with unimproved Euler, and substitute into the formula above. This leads to the improved Euler "pseudocode" for how to increment approximate solutions.

<u>Improved Euler pseudocode:</u>

$$\begin{cases} k_1 = f(x_j, y_j) & \text{#current slope} \\ k_2 = f(x_j + h, y_j + h k_1) & \text{#use unimproved Euler guess for } y(x_j + h) \text{ to estimate slope function when } x = x_j + h \\ k = \dfrac{1}{2}(k_1 + k_2) & \text{#average of two slopes} \\ x_{j+1} = x_j + h & \text{#increment } x \\ y_{j+1} = y_j + h k & \text{#increment } y \end{cases}$$

$f(x, y) = y$

<u>Exercise 6</u> Contrast Euler and Improved Euler for our IVP

$$\begin{array}{c} y'(x) = y \\ y(0) = 1 \end{array} \quad \} \ y(x) = e^x$$

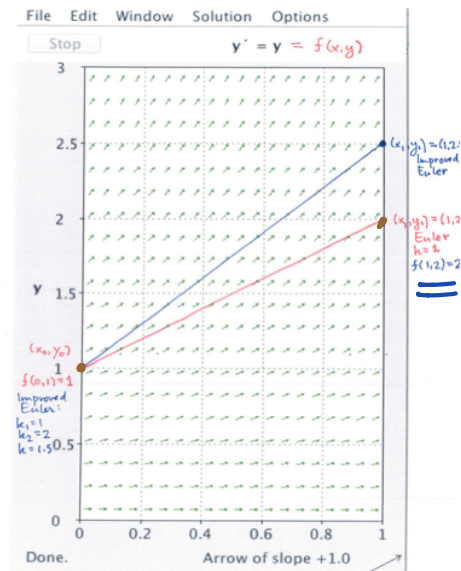to estimate $y(1) \approx$ e with one step of size $h = 1$. Your computations should mirror the picture below. Note that with improved Euler you actually get a better estimate for e with ONE step of size 1 than you did with 5 steps of size $h = 0.2$ using unimproved Euler. (It's still not a very good estimate.)

$x_0 = 0, \ y_0 = 1 \qquad k_1 = 1 \quad (= f(0, 1) = 1)$

$k_2 = f(0+1, y_0 + 1 \cdot 1)$
$\qquad = f(1, 2) = 2$

$k = \frac{1}{2}(k_1 + k_2) = 1.5$

$x_1 = 1, \ y_1 = y_0 + hk = 1 + 1(1.5) = \boxed{2.5}$

In the same vein as ``improved Euler'' we can use the Simpson approximation for the integral for Δy instead of the Trapezoid rule, and this leads to the Runge-Kutta method. You may or may not have talked about Simpson's Parabolic Rule for approximating definite integrals in Calculus. It is based on a quadratic approximation to the integrand g, whereas the Trapezoid rule is based on "linear" approximation.
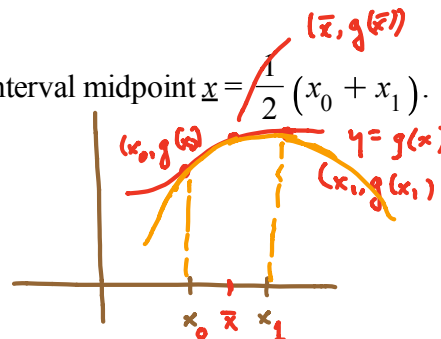
<u>Simpson's Rule:</u>

Consider two numbers $x_0 < x_1$, with interval width $h = x_1 - x_0$ and interval midpoint $\underline{x} = \dfrac{1}{2}(x_0 + x_1)$.
If you fit a parabola $p(x)$ to the three points
$$\left(x_0, g(x_0)\right), \quad (\underline{x}, g(\underline{x})), \quad \left(x_1, g(x_1)\right)$$
then

$$\int_{x_0}^{x_1} p(t)\,dt = \frac{h}{6}\left(g(x_0) + 4\cdot g(\underline{x}) + g(x_1)\right).$$

(You will check this fact in your homework this week!) This formula is the basis for Simpson's rule, which you can also review in your Calculus text or at Wikipedia. (Wikipedia also has a good entry on Runge-Kutta.) Applying the Simpson's rule approximation for our DE, and if we already knew the solution function $y(x)$, we would have

$$y(x+h) = y(x) + \int_x^{x+h} f(t, y(t))\,dt$$

$$\approx y(x) + \frac{h}{6}\cdot\left(f(x, y(x)) + 4f\left(x + \frac{h}{2}, y\left(x + \frac{h}{2}\right)\right) + f(x+h, y(x+h))\right).$$

However, we have the same issue as in Trapezoid - that we've only approximated up to $y(x)$ so far. Here's the pseudo-code for how Runge-Kutta takes care of this. (See also section 2.6 to the text.)

<u>Runge-Kutta pseudocode</u>:

$k_1 = f\left(x_j, y_j\right)$ # left endpoint slope

$k_2 = f\left(x_j + \dfrac{h}{2}, y_j + \dfrac{h}{2}k_1\right)$ # first midpoint slope estimate, using k_1 to increment y_j

$k_3 = f\left(x_j + \dfrac{h}{2}, y_j + \dfrac{h}{2}k_2\right)$ # second midpoint slope estimate using k_2 to increment y_j

$k_4 = f\left(x_j + h, y_j + h\,k_3\right)$ # right hand slope estimate, using k_3 to increment y_j

$k = \dfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$ #weighted average of all four slope estimates, consistent with Simpson's rule

$x_{j+1} = x_j + h$ # increment x

$y_{j+1} = y_j + h\,k$ # increment y

<u>Exercise 7</u> Contrast Euler and Improved Euler to Runge-Kutta for our IVP

$$y'(x) = y \leftarrow f(x,y) = y$$
$$y(0) = 1 \qquad x_0 = 0, \; y_0 = 1$$

to estimate $y(1) \approx$ e with one step of size $\underline{h = 1}$. Your computations should mirror the picture below.
Note that with Runge Kutta you actually get a better estimate for e with ONE step of size $h = 1$ than you
did with 100 steps of size $h = 0.01$ using unimproved Euler!

$x_0 = 0, \; y_0 = 1$

$k_1 = f(x_0, y_0) = f(0,1) = 1$

$k_2 = f(.5, \; y_0 + .5 k_1) = f(.5, 1.5) = 1.5$

$k_3 = f(.5, \; y_0 + .5 \cdot k_2) = f(.5, 1.75)$
$\qquad\qquad\quad 1 \qquad 1.5 \qquad = 1.75$

$k_4 = f(1, \; y_0 + 1 \cdot k_3) = f(1, 2.75) = 2.75$
$\qquad\qquad\quad 1 \qquad 1.75$

$k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$\quad = \frac{1}{6}(1 + 3 + 3.5 + 2.75)$

$\quad = \frac{1}{6}(10.25) \qquad \begin{array}{r} 7.5 \\ 2.75 \\ \hline 2.5 \end{array}$

$\quad = 1.708$

$x_1 = 1$

$y_1 = y_0 + hk = 1 + 1 \cdot (1.708)$
$\qquad\qquad\quad = 2.708$

vs. 100 steps of Euler.
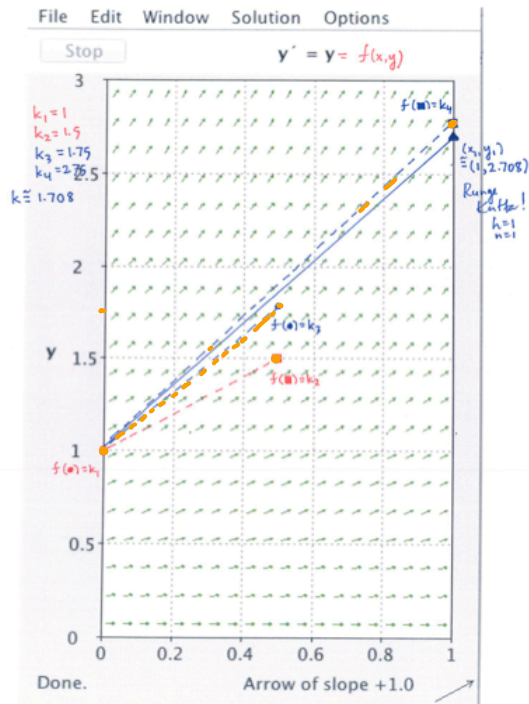
Numerical experiments

I'll illustrate by doing the "famous numbers" problem of estimating e.

Estimate e *by estimating* $y(1)$ *for the solution to the IVP*
$$y'(x) = y$$
$$y(0) = 1.$$
Apply Runge-Kutta with n = 10, 20, 40 ... subintervals, successively doubling the number of subintervals until you obtain the target number below - rounded to 9 decimal digits - twice in succession.

> $evalf(e);$

$$2.718281828459045 \qquad\qquad\qquad\text{(1)}$$

Initialize

> $restart : \# \textit{clear any memory from earlier work}$
> $Digits := 15 : \# \textit{we need lots of digits for "famous numbers"}$
>
> $unassign(`x`, `y`); \# \textit{in case you used the letters elsewhere, and came back to this piece of code}$
> $f := (x, y) \rightarrow y; \# \textit{slope field function for our DE i.e. for y'(x)=f(x,y)}$
> $x[0] := 0; y[0] := 1; \#initial\ point$
> $h := 0.1; n := 10; \quad \#step\ size\ and\ number\ of\ steps - first\ attempt\ for\ "famous\ number\ e"$

Runge Kutta loop. WARNING: ONLY RUN THIS CODE AFTER INITIALIZING

> **for** i **from** 0 **to** n **do** *#this is an iteration loop, with index "i" running from 0 to n*
> $print(i, x[i], y[i]); \textit{\#print iteration step, current x,y, values, and exact solution value}$
> $k1 := f(x[i], y[i]); \textit{\#current slope function value}$
>
> $k2 := f\left(x[i] + \dfrac{h}{2}, y[i] + \dfrac{h}{2} \cdot k1\right);$
>
>   *# first estimate for slope at right endpoint of midpoint of subinterval*
>
> $k3 := f\left(x[i] + \dfrac{h}{2}, y[i] + \dfrac{h}{2} \cdot k2\right); \textit{\#second estimate for midpoint slope}$
>
> $k4 := f(x[i] + h, y[i] + h \cdot k3); \textit{\#estimate for right-hand slope}$
>
> $k := \dfrac{(k1 + 2 \cdot k2 + 2 \cdot k3 + k4)}{6}; \textit{\# Runge Kutta estimate for rate of change of y}$
>
> $x[i + 1] := x[i] + h;$
> $y[i + 1] := y[i] + h \cdot k;$
>   **end do**: *#how to end a for loop in Maple*
>

For other problems you may wish to use or modify the following Euler and improved Euler loops.

Euler loop: WARNING: ONLY RUN THIS CODE AFTER INITIALIZING

```
> for i from 0 to n do  #this is an iteration loop, with index "i" running from 0 to n
      print(i, x[i], y[i]);
          #print iteration step, current x,y, values, and exact solution value
      k := f(x[i], y[i]);  #current slope function value
      x[i + 1] := x[i] + h;
      y[i + 1] := y[i] + h·k;
   end do:  #how to end a for loop in Maple
>
```

Improved Euler loop: WARNING: ONLY RUN THIS CODE AFTER INITIALIZING

```
> for i from 0 to n do  #this is an iteration loop, with index "i" running from 0 to n
      print(i, x[i], y[i]);  #print iteration step, current x,y, values, and exact solution value
      k1 := f(x[i], y[i]);  #current slope function value
      k2 := f(x[i] + h, y[i] + h·k1);  #estimate for slope at right endpoint of subinterval
      k := (k1 + k2)/2;  # improved Euler estimate
      x[i + 1] := x[i] + h;
      y[i + 1] := y[i] + h·k;
   end do:  #how to end a do loop in Maple
>
```

Math 2250-004
Friday Feb 3

- Finish the discussion of 2.4-2.6, numerical methods for solving first order IVP's, if necessary. Also demo the matlab program you'll be using for you lab, if necessary. ☺

- In the event that there's any left-over time, use today's notes to introduce Chapter 3, in particular 3.1 -3.2 Linear systems of (algebraic) equations and how to solve them...the first exercise relates to this week's homework.

Exercise 1: Find the quadratic function $p(x) = a x^2 + b x + c$ so that the graph $y = p(x)$ interpolates (i.e. passes through) the three points $(-1, 0), (0, 2), (1, 2)$:

$$\begin{cases} p(-1) = 0 = a \cdot 1 - b + c = a - b + c \\ p(0) = 2 = c \\ p(1) = 2 = a + b + c \end{cases}$$

quadratic fit to three points
(-1,0),(0,2),(1,2)

$y = p(x)$

$a - b + c = 0$

$\boxed{c = 2}$

$a + b + c = 2$

$E_1 \implies$ $a - b + 2 = 0$
E_2 $a + b + 2 = 2$

$E_1 + E_2:$ $2a + 4 = 2$
$2a = -2$
$\boxed{a = -1}$

$E_2: \quad -1 + b + 2 = 2$
$\boxed{b = 1}$
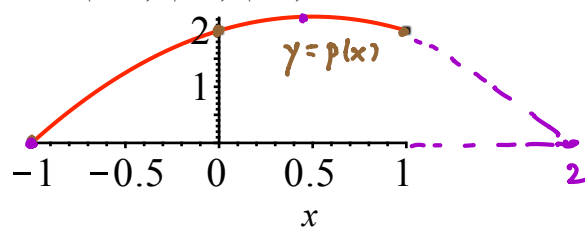
$p(x) = -x^2 + x + 2 = -(x^2 - x - 2) = -(x+1)(x-2)$

check: $p(-1) = -1 - 1 + 2 = 0$ ✓
$p(0) = 2$ ✓
$p(1) = -1 + 1 + 2 = 2$ ✓

In Chapters 3-4 we temporarily leave differential equations in order to study basic concepts in linear algebra. You've all studied linear systems of equations and matrices before, and that's where Chapter 3 starts. Linear algebra is foundational for many different disciplines, and in this course we'll use the key ideas when we return to higher order linear differential equations and to systems of differential equations. As it turns out, there's an example of solving simultaneous linear equations in this week's homework. It's related to Simpson's rule for numerical integration, which is itself related to the Runge-Kutta algorithm for finding numerical solutions to differential equations.

In 3.1-3.2 our goal is to understand systematic ways to solve simultaneous linear equations. Although we used a, b, c for the unknowns in the previous problem, this is not our standard way of labeling.
- We'll often call the unknowns $x_1, x_2, \ldots x_n$, or write them as elements in a vector

$$\underline{x} = \left[x_1, x_2, \ldots x_n \right]^{\top} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Then the general linear system (LS) of m equations in the n unknowns can be written as

$$\left\{ \begin{array}{l} a_{11}\, x_1 + a_{12}\, x_2 + \ldots + a_{1n}\, x_n = b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \ldots + \boxed{a_{2n}}\, x_n = b_2 \\ \quad\vdots \qquad\qquad \vdots \quad\; \vdots \\ a_{m1}\, x_1 + a_{m2}\, x_2 + \ldots + a_{mn}\, x_n = b_m \end{array} \right.$$

m eqtns

a_{ij} : i^{th} eqtn
j^{th} variable coefft

where the coefficients a_{ij} and the right-side number b_j are known. The goal is to find values for the vector $\underline{x}$ so that all equations are true. (Thus this is often called finding "simultaneous" solutions to the linear system, because all equations will be true at once.)

Notice that we use two subscripts for the coefficients a_{ij} and that the first one indicates which equation it appears in, and the second one indicates which variable its multiplying; in the corresponding *coefficient matrix A* , this numbering corresponds to the row and column of a_{ij}:

$$A := \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & \boxed{a_{23}} & \cdots & a_{2n} \\ & \vdots & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

2nd row
3rd column

Let's start small, where geometric reasoning will help us understand what's going on:
Exercise 2: Describe the solution set of each single equation below; describe and sketch its geometric realization in the indicated Euclidean spaces.   just one linear equation.
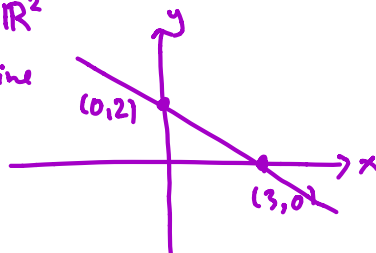2a) $3x = 5$, for $x \in \mathbb{R}$.
2b) $2x + 3y = 6$, for $[x, y] \in \mathbb{R}^2$.
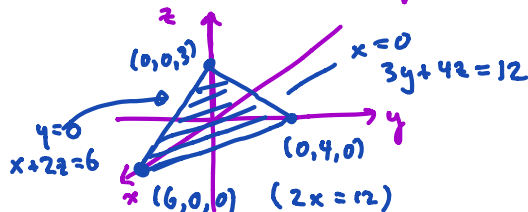2c) $2x + 3y + 4z = 12$, for $[x, y, z] \in \mathbb{R}^3$.

$5/3$

2a) $3x = 5 \implies x = 5/3$ point.

2b) $2x + 3y = 6$, $\begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$
    soln set is line

$(0,2)$

$(3,0)$

2c) solution set is plane!

$(0,0,3)$
$x=0$
$3y + 4z = 12$
$y=0$
$x + 2z = 6$
$(0,4,0)$
$(6,0,0)$ $(2x = 12)$

2 linear equations in 2 unknowns:

solns
in $\mathbb{R}^2$

$$a_{11} x + a_{12} y = b_1$$
$$a_{21} x + a_{22} y = b_2$$

goal: find all $[x, y]$ making both of these equations true. So geometrically you can interpret this problem as looking for the intersection of two lines.

Exercise 3: Consider the system of two equations E_1, E_2:

$$E_1 \qquad 5x + 3y = 1$$
$$E_2 \qquad x - 2y = 8$$

3a) Sketch the solution set in $\mathbb{R}^2$, as the point of intersection between two lines.
3b) Use the following three "elementary equation operations" to systematically reduce the system E_1, E_2 to an equivalent system (i.e. one that has the same solution set), but of the form

$$1x + 0y = c_1$$
$$0x + 1y = c_2$$

(so that the solution is $x = c_1, y = c_2$). Make sketches of the intersecting lines, at each stage.

The three types of elementary equation operation are below. Can you explain why the solution set to the modified system is the same as the solution set before you make the modification?
- interchange the order of the equations •
- multiply one of the equations by a non-zero constant
- replace an equation with its sum with a multiple of a different equation. •

$E_1 \qquad 5x + 3y = 1$

$E_2 \qquad x - 2y = 8$

$E_2 \qquad x - 2y = 8$
$E_1 \qquad 5x + 3y = 1$.

$\qquad x - 2y = 8$
$-5E_1 + E_2 \qquad 0 + 13y = -39$.

$\qquad x - 2y = 8$
$E_2 / 13 \qquad y = -3$ ←

$2E_2 + E_1 \qquad x \quad = 2$

$\qquad y = -3$

sol'n is $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$

check: $5 \cdot 2 + 3(-3) = 1$ ✓
$\qquad 2 - 2(-3) = 8$ ✓

$\begin{array}{cc|c} 5 & 3 & 1 \\ 1 & -2 & 8 \end{array}$

$3y = -5x + 1$
$y = -\frac{5}{3}x + \frac{1}{3}$

L_2

$(8,0)$

$(0,-4)$

L_1

L_1

8

$(0,-4)$

$L_1 \quad x = 2$

$y = 3$
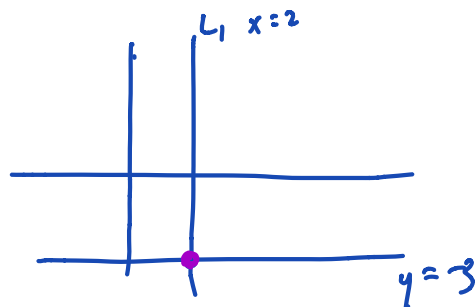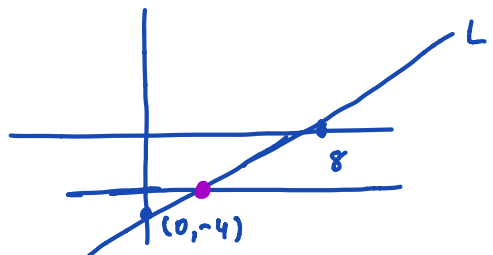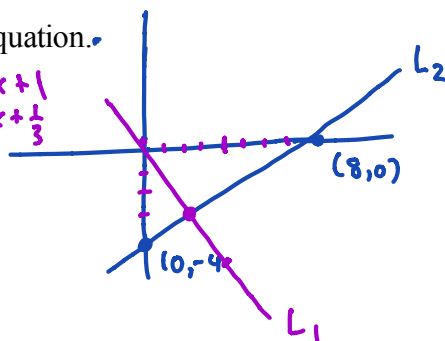
3c) Look at your work in 3b. Notice that you could have save a lot of writing by doing this computation "synthetically", i.e. by just keeping track of the coefficients and right-side values. Using R_1, R_2 as symbols for the rows, your work might look like the computation below. Notice that when you operate synthetically the "elementary equation operations" correspond to "elementary row operations":

- interchange two rows •
- multiply a row by a non-zero number
- replace a row by its sum with a multiple of another row.

$$
\begin{array}{c|cc|c}
& 5 & 3 & 1 \\
& 1 & -2 & 8 \\
\hline
R_2 & 1 & -2 & 8 \\
R_1 & 5 & 3 & 1 \\
\hline
& 1 & -2 & 8 \\
-5R_1 + R_2 & 0 & 13 & -39 \\
\hline
& 1 & -2 & 8 \\
R_2/13 & 0 & 1 & -3 \\
\hline
2R_2 + R_1 & 1 & 0 & 2 \\
& 0 & 1 & -3 \\
\end{array}
\longrightarrow
\boxed{\begin{array}{l} x = 2 \\ y = -3 \end{array}}
$$

3d) What are the possible geometric solution sets to 1, 2, 3, 4 or any number of linear equations in two unknowns?

line, pt, $\mathbb{R}^2$, $\emptyset$
empty set

Solutions to linear equations in 3 unknowns:

What is the geometric question you're answering? $\big]$ (intersection) common points of a collection of planes

Exercise 4) Consider the system
$$x + 2y + z = 4$$
$$3x + 8y + 7z = 20$$
$$2x + 7y + 9z = 23 .$$

Use elementary equation operations (or if you prefer, elementary row operations in the synthetic version) to find the solution set to this system. There's a systematic way to do this, which we'll talk about. It's called Gaussian elimination.

Hint: The solution set is a single point, $[x, y, z] = [5, -2, 3]$.

$$-3\textcircled{1} \quad -2 \quad -\cancel{1} \mid 4$$
$$3 \quad 8 \quad 7 \mid 20$$
$$2 \quad 7 \quad 9 \mid 23$$

$$\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & \textcircled{2} & 4 & 8 \\ 0 & 3 & 7 & 15 \end{array}$$

$-3R_1+R_2$
$-2R_1+R_3$

$$\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & \textcircled{1} & 2 & 4 \\ 0 & 3 & 7 & 15 \end{array}$$

$R_2/2$

$$\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 3 \end{array}$$

$-3R_2+R_3$

$$\begin{array}{ccc|c} 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}$$

$-R_3+R_1$
$-2R_3+R_2$

$$\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 3 \end{array}$$

$-2R_2+R_1$

backsolve

$z = 3$

$\Rightarrow x + 2(-2) + 1\cdot 3 = 4 \Rightarrow x = 5$

$\Rightarrow y + 2z = 4 \Rightarrow y = 4 - 2z = -2$

"unclash"

$x = 5$
$y = -2$
$z = 3$

Exercise 5 There are other possibilities. In the two systems below we kept all of the coeffients the same as in Exercise 4, except for a_{33} , and we changed the right side in the third equation, for 4$\underline{a}$. Work out what happens in each case.

5a)

$$x + 2\,y + z = 4$$
$$3\,x + 8\,y + 7\,z = 20$$
$$2\,x + 7\,y + 8\,z = 20 \ .$$

5b)

$$x + 2\,y + z = 4$$
$$3\,x + 8\,y + 7\,z = 20$$
$$2\,x + 7\,y + 8\,z = 23 \ .$$

5c) What are the possible solution sets (and geometric configurations) for 1, 2, 3, 4,... equations in 3 unknowns?