August 27 - 0, I.1, I.2

Monday, August 26, 2024 09:25

Math 246:

Ch 0:

A differential equation is an algebraic relation between functions and their derivatives & independendent variables.

eg. $\left(\frac{dx}{dt}\right)^2 + x\sin t = \cos x$

eg. $y'' + ty' + y = \cos t$ $\quad\left(y' = \frac{dy}{dt}\right)$

eg. $\frac{\partial y}{\partial t} \cdot \frac{\partial y}{\partial s} + y\frac{\partial z}{\partial t} = \sin(st)$

Notation: For $\frac{dy}{dx}$, we can write y' or $\dot{y}$

An ordinary differential equation (ODE) has no partial derivatives. (as opposed to a partial DE, PDE).

The order of a DE is the order of the highest deriv.

that appears.

eg $\left(\frac{dx}{dt}\right)^2 + x\sin t = \cos x$ $\longrightarrow$ order = 1

eg. $y'' + ty' + y = \cos t$ $\longrightarrow$ order : 2

An ODE with function y and independent variable t is <u>linear</u>

if it can be written as:

$$a_n(t)\, y^{(n)} + a_{n-1}(t)\, y^{(n-1)} + \cdots + a_1(t)\, y' + a_0(t)\, y = f(t).$$

($y^{(n)}$ = n^{th} derivative of y)

eg. $\left(\frac{dx}{dt}\right)^2 + x\sin t = \cos x$

$\rightarrow$ not linear : $\left(\frac{dx}{dt}\right)^2$ and $\cos x$

eg. $y'' + ty' + y = \cos t$

linear

eg. $y^{(4)} + y^{(2)} = 2$

linear

A <u>System</u> of ODEs is $\begin{cases} \text{ODE } 1 \\ \text{ODE } 2 \\ \vdots \\ \text{ODE } n \end{cases}$

I: First-order ODEs

Ch 1: Intro

First-order ODEs can be complicated.

We will deal with those that can be put into the form $\boxed{\dfrac{dy}{dt} = f(t, y).}$ ← standard form

eg. $6W \dfrac{dW}{dz} = -z$

$$\rightsquigarrow \quad \frac{dW}{dz} = \frac{-z}{6W}.$$

A function $Y(t)$ is a <u>solution</u> to $y' = f(t,y)$ on the interval (a,b) if:

- $Y(t)$ and $Y'(t)$ exist on (a,b)
- $f(t, Y(t))$ exists on (a,b)
- $Y'(t) = f(t, Y(t))$ on (a,b).

<u>eg</u>. $y' = \dfrac{-t}{y}$, $Y(t) = \sqrt{4-t^2}$.

<u>Check</u>: $Y'(t) = \dfrac{-t}{\sqrt{4-t^2}}$.

$Y(t)$ is defined on $[-2,2]$

But $f(t, Y(t)) = \dfrac{-t}{\sqrt{4-t^2}}$ is only defined for $(-2,2)$.

not at $t = \pm 2$.

So $Y(t)$ is a solution on $(-2,2)$, not on $[-2,2]$.

<u>Explicit Equations</u>

These are $y' = f(t)$.

These are $y' = f(t)$.

The general solution is $y = \int f(t)\,dt = F(t) + C$

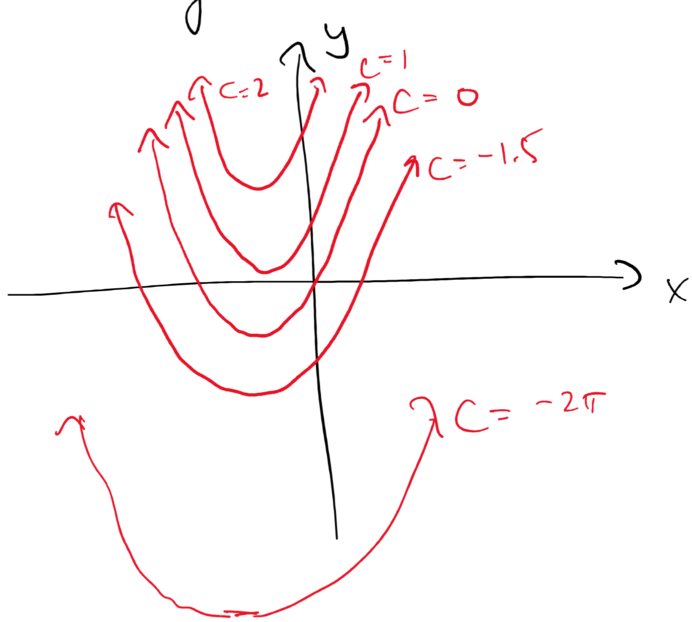
where $F(t)$ is an antiderivative of $f(t)$

$\hookrightarrow (F'(t) = f(t))$

C = constant

eg $\dfrac{dy}{dx} = 2x+1$

$\rightsquigarrow y = x^2 + x + C$ is the general solution.



To pick out one solution out of the general sol'n, need an initial condition: $\boxed{y(t_I) = y_I}$.

the pair $y' = f(t)$ with $y(t_I) = y_I$ is called an initial value problem. (IVP)

eg. Solve the IVP $\frac{dy}{dx} = 2x + 1$ with $y(0) = 2$.

Sol'n: The general solution is $y = x^2 + x + C$.

$y(0) = 2 \longrightarrow 2 = (0)^2 + (0) + C$

$\Rightarrow C = 2$

So $y = x^2 + x + 2$ is the solution.

The interval of definition/existence of a solution to an IVP is the largest interval (a,b) with

- $t_I \in (a,b)$
- $f(t)$ is continuous on (a,b)

Ch2: Linear Equations

These look like

$$p(t)\, y' + q(t)\, y = r(t).$$

and $p(t) \neq 0$ for the values of t we are considering.

in standard form : $y' = -\underbrace{\frac{q(t)}{p(t)}}_{a(t)}\, y + \underbrace{\frac{r(t)}{p(t)}}_{f(t)}$

we will write $y' + a(t)\, y = f(t)$.

$f(t)$ is called the <u>forcing function</u>.

if $f(t) = 0$, the ODE is called <u>homogeneous</u>

<u>Recipe for solving first-order linear ODEs</u> :

$$y' + a(t)\, y = f(t).$$

Choose an antiderivative $A(t)$ of $a(t)$.

Multiply both sides by $e^{A(t)}$

$$\rightarrow \underbrace{e^{A(t)} y' + a(t) e^{A(t)} y}_{\text{product rule}} = \underbrace{f(t) e^{A(t)}}_{g(t)}.$$

$$\rightarrow \frac{d}{dt}\left(e^{A(t)} y \right) = g(t)$$

antideriv. of $g(t)$

Integrate both sides : $e^{A(t)} y = G(t) + C$

$$\Rightarrow y = e^{-A(t)} G(t) + C e^{-A(t)}.$$

is the general solution.

eg. $\frac{dy}{dt} = -y$

Sol'n : rewrite as $y' + y = 0$

$a(t) = 1 \rightarrow$ Choose $A(t) = t$

Multiply both sides by e^t : $e^t y' + e^t y = 0$

$$\rightarrow \frac{d}{dt}(e^t y) = 0$$

Integrate : $e^t y = C \Rightarrow y = Ce^{-t}.$

Integrate: $e^t y = C \Rightarrow y = Ce$. is the general solution.

product rule: $\frac{d}{dt}(e^t)\, y + e^t \frac{d}{dt}(y) = e^t y + e^t y'$

eg. $y' = -y + e^t$

<u>Sol'n</u>: rewrite $y' + \underbrace{y}_{\downarrow} = e^t$

$a(t) = 1 \longrightarrow \overset{\text{pick}}{A(t)} = t$

Mult. by e^t : $e^t y' + e^t y = e^{2t}$

$\longrightarrow (e^t y)' = e^{2t}$

Integrate: $e^t y = \frac{1}{2} e^{2t} + C$

$\longrightarrow y = \frac{1}{2} e^t + Ce^{-t}$ is the general sol'n.

eg. Solve the IVP $\frac{dx}{dt} + \underline{\cos t}\, x = \cos t$ with $x(\frac{\pi}{2}) = 0$.

<u>Sol'n</u>: $a(t) = \cos t$, so choose $A(t) = \sin t$.

Mult. both sides by $e^{\sin t}$.

$$e^{\sin t}x' + \underbrace{\cos t\, e^{\sin t}}_{\frac{d}{dt}\left(e^{\sin t}\right)} x = \underbrace{\cos t\, e^{\sin t}}_{\frac{d}{dt}\left(e^{\sin t}\right)}$$

$$\longrightarrow \quad \left(e^{\sin t}x\right)' = \cos t\, e^{\sin t}$$

Integrate: $e^{\sin t}x = \int \cos t\, e^{\sin t}\,dt = e^{\sin t} + C$

$$\Rightarrow \quad x = 1 + Ce^{-\sin t}.$$

with $x(\frac{\pi}{2}) = 0$: $\quad 0 = 1 + Ce^{-\sin(\frac{\pi}{2})}$

$$\Rightarrow \quad 0 = 1 + Ce^{-1}$$

$$\Rightarrow \quad C = -e$$

$$\Rightarrow \quad \underline{x = 1 - e^{1-\sin t}}.$$

August 29 - I.2, I.3

Thursday, August 29, 2024 10:58

I.2 (cont.)

Consider $y' + a(t)y = f(t)$, $y(t_I) = y_I$.

If $a(t)$, $f(t)$ are continuous over the interval (a,b)

$t_I \in (a,b)$, then there is a unique solution to the IVP

that is continuous on (a,b), and it's given by our method.

eg. Consider $z' + \cot(t)\, z = \dfrac{1}{\ln(t^2)}$, $z(4) = 3$.

Find the largest interval on which we can guarantee a unique continuous sol'n to this IVP.

Sol'n: $\ln(t^2)$ is continuous on $(-\infty, 0)$ and $(0, \infty)$

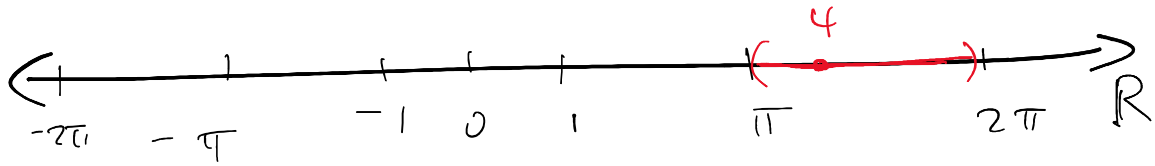
$\frac{1}{\ln(t^2)}$ is thus continuous except at $t=0$, and when $\ln(t^2) = 0$

i.e. $t = \pm 1$

$\cot(t)$ has discontinuities at multiples of π.

the largest interval of continuity that includes $t = 4$

is $\underline{(\pi, 2\pi)}$.



$\underline{\text{I}.3}$: Separable Equations

A first-order ODE is *separable* if it can be written as

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

e.g. $y' = 2ty^2 + 3t^2y^2$

$$= \underbrace{(2t + 3t^2)}_{f(t)}\underbrace{y^2}_{g(y)}$$

An ODE $y' = g(y)$ is called *autonomous*

A solution is called <u>stationary</u> if it's constant.

If $y = C$ is a stationary sol'n, then

$y' = 0 \Rightarrow \boxed{0 = g(C).}$

<u>eg</u> $y' = 4y - y^3$. Find the stationary sol'ns.

<u>Sol'n</u>: Set $4y - y^3 = 0$.

$\Rightarrow y(4 - y^2) = 0$

$\Rightarrow y(2-y)(2+y) = 0 \rightarrow y = 0, 2, -2.$

So $y = 0$, $y = 2$, $y = -2$ are stationary sol'ns.

To find other sol'ns (non-stationary sol'ns):

$y' = g(y) \rightsquigarrow \frac{1}{g(y)} y' = 1$

$$\rightsquigarrow \int \frac{1}{g(y)} y' \, dt = \int 1 \, dt = t + C$$

"$\frac{dy}{dt} \cdot dt = dy$" $\rightsquigarrow$ $$\int \frac{1}{g(y)} \, dy = t + C$$

The result is an <u>implicit</u> equation for our solution.

Why can we divide by $g(y)$? $g(y) = 0$ corresponds to stationary sol'ns. We're looking for non-stationary sol'ns, i.e. $g(y) \neq 0$.

<u>eg</u>. Find all solution to $y' = y^2$

<u>Sol'n</u>: Stationary sol'ns: set $y^2 = 0 \longrightarrow \underline{y = 0}$

For other sol'ns:

$$\frac{1}{y^2} y' = 1$$

$y' dt$

$$\rightsquigarrow \int \frac{1}{y^2}\, dy = \int 1\, dt$$

$y'\,dt$

$$\rightsquigarrow -\frac{1}{y} = t + C$$

We can find an explicit sol'n:

$$\rightarrow -y = \frac{1}{t+C} \rightarrow y = \frac{-1}{t+C}.$$

Each $y = \frac{-1}{t+C}$ is actually two sol'ns,

one on $(-\infty, -C)$, one on $(-C, \infty)$

Note: our sol'n is discontinuous, even though all functions in the ODE $y' = y^2$ are continuous.

General Separable Eqns:

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

if $g(c)=0$, then $y=C$ is still a Stationary sol'n. so set $\underline{g(y)=0}$.

Non-stationary sol'ns:

$$\longrightarrow \frac{1}{g(y)}y' = f(t)$$

$$\longrightarrow \int \frac{1}{g(y)}\underbrace{y'dt}_{=dy} = \int f(t)dt$$

$$\longrightarrow \int \frac{1}{g(y)}dy = F(t)+C$$

$\rightarrow$ antideriv. of $f(t)$

eg. Find all sol'ns to $\dfrac{dz}{dx} = \dfrac{3x+xz^2}{z+x^2z}$.

Sol'n: $$\frac{dz}{dx} = \frac{x\cdot(3+z^2)}{(1+x^2)\cdot z} = \underbrace{\left(\frac{x}{1+x^2}\right)}_{f(x)}\underbrace{\left(\frac{3+z^2}{z}\right)}_{g(z)}$$

Stationary sol'ns: $g(z)=0 \longrightarrow \dfrac{3+z^2}{z}=0$

$\rightarrow 3+z^2=0$ no solution.

Non-stationary sol'ns: $\longrightarrow \frac{1}{g(z)}\frac{dz}{dx} = f(x)$

$$\longrightarrow \left(\frac{z}{3+z^2}\right)\frac{dz}{dx} = \frac{x}{1+x^2}.$$

$$\longrightarrow \int\left(\frac{z}{3+z^2}\right)\underbrace{\frac{dz}{dx}\,dx}_{=dz} = \int \frac{x}{1+x^2}\,dx$$

u-sub: $u = 3+z^2$
$du = 2z\,dz$

v-sub: $v = 1+x^2$
$dv = 2x\,dx$

$$\longrightarrow \int \frac{\frac{1}{2}}{u}\,du = \int \frac{\frac{1}{2}}{v}\,dv$$

$$\longrightarrow \frac{1}{2}\ln|u| = \frac{1}{2}\ln|v| + C$$

$$\longrightarrow \frac{1}{2}\ln|3+z^2| = \frac{1}{2}\ln|1+x^2| + C.$$

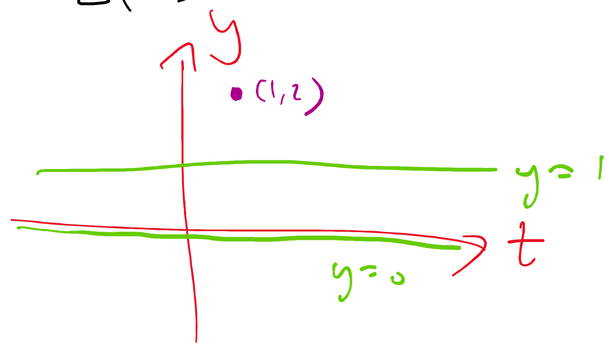
IVPs:

eg. $y' = ty^2 - ty$, $y(1) = 2$
$= t(y^2-y)$

Sol'n: Stationary sol'ns: $y^2 - y \overset{\text{set}}{=} 0$

$\longrightarrow \quad y = 0, 1$

neither satisfies the IVP.



other sol'ns:

$\longrightarrow \quad \frac{1}{y^2-y}\frac{dy}{dt} = t \quad \longrightarrow \quad \frac{1}{y^2-y}\,dy = t\,dt$

$\longrightarrow \quad \int \frac{1}{y^2-y}\,dy = \int t\,dt$

partial fractions: $\frac{1}{y(y-1)} = \frac{A}{y} + \frac{B}{y-1}$

$\Rightarrow \quad 1 = A(y-1) + By$

guess $A=-1,\ B=1$

$\int \frac{-1}{y} + \frac{1}{y-1}\,dy = \int t\,dt$

$\Rightarrow \quad -\ln|y| + \ln|y-1| = \frac{t^2}{2} + C$

$$\Rightarrow \quad -\ln|y| + \ln|y-1| = \frac{t^2}{2} + C$$

$\ln(a) - \ln(b)$
$\quad ''$
$\ln(a/b)$

$$\Rightarrow \quad \ln\left|\frac{y-1}{y}\right| = \frac{t^2}{2} + C.$$

our initial condition is $y(1) = 2$

$$\longrightarrow \ln\left|\frac{2-1}{2}\right| = \frac{(1)^2}{2} + C$$

$$\rightarrow \ln\left(\frac{1}{2}\right) = \frac{1}{2} + C \quad \rightarrow \quad C = \ln\left(\frac{1}{2}\right) - \frac{1}{2}$$

So $\ln\left|\frac{y-1}{y}\right| = \frac{t^2}{2} + \ln\left(\frac{1}{2}\right) - \frac{1}{2}.$

Uniqueness & Existence Thm:

If $f(t)$ is continuous on (a,b)
$g(y)$ is cont. & diff. on (c,d)

then for every $t_I \in (a,b)$, $y_I \in (c,d)$
there is a unique continuous sol'n to $y' = f(t)g(y)$
with $y(t_I) = y_I$ defined on some interval
around t_I. It's determined by our method.

---

non-eg. $\frac{dy}{dt} = 3y^{2/3}$, $y(0) = 0$.

$y = 0$ is a stationary sol'n, and solves our IVP.

But $g(y) = 3y^{2/3}$ is not diff. at $y = 0$

might have other sol'ns!

$$\frac{1}{3y^{2/3}} \frac{dy}{dt} = 1 \rightarrow \int \frac{1}{3y^{2/3}}\, dy = \int 1\, dt$$

$$\rightarrow \quad y^{1/3} = t + C$$

$$\rightarrow \quad y = (t+c)^3$$

For $y(0) = 0$, get $C = 0 \rightarrow y = t^3$

also solves our IVP.

September 3 - I.4, I.5

Tuesday, September 3, 2024 07:19

Quiz tomorrow — up to I.3

I.4 : Theory :

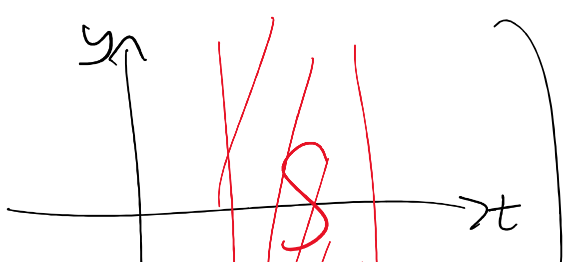
Consider IVPs $y' = f(t,y)$, $\underbrace{y(t_I) = y_I}$.

we say that the problem is <u>well-posed</u> if

- there is a solution
- the solution is unique
- the solution depends continuously on the data.

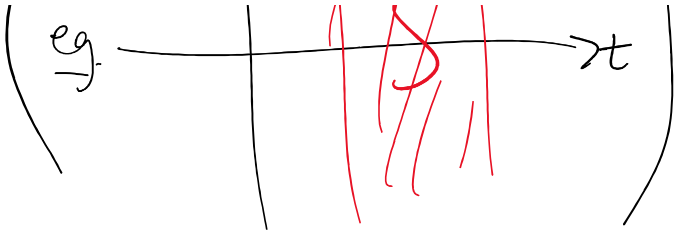
Existence & Uniqueness :

Theorem : Consider a set S of points in the (t,y)-plane.

(eg. )

If $f(t,y)$ is continuous on S,

and ∂f is continuous on S,

 and $\frac{\partial f}{\partial y}$ is continuous on S,

then for any (t_I, y_I) in S, there is a unique continuous solution $Y(t)$ to $y' = f(t,y)$, $y(t_I) = y_I$ defined over some interval (a,b) containing t_I.

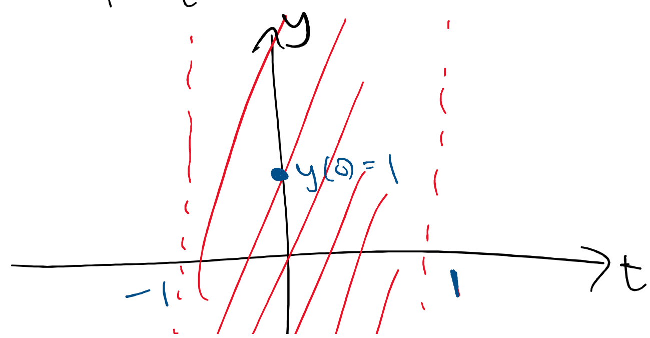
can make (a,b) large as long as $(t, Y(t))$ lies inside S.

eg Consider $y' = \frac{\sin(t + ty^2)}{1-t^2}$, $y(0) = 1$.

Show that there is a unique sol'n defined on $(-1,1)$

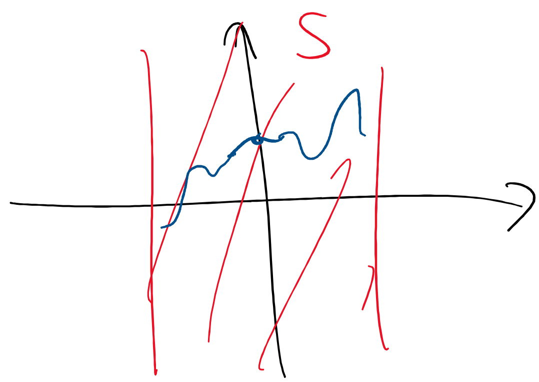
Sol'n: For $f(t,y) = \frac{\sin(t+ty^2)}{1-t^2}$, continuous except at $t = \pm 1$.

$$\frac{\partial f}{\partial y} = \frac{2ty\cos(t+ty^2)}{1-t^2},$$ continuous except at $t = \pm 1$.

So choose S: 

Since $t_I = 0, y_I = 1 \rightsquigarrow (0,1)$ is in S.

Theorem tells us we have a unique sol'n $Y(t)$ defined on larger (a,b) such that $Y(t)$ remains inside S.



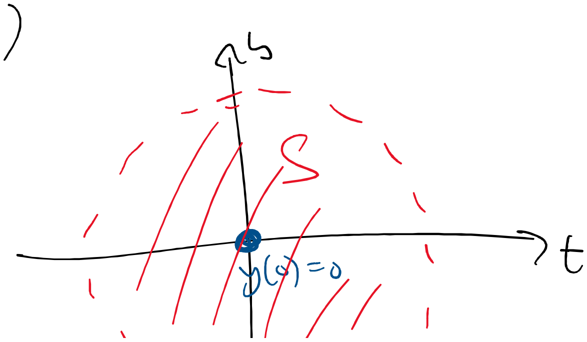
Since any solution will not leave S as long as $-1<t<1$,

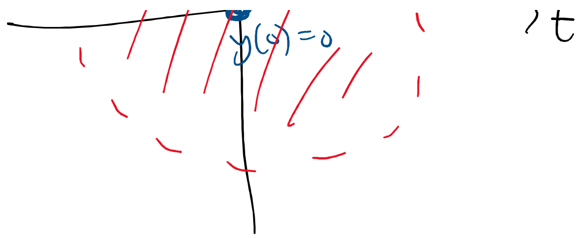
we get $(a,b) = (-1,1)$.

eg. $y' = \dfrac{1}{t^2+y^2-1}$, $y(0)=0$. Find S.

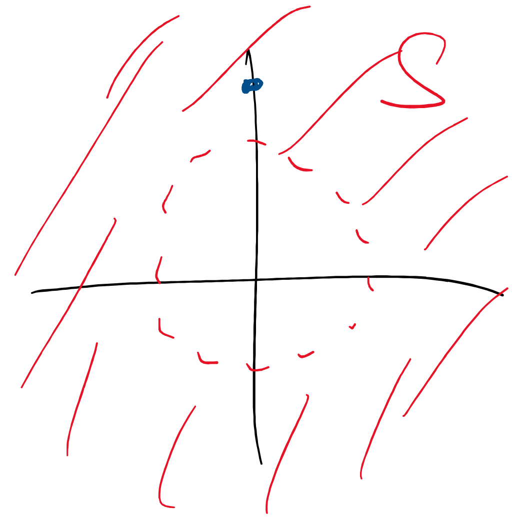
Sol'n: function is discont. when $\underbrace{t^2+y^2=1}_{\text{circle of } r=1}$

(Check $\frac{\partial f}{\partial y}$ too)





If we want $y(0) = 3$



I.5 – Graphical Methods

Phase Portraits for Autonomous Equation:

$$\frac{dy}{dt} = g(y).$$

Without solving, want to describe qualitative behavior of solutions.

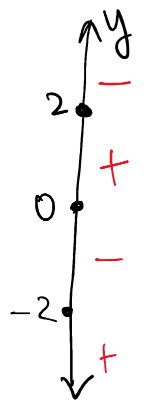
- where $g(y) = 0$, $y' = 0$, and we have a stationary sol'n.
- where $g(y) > 0$, $y' > 0$, so y is increasing.

- where $g(y) > 0$, $y' > 0$, so y is increasing.
- where $g(y) < 0$, $y' < 0$, so y is decreasing.

eg. $y' = 4y - y^3$

$$4y - y^3 = y(2-y)(2+y)$$

$\rightsquigarrow$ stationary sol'ns are $y = 0, \pm 2$.



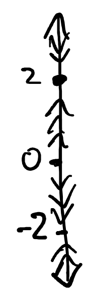
plug in test points in between zeroes to determine sign:

$g(y) = 4y - y^3$:

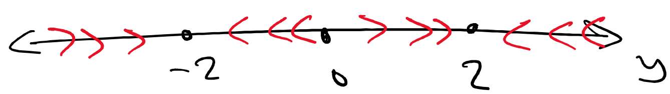
$g(1) = 3 > 0$, $g(3) = 12 - 27 < 0$

$g(-1) = -4 + 1 < 0$, $g(-3) = -12 + 27 > 0$

Describe this information as:



or



This is called the <u>phase portrait</u> or <u>phase line</u>

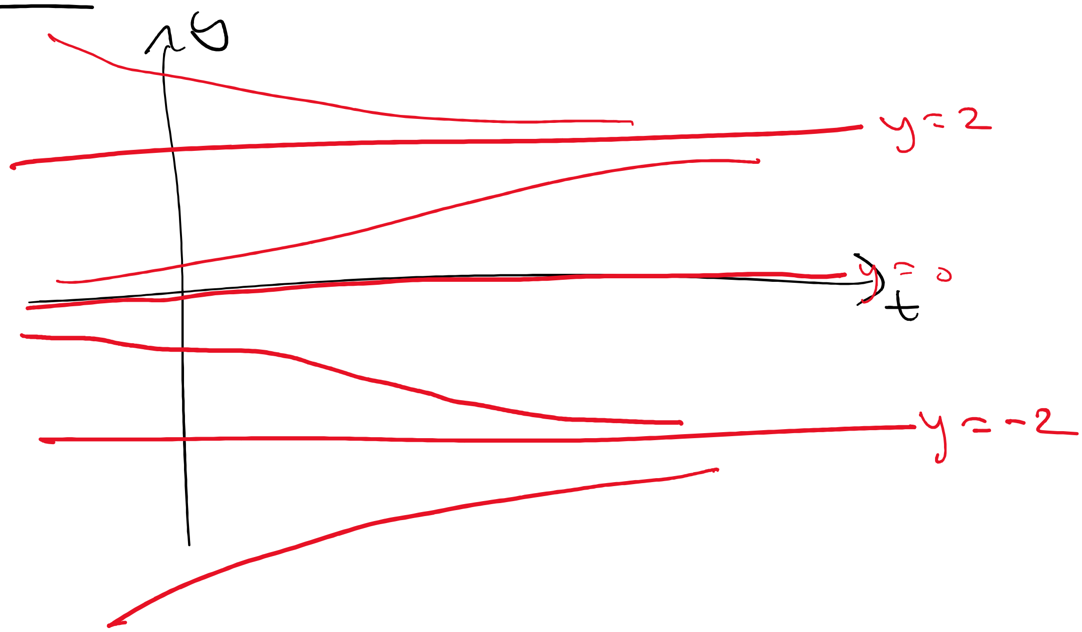
We learn from this portrait:

if $y(t)$ starts in $(-\infty,-2)$, then $y(t)$ increases as t increases, and $y(t) \to -2$ as $t \to \infty$. (but never reaches -2)

if $y(t)$ starts in $(-2,0)$, then $y(t)$ is decreasing, and $y(t) \to -2$ as $t \to \infty$.

if $y(t)$ starts in $\begin{cases}(0,2)\\(2,\infty)\end{cases}$ then $y(t)$ is $\begin{cases}\text{increasing}\\\text{decreasing}\end{cases}$, and $y(t) \to 2$ as $t \to \infty$.

Sketch solutions:



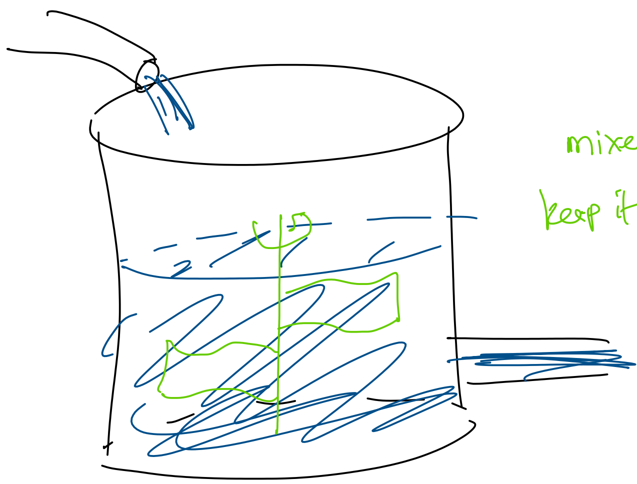
We c.

September 5 - I.6

Thursday, September 5, 2024 10:57

I.6 : Application

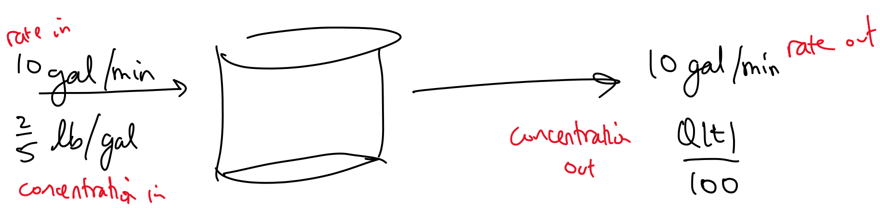
Tanks & Mixtures :



mixer in the tank to keep it well-mixed.

eg. A tank initially contains 100 gallons of brine (water + salt) with 20 lbs of salt dissolved in it. Brine containing $\frac{2}{5}$ lb/gal salt flows in at 10 gal/min. The well-mixed mixture flows out of the tank at 10 gal/min. Find a formula for the salt content after t minutes.

Sol'n : let $Q(t)$ = quantity of salt at time t.



then $\frac{dQ}{dt}$ = salt in − salt out

$$= (10)\left(\frac{2}{5}\right) - (10)\left(\frac{Q(t)}{100}\right)$$

and $Q(0) = 20$.

So our IVP is $\quad Q' = 4 - \frac{1}{10} Q \,, \quad Q(0) = 20.$

Solve the IVP: it's linear: $\quad Q' + \frac{1}{10} Q = 4$

$a(t) = \frac{1}{10} \quad \Rightarrow A(t) = \frac{t}{10}.$

mult. by $e^{t/10}$: $\quad e^{t/10} Q' + \frac{1}{10} e^{t/10} Q = 4 e^{t/10}.$

integrate: $\quad e^{t/10} Q = 40 e^{t/10} + C$

$$\Rightarrow \quad Q(t) = 40 + C e^{-t/10}.$$

$Q(0) = 20 \quad \Rightarrow \quad 20 = 40 + C e^{0} \quad \Rightarrow \quad C = -20$

$$\Rightarrow \quad \underline{Q(t) = 40 - 20 e^{-t/10}}.$$

__Further questions:__

As $t \to \infty$, what happens to $Q(t)$?

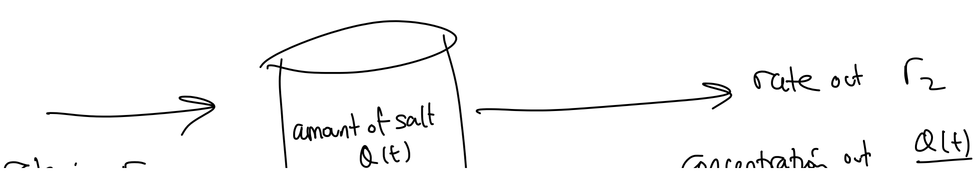
$\underline{A}$: $\quad 40 - 20 e^{-t/10} \longrightarrow 40 - 20(0) = 40.$

After what time will there be more than 30 lbs of salt in the tank?

$\underline{A}$: $\quad$ set $Q(t) = 30$ and solve for t.

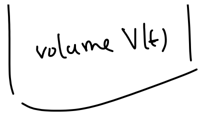
get $t = 10 \cdot \ln(2) \approx 6.9$ min.

General Case:



rate in r_1

Concentration in C_1



$V(t)$

initial data: $Q(0) = Q_0$, $V(0) = V_0$.

$$V' = r_1 - r_2 \implies V(t) = (r_1 - r_2)t + V_0.$$

$$Q' = \underset{\text{salt in}}{r_1 C_1} - \underset{\text{salt out}}{r_2\left(\frac{Q(t)}{V(t)}\right)} = r_1 C_1 - \frac{r_2 \, Q(t)}{(r_1 - r_2)t + V_0}.$$

Population Dynamics:

let $P(t)$ = population at time t.

look at
$$\frac{dP}{dt} = \underset{\text{growth rate}}{R(P)}\,P - \underset{\text{harvest rate.}}{h(t)}$$

Exponential model: take $h(t) = 0$, $R(P) = r$ = constant.

$$\frac{dP}{dt} = rP.$$

Solution is $P(t) = Ce^{rt}$.

with initial condition $P(0) = P_0$, get $P(t) = P_0 e^{rt}$.

eg. A population of monkeys starts at 100, and has a growth rate of 4% per year. 8 new monkeys join from surrounding tribes every year. Set up an IVP to model this system.

Sol'n: $M(t) = \#$ monkeys.

$M(0) = 100$

$$\frac{dM}{dt} = 0.04M + 8$$

(sol'n $M(t) = 300e^{0.04t} - 200$)

eg. A population of rabbits doubles in size every year. What is the growth rate, assuming exponential growth?

Sol'n: if $P' = rP \rightsquigarrow P(t) = P_0 e^{rt}$

$P(0) = P_0$

$P(1) = 2P_0 = P_0 e^{r \cdot 1}$

Solve for r: $2P_0 = P_0 e^r$

$\Rightarrow 2 = e^r \Rightarrow r = \ln 2$.

Logistic Model: models finite resources / competition,

$$\frac{dP}{dt} = (r - ap)p$$

$\hookrightarrow R(p) = r - ap$, a, r are positive

when P is small, $R(p) \approx r$

as P grows, $R(p)$ decreases.

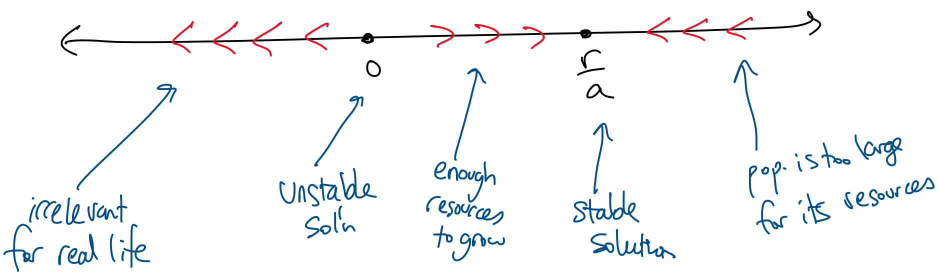
at $P = \frac{r}{a}$, $R(p) = 0$

↳ called the carrying capacity

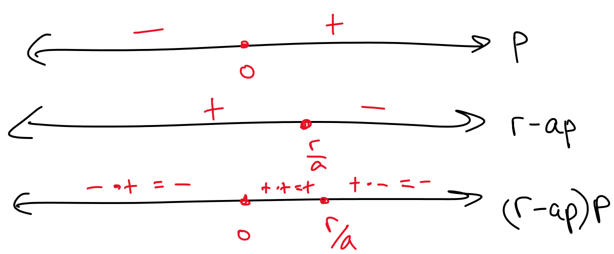
Stationary solutions: set $(r - ap)p = 0$

$$\Rightarrow \quad P = 0, \quad P = \frac{r}{a}.$$

phase-line portrait:



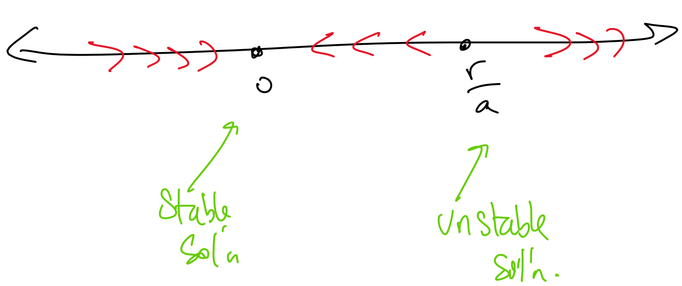
[Aside, for drawing phase-line portraits:



]

Variant logistic model:

$$\frac{dP}{dt} = -(r - ap)P.$$



this describes a context where a pop. needs a certain critical size $\left(\frac{r}{a}\right)$ to be able to grow, otherwise it dies out.

September 10 - I.6, I.7

Tuesday, September 10, 2024 10:57

Quiz 2 — on I.4 — I.6.3 (not motion)

I.6 (cont.)

Motion: falling objects:

object with mass m and force F acting on it

satisfies $F = ma$

$\hookrightarrow$ acceleration $= \frac{dv(t)}{dt}$, where $v(t) =$ velocity

axis for us: $\updownarrow$ $\uparrow$ positive, $\downarrow$ negative.

for falling objects, Force is composed of $F_{gravity}$ and F_{drag}

$\downarrow F_{grav}$

F_{grav} = force from gravity $= mg$, where $g = -9.8\ m/s^2$.

for us: F_{drag} = force from drag $= mCv^2$

So our ODE model for falling objects is:

$$m\frac{dv}{dt} = F = mg + mcv^2$$

C = <u>drag constant</u>

Cancel out m: $\quad v' = g + cv^2$

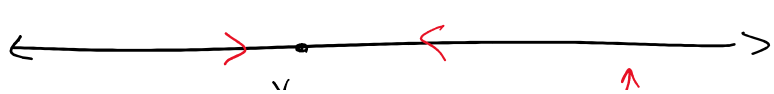
this non-linear, but it is separable.

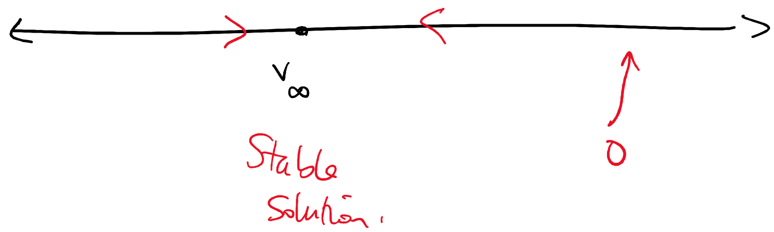
<u>Stationary solutions</u>: $\quad g + cv^2 = 0 \longrightarrow v = \pm\sqrt{\frac{-g}{c}}$

for falling objects, want neg. velocity.

So $v = -\sqrt{\frac{-g}{c}}$.

called <u>terminal velocity</u>, v_∞.





eg. A skydiver ~~of mass 60 kg~~ jumps from an airplane and assumes a position with drag coeff. $0.002\ m^{-1}$. What is her terminal velocity? setup the IVP for her velocity. (Solve).

Soln: $v(0) = 0$, $c = 0.002$

$$\Rightarrow \quad v' = g + 0.002\, v^2$$

terminal velocity $v_\infty = -\sqrt{\frac{-g}{c}} = -\sqrt{\frac{9.8}{0.002}} = -70\,m/s.$

rewrite ODE as
$$v' = c\left(\frac{g}{c} + v^2\right)$$
$$= c\left(v^2 - v_\infty^2\right)$$
$$= c\,(v - v_\infty)(v + v_\infty)$$

Solve:
$$\frac{1}{(v - v_\infty)(v + v_\infty)}\, v' = c$$

Integrate: $\frac{1}{\,}\left(\ln|v - v_\infty| - \ln|v + v_\infty|\right) = ct + C_0$

Integrate: $\frac{1}{2v_\infty}\left(\ln|v-v_\infty| - \ln|v+v_\infty|\right) = ct + C_0$

(partial fractions)

$C_0 \downarrow$ Constant of integration.

$v(0) = 0 \longrightarrow C_0 = 0.$

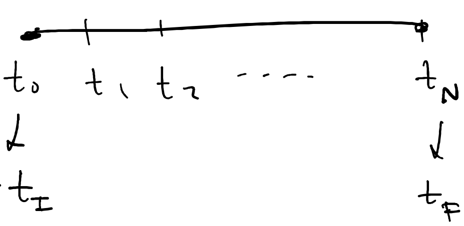
$$\Rightarrow \ln\left|\frac{v-v_\infty}{v+v_\infty}\right| = 2cv_\infty t$$

I.7 : Numerical Methods

Consider $y' = f(t,y)$, $y(t_I) = y_I$.

Say we want to know $y(t_F)$, where $t_F > t_I$.

break $[t_I, t_F]$ into



use $y(t_0) = y_I$ to approx. $y(t_1)$

use approx. of $y(t_1)$ to approx. $y(t_2)$

$\vdots \qquad\qquad \vdots$

use approx. of $y(t_{N-1})$ to approx. $y(t_N) = y(t_F)$

use approx. of $y(t_{N-1})$ to approx. $y(t_N) = y(t_F)$.

For good methods, the accuracy of the approximation increases as N increases.

We will use uniform step sizes:

$$\text{step size} = h = \frac{t_F - t_I}{N}.$$

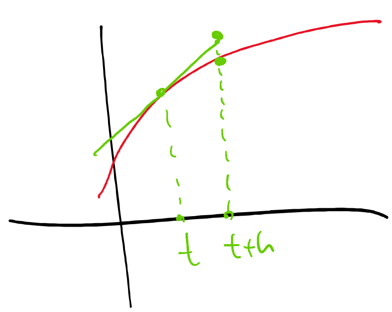
$$t_i = t_I + ih \quad \text{for } i = 0, \dots, N.$$

Euler Method:

Idea: $f'(t) = \lim\limits_{h \to 0} \dfrac{f(t+h) - f(t)}{h}$

for small h, $f'(t) \approx \dfrac{f(t+h) - f(t)}{h}$

$$\Rightarrow f(t+h) \approx f(t) + h f'(t)$$



if $Y(t)$ is the solution to $\underline{y' = f(t,y)}$, $y(t_I) = y_I$

$$Y(t_I + h) \approx Y(t_I) + h\,Y'(t_I)$$

$$\Rightarrow Y(t_1) \approx \underbrace{Y(t_0)}_{=Y_I} + h\,f(t_0, \underbrace{Y(t_0)}_{Y_I})$$

Then $Y(t_2) \approx \underbrace{Y(t_1)}_{\approx} + h\,f(t_1, \underbrace{Y(t_1)}_{\approx})$

etc.

<u>Algorithm</u>:

set $y_0 = y_I$

for $i = 1, \ldots, N$:

set $y_i = y_{i-1} + h\,f(t_{i-1}, y_{i-1})$

The $Y(t_F) = Y(t_N) \approx y_N$.

eg. Let $Y(t)$ be the solution to $y' = t^2 + y^2$, $y(0) = 1$.

Approximate $Y(0.2)$ with step size $h = 0.1$

Approximate $Y(0.2)$ with step size $h = 0.1$ using Euler method.

Sol'n: $t_0 = 0$, $t_1 = 0.1$, $t_2 = 0.2$

$y_0 = 1$

Then $y_1 = y_0 + h f(t_0, y_0)$

$= (1) + (0.1)\left((0)^2 + (1)^2\right) = 1.1$

and $y_2 = y_1 + h f(t_1, y_1)$

$= (1.1) + (0.1)\left((0.1)^2 + (1.1)^2\right)$

$= 1.1 + 0.1\left(0.01 + 1.21\right)$

$= 1.1 + 0.1(1.22) = 1.1 + 0.122 = 1.222.$

(in textbook, this is called <u>Explicit Euler Method</u>, we won't cover the implicit Euler method.)

Euler Method approximation comes from a Taylor series approx.

$$Y(t+h) = Y(t) + hY'(t) + h^2 Y''(t) + \cdots$$

$$y(t+h) = y(t) + h y'(t) + \underbrace{\frac{h^2}{2} y''(t) + \cdots}_{O(h^2)}$$

"order h^2"

$O(h^2)$ represents the local error, ie. the error at each step.

The global error / total error is $N \cdot O(h^2)$

But $N = \dfrac{t_F - t_I}{h} = \dfrac{\text{Constant}}{h}$

and $\dfrac{\text{Constant}}{h} \cdot O(h^2) = O(h)$

> Upshot: "Error is $O(h)$" tells us that if we scale h by a constant c, the error should also scale by c.

Higher-order Taylor series approximations:

eg. order 2:

$$Y(t+h) = Y(t) + hY'(t) + \frac{h^2}{2} Y''(t) + O(h^3)$$

So approx. $Y(t+h) \simeq Y(t) + hY'(t) + \frac{h^2}{2} Y''(t)$.

If $Y(t)$ is the sol'n to $y' = f(t,y)$

then $Y'(t) = f(t, Y)$

and $Y''(t) = \frac{d}{dt} f(t, Y(t))$

chain rule $= \frac{\partial f}{\partial t}(t, Y(t)) + \frac{\partial f}{\partial y}(t, Y(t)) \cdot \underbrace{Y'(t)}_{= f(t, Y(t))}$

$$\Rightarrow Y(t+h) \simeq Y(t) + hf(t, Y(t)) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t, Y(t)) + \frac{\partial f}{\partial y}(t, Y(t)) \cdot f(t, Y(t))\right).$$

This approximation method would have error of $O(h^2)$.

September 12 - I.7, I.9

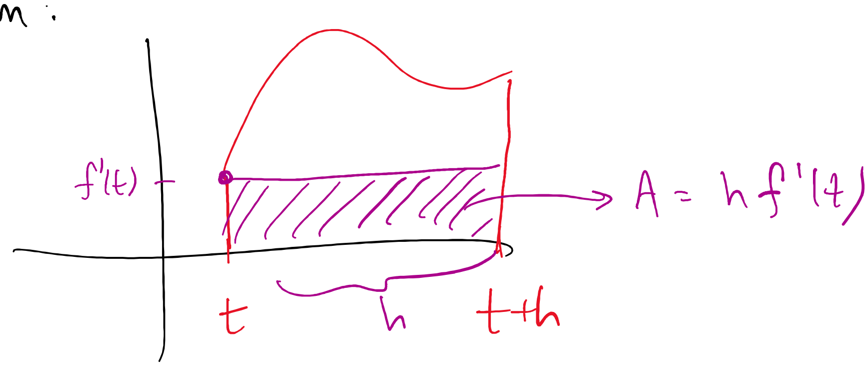
Thursday, September 12, 2024 10:59

I.7 (cont.) :

Recall : $f(t+h) - f(t) = \int_t^{t+h} f'(x)\,dx$

So $f(t+h) = f(t) + \underbrace{\int_t^{t+h} f'(x)\,dx}_{\text{need to approximate}}$

one way : left sum :



$\Longrightarrow f(t+h) \approx f(t) + hf'(t)$.

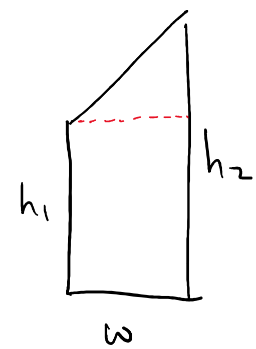
this gives the approximation that goes into the Euler Method

Other approximations of the integral :

Other approximations of the integral :

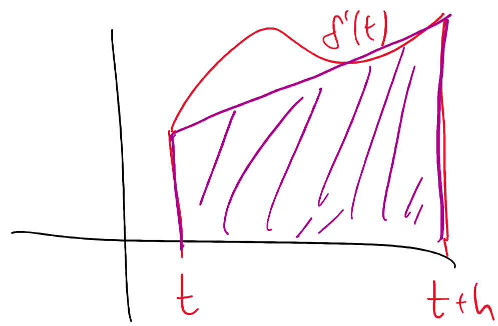
Trapezoid :

area of



$= w \cdot \left(\frac{h_1 + h_2}{2} \right)$

For $\displaystyle\int_t^{t+h} f'(t)\,dt$:



$$\approx h \left(\frac{f'(t) + f'(t+h)}{2} \right)$$

So $\quad f(t+h) \approx f(t) + \frac{h}{2}\left(f'(t) + f'(t+h) \right)$

Now say $Y(t)$ is the solution to $y' = f(t,y)$, $y(t_I) = y_I$

want $Y(t_F)$, Choose N steps, step size $h = \frac{t_F - y_I}{N}$.

$t_k = t_I + kh$

$y_0 = y_I$

$Y'(t_k) \qquad Y'(t_{k+1})$

$y_0 = y_I$

and $$y_{k+1} = y_k + \frac{h}{2}\Big(\underbrace{f(t_k, y_k)}_{Y(t_k)} + \underbrace{f(t_{k+1}, y_{k+1})}_{Y(t_{k+1})}\Big)$$

! problem

Can't use y_{k+1} to estimate y_{k+1}

replace this by

$y_k + hf(t_k, y_k)$

So we set $$y_{k+1} = y_k + \frac{h}{2}\Big(f(t_k, y_k) + f\big(t_{k+1}, y_k + hf(t_k, y_k)\big)\Big)$$

Turns out: global error of this method is $O(h^2)$

this is called <u>Runge-trapezoid method.</u>

eg. $y' = t^2 + y^2$, $y(0) = 1$

Estimate $y(0.2)$ using trapezoid method, $h = 0.1$

<u>Sol'n</u>: $f(t,y) = t^2 + y^2$. $h = 0.1$, $t_0 = 0$, $t_1 = 0.1$, $t_2 = 0.2$

$y_0 = 1$

$$y_1 = y_0 + \frac{h}{2}\Big(f(t_0, y_0) + f\big(t_1, y_0 + hf(t_0, y_0)\big)\Big)$$

$$= 1 + \frac{0.1}{2}\Big((0^2 + 1^2) + f\big(0.1, 1 + 0.1(0^2 + 1^2)\big)\Big)$$

$$= 1 + \frac{0.1}{2}\Big(1 + (0.1)^2 + (1.1)^2\Big)$$

$$= 1 + \frac{0.1}{2}\left(1 + (0.1)^2 + (1.1)^2\right)$$

$$= 1 + \frac{0.1}{2}\left(1 + 0.01 + 1.21\right)$$

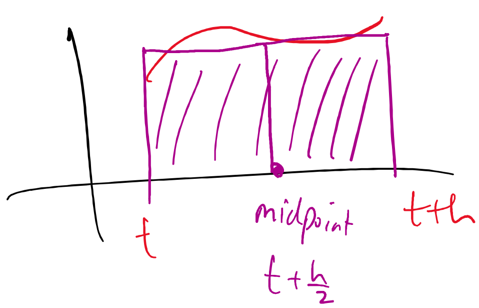
$$= 1 + \frac{0.1}{2}(2.22) = 1 + 0.111 = 1.111.$$

$$y_2 = y_1 + \frac{h}{2}\left(f(t_1, y_1) + f(t_2, y_1 + hf(t_1, y_1))\right)$$

$$= \dots$$

$$= 1.248.$$

Runge-Midpoint Method:



So we want to set $y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, \underline{y(t_k + \frac{h}{2})}\right)$

don't know this! approx. it using Euler Method

$\longrightarrow y_k + \frac{h}{2} f(t_k, y_k)$

So we set $y_{k+1} = y_k + h f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f(t_k, y_k)\right)$.

The global error is $O(h^2)$.

see textbook (p13) for an example.

I.9: Exact ODEs & Integrating Factors

$\frac{dy}{dx} = f(x,y)$. First-order ODEs.

When do we have an implicit solution $H(x,y) = C$?

Differentiate both sides with respect to x, remember $y = y(x)$:

$$\frac{\partial H}{\partial x} + \frac{\partial H}{\partial y} \cdot \frac{dy}{dx} = 0$$

So say we have an ODE:

$$M(x,y) + N(x,y)\frac{dy}{dx} = 0.$$

alternative notation: $M(x,y)\,dx + N(x,y)\,dy = 0$

<u>Q</u>: When can we find a function $H(x,y)$ with $\frac{\partial H}{\partial x} = M$, $\frac{\partial H}{\partial y} = N$?

If H has continuous second derivatives, then

$$\frac{\partial^2 H}{\partial x \partial y} = \frac{\partial^2 H}{\partial y \partial x}.$$

<u>A</u>: If domain of M and N has no "holes",

and $\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}$, then we can find a function $H(x,y)$ with $\frac{\partial H}{\partial x} = M$, $\frac{\partial H}{\partial y} = N$.

Such an ODE is called <u>exact</u>.

eg. Solve the IVP $\begin{cases} (e^x y + 2x) + (2y + e^x) y' = 0 \\ y(0) = 0. \end{cases}$

Sol'n: We have $M = e^x y + 2x$, $N = 2y + e^x$

Check: $\frac{\partial M}{\partial y} = e^x$, $\frac{\partial N}{\partial x} = e^x$

these are equal, so the ODE is exact.

So we want $H(x,y)$ so that $\begin{cases} \frac{\partial H}{\partial x} = e^x y + 2x & ① \\ \frac{\partial H}{\partial y} = 2y + e^x & ② \end{cases}$

① $\frac{\partial H}{\partial x} = e^x y + 2x$

$$\Rightarrow H(x,y) = \int e^x y + 2x \, dx$$

$$= e^x y + x^2 + C(y)$$

$$\frac{\partial H}{\partial y} = e^x + 0 + C'(y)$$

② $\frac{\partial H}{\partial y} = 2y + e^x$

$$\Rightarrow e^x + C'(y) = 2y + e^x$$

$$\Rightarrow C'(y) = 2y \quad \Rightarrow C(y) = y^2 \ (+C)$$

$$\Rightarrow \quad C'(y) = 2y \quad \Rightarrow \quad C(y) = y^2 \; (+C)$$

but choose $C=0$.

$$\Rightarrow \quad H(x,y) = e^x y + x^2 + y^2$$

So $e^x y + x^2 + y^2 = C$ is our general solution to the ODE.

$y(0) = 0 \longrightarrow e^0 \cdot 0 + 0^2 + 0^2 = C \Rightarrow C = 0$

So $e^x y + x^2 + y^2 = 0$ is our solution to the IVP.

Alt.: Start with ②: $\dfrac{\partial H}{\partial y} = 2y + e^x$

$$\Rightarrow H(x,y) = y^2 + e^x y + C(x)$$

plug $\dfrac{\partial H}{\partial x} = 0 + e^x y + C'(x)$ into ① and solve for $C(x)$ etc.

eg. Solve $(3t^2 y + 8ty^2)\,dt + (t^3 + 8t^2 y + 12y^2)\,dy = 0$

Sol'n: $M = 3t^2 y + 8ty^2$, $N = t^3 + 8t^2 y + 12y^2$

$\dfrac{\partial M}{\partial \ldots}$ $= 3t^2 \ldots$ $\quad \dfrac{\partial N}{\ldots}$ $= 3t^2 \ldots$

$$\frac{\partial M}{\partial y} = 3t^2 + 16ty \quad , \quad \frac{\partial N}{\partial t} = 3t^2 + 16ty + 0$$

$=$

So the ODE is exact.

We're looking for $H(t,y)$ with $\begin{cases} \frac{\partial H}{\partial t} = M & (1) \\ \frac{\partial H}{\partial y} = N & (2) \end{cases}$

(1) $\frac{\partial H}{\partial t} = M \Rightarrow H(t,y) = t^3 y + 4t^2y^2 + C(y)$

(2) $\frac{\partial H}{\partial y} = N \Rightarrow t^3 + 8t^2 y + C'(y) = t^3 + 8t^2 y + 12y^2$

$$\Rightarrow C'(y) = 12y^2 \Rightarrow C(y) = 4y^3$$

So $H(t,y) = t^3 y + 4t^2 y^2 + 4y^3$

and our general sol'n to the ODE is

$$t^3 y + 4t^2y^2 + 4y^3 = C.$$

MATH246 Lecture2

Shuo Liu

Sep4

I3

All the ODEs below are only involved in the first-order derivative of y.

I3

All the ODEs below are only involved in the first-order derivative of y.

1. An ODE of the form $y' = g(y)$ is called **autonomous**.

I3

All the ODEs below are only involved in the first-order derivative of y.

1. An ODE of the form $y' = g(y)$ is called **autonomous**.
2. A solution is called **stationary** if it is a constant.

I3

All the ODEs below are only involved in the first-order derivative of y.

1. An ODE of the form $y' = g(y)$ is called **autonomous**.
2. A solution is called **stationary** if it is a constant.
3. A general **separable** equation is of the form $y' = f(t)g(y)$

I3

All the ODEs below are only involved in the first-order derivative of y.

1. An ODE of the form $y' = g(y)$ is called **autonomous**.
2. A solution is called **stationary** if it is a constant.
3. A general **separable** equation is of the form $y' = f(t)g(y)$

Example

$y' + y^2 \cos t = 0$ where $y(0) = \frac{1}{2}$;

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

Theorem (Exisence and Uniqueness)

Let $f(t, y)$ be a function defined over a ***set S*** *in the ty-plane,*

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

Theorem (Exisence and Uniqueness)

Let $f(t, y)$ be a function defined over a ***set S*** *in the ty-plane,*

1. *f is continuous in S.*

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

Theorem (Exisence and Uniqueness)

Let $f(t, y)$ be a function defined over a ***set S*** *in the ty-plane,*

1. *f is continuous in S.*
2. *f is differentiable with respect to y over S.*

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

Theorem (Exisence and Uniqueness)

Let $f(t, y)$ be a function defined over a ***set S*** *in the ty-plane,*

1. *f is continuous in S.*
2. *f is differentiable with respect to y over S.*
3. *$\partial_y f$ is continuous over S.*

The initial value problem is of the form

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_1) = y_1 \end{cases}$$

Theorem (Exisence and Uniqueness)

Let $f(t, y)$ be a function defined over a ***set S*** *in the ty-plane,*

1. *f is continuous in S.*
2. *f is differentiable with respect to y over S.*
3. *$\partial_y f$ is continuous over S.*

Then for every initial point (t_I, y_I) is in the interior of S, there ***exists a unique*** *solution $y = Y(t)$ to initial-value problem that is defined over the* ***largest time interval (a, b)*** *such that*

- *t_I is in (a, b),*
- *$\{(t, Y(t)) : t \in (a, b)\}$ lies within the interior of S.*

Moreover, $Y'(t)$ is continuous over that interval.

Exercise

$$t\dot{y} = \sqrt{e^y - 1}, y(1) = 1$$

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$.

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1$.

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1.$

1. Identify all points where $g(y)$ is undefined. Plot these points on the interval (y_L, y_R) with $\circ$.

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1$.

1. Identify all points where $g(y)$ is undefined. Plot these points on the interval (y_L, y_R) with $\circ$.
2. Find all the zeros of g(y). Plot these zeroes on the interval (y_L, y_R) with $\bullet$.

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1$.

1. Identify all points where $g(y)$ is undefined. Plot these points on the interval (y_L, y_R) with $\circ$.
2. Find all the zeros of g(y). Plot these zeroes on the interval (y_L, y_R) with $\bullet$.
3. These undefined and stationary points partition (y_L, y_R) into subintervals. Determine the sign of g(y) on each of these subintervals

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1$.

1. Identify all points where $g(y)$ is undefined. Plot these points on the interval (y_L, y_R) with $\circ$.
2. Find all the zeros of g(y). Plot these zeroes on the interval (y_L, y_R) with $\bullet$.
3. These undefined and stationary points partition (y_L, y_R) into subintervals. Determine the sign of g(y) on each of these subintervals
   - Plot right arrows on each subinterval where g(y) is positive.

Phase-Line Portraits for Autonomous Equation

Autonomous equations are of the form $\frac{dy}{dt} = g(y)$. We have a guideline presenting the sign analysis of $g(y)$ on a graph of the interval (y_L, y_R).

Example

$\frac{dy}{dt} = y^2 - 1$.

1. Identify all points where $g(y)$ is undefined. Plot these points on the interval (y_L, y_R) with $\circ$.
2. Find all the zeros of g(y). Plot these zeroes on the interval (y_L, y_R) with $\bullet$.
3. These undefined and stationary points partition (y_L, y_R) into subintervals. Determine the sign of g(y) on each of these subintervals
   - Plot right arrows on each subinterval where g(y) is positive.
   - left arrows on each subinterval where g(y) is negative.

The resulting graph is called the phase portrait for the autonomous equation along the interval (y_L, y_R) of the **phase-line**.

The resulting graph is called the phase portrait for the autonomous equation along the interval (y_L, y_R) of the **phase-line**.

The arrows in the graph indicate the direction in which solutions of the equation will move along (y_L, y_R) as time increases. (See in the next exercise)

The resulting graph is called the phase portrait for the autonomous equation along the interval (y_L, y_R) of the **phase-line**.

The arrows in the graph indicate the direction in which solutions of the equation will move along (y_L, y_R) as time increases. (See in the next exercise)

Exercise

Describe the behavior of the solutions of the autonomous equations by drawing a phase-line portrait.

Suppose $f(t)$ is the solution to

$$\frac{dy}{dt} = (y-2)(y-1)^2(y+1)(y+3)^3$$

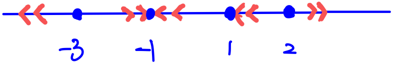
1. $f(0) = -2$, What is $\lim_{t\to\infty} f(t)$?
2. Same differential equation, but change the initial condition to $f(5) = 0$, what is $\lim_{t\to-\infty} f(t)$?
3. Change again to $f(-3) = 2$, what is $\lim_{t\to\infty} f(t)$?

$$g(y) = (y-2)(y-1)^2(y+1)(y+3)^3$$

- Zeros are -1,-3,1,2. They divide $\mathbb{R}$ into 5 disjoint small intervals. $(-\infty,-3),(-3,-1),(-1,1),(1,2),(2,\infty)$.
-

	$(-\infty,-3)$	(-3,-1)	(-1,1)	(1,2)	$(2,\infty)$
y-2	-	-	-	-	+
$(y-1)^2$	+	+	+	+	
y+1	-	-	+	+	+
y+3	-	+	+	+	+
g(y)	-	+	-	-	+

The phase portrait along $\mathbb{R}$ of the phase-line is



The portrait can be utilized to analyze the graph of solutions corresponding to different initial values. **The graph of the solution is bounded by the stationary points.**

1. $(-\infty, -1)$: The solution will be decreasing unboundedly. Backward the time $\lim_{t\to-\infty} f(t) = -1$.
2. $(-3, 1)$: The solution will decreasing and approaches to -3. So,

$$\lim_{t\to\infty} f(t) = -3, \lim_{t\to-\infty} f(t) = 1$$

Table of Contents


```
%Person A, Person B, Person C
%Section 101
%Matlab Project 1
```

Task 1

```
3+4
disp('We just computed a sum')
```

```
ans =

     7

We just computed a sum
```

Task 2

```
A=[1 2; 3 4; 5 6]

A  % Shows the matrix again
A; % Suppressed;  Matrix A is not shown again
```

```
A =

     1     2
     3     4
     5     6


A =

     1     2
     3     4
     5     6
```

Task 3

```
syms x
f(x)=sin(x)
```

```
f(pi)
disp('This shows that if we evaluate the sine function at pi, the value is 0')
```

```
f(x) =

sin(x)


ans =

0

This shows that if we evaluate the sine function at pi, the value is 0
```

Published with MATLAB® R2023a

MATH 246 – Matlab Project 1 – due September 30, 11:59pm

- **What to turn in:** You will need to write a script (.m file) and then "publish" it in Matlab as a PDF in order to turn it in on Gradescope.
- **What is a script?** A script is just a sequence of Matlab commands, with each command on its own line. If you click "Run", Matlab will execute all of the commands in the script, in order. The output will be produced in the Command Window, as if you entered each command individually. Your script should include all the necessary commands to solve the problems in the assignment. Scripts are useful because you can easily make edits to the commands if you make a mistake, or if you want to change something. To open a new script, click on 'New Script' in the top menu bar.
- **What is publishing?** When you "publish" a script, Matlab will produce a nice-looking document that displays all of the code in your script **as well as** the output produced by the commands after they were executed.
  - You may need to go into the options in Matlab to change the format of the published document from HTML to PDF. Please be sure to submit a PDF document. This can be done by going to Publish → Edit Publishing Options → Output File Format.
  - In the published document, Matlab displays ALL of the commands and then ALL of the output afterwards. For a long script, this results in something that is very difficult to read. Here is how you separate your code problem-by-problem in order to make it better: Before each numbered problem, type %% in the script to make a heading. This signifies to Matlab that it is a new portion of code, and Matlab separates it accordingly when published. Any text after the %% reads as a heading; remember to put a space between it and any text!

Do this	Not this
%% Problem 1	%%Problem 1

    After you publish, it should look like
    - Code for Problem 1
    - Output for Problem 1
    - Code for Problem 2
    - Output for Problem 2
    - ⋮
  - You will need to include some answers to questions that are not Matlab commands (e.g. you may have to explain why something happened). These questions are marked with a $\star$, and they require you to type an answer that is not just a Matlab command. You should have Matlab print these answers using the `disp` or `fprintf` commands in your script. You may need to manually add line breaks if your explanation is long. If the grader cannot read what you wrote because the text was chopped off, you won't get full credit!
  - You can see an example .m file and its output in the Files section on Canvas.
- **Rules for working in groups:** You are encouraged to collaborate on this project, if it is a true collaboration: no slackers allowed. You can't learn a programming language without typing it yourself. Projects may be turned in by groups of no more than three students, and in Gradescope for group submissions, there will be an option to include both group member's names. **If you don't include the names of all group members, not everyone will get credit.** It is possible to add names to the group even after submission.

Matlab Project 1

dsolve

One of the most basic Matlab commands is `dsolve`, which symbolically solves ordinary differential equations and sytems of differential equations. First we declare a symbolic function like $y(t)$:

```
syms y(t)
```

From here, the first derivative is `diff(y,1)` or just `diff(y)`, the second is `diff(y,2)`, and so on. We can apply `dsolve` to solve the differential equation $y' = y + t$ as follows. Note the use of double equal signs `==`:

```
dsolve(diff(y,1) == y + t)

ans =
C1*exp(t) - t - 1
```

If you run this, your `C` might have a different number. This is just a constant of integration.

TASK 1: Define the symbolic function $y(t)$, and use Matlab to solve the differential equation $y' = t^2 + \sin(t)$.

An initial condition may be put in put in simply as a second argument of `dsolve`, for example:

```
dsolve(diff(y,1) == y + t, y(1) == 2)
```

```
ans =
4*exp(-1)*exp(t) - t - 1
```

TASK 2: Use Matlab to solve the initial value problem $y' - 2ty = t$ with $y(-1) = 2$.

Many fairly simple differential equations have no explicit solutions, and we want to know what Matlab will do in these circumstances. For example, consider $(y^5 + 1)y' = \frac{1}{t} + t$, which we recognize as separable. With our first try, Matlab tells us that it can't find an explicit solution:

```
dsolve((y^5+1)*diff(y,1) == 1/t + t)
Warning:  Unable to find symbolic solution.
```

Instead, we have to tell Matlab that we are OK with an implicit solution:

```
dsolve((y^5+1)*diff(y,1) == 1/t + t, 'Implicit', true)

ans =
y(t) + y(t)^6/6 == C1 + log(t) + t^2/2
1/((y(t) + 1)*(y(t)^2 - y(t) - y(t)^3 + y(t)^4 + 1)) == 0
```

The first solution we recognize from trying to solve the separable equation:

$$(y^5 + 1)y' = \frac{1}{t} + t \Rightarrow \int y^5 + 1\, dy = \int \frac{1}{t} + t\, dt \Rightarrow \frac{1}{6}y^6 + y = \ln|t| + \frac{1}{2}t^2 + C.$$

The second is (once you multiply out) $\frac{1}{y^5+1} = 0$. Since this never happens (1 over something can never be 0), we can ignore this “solution”.

Note that if Matlab can't find a solution, even an implicit one, it will tell you so. Try:
`dsolve(exp(diff(y,1))==0,'Implicit',true)` (ie. solving $e^{y'} = 0$; exponentials can never equal 0).

TASK 3: Use Matlab to solve the differential equation $(1 + \frac{1}{y^2})y' = t$.

TASK 4: Use Matlab to solve the differential equation $(y')^2 + 1 = 0$.

TASK 5: Use Matlab to solve the initial value problem $y' + 2ty = t$ with $y(2) = -3$.

Plotting Pictures

Matlab is very good at graphing. For one solution, we can use `fplot`, as follows:

```
clear all
syms y(t)
fplot(dsolve(diff(y,1) == sin(t), y(0) == 2))
```

TASK 6: Use Matlab to plot the solution to the initial value problem $y' = y(y-6)$ with $y(0) = 1$.

To plot families of solutions, we need to do a few things. First observe that each new figure replaces the previous one unless we tell Matlab otherwise. We do so by using the commands `hold on` (to keep plotting on the current figure) and `hold off` (when we're done). We can do something like:

```
figure;
hold on
for v = [-5:5]
fplot(dsolve(diff(y,1) == 2*y, y(0) == v), [-1,3])
end
hold off
```

The `figure` command initializes a figure to be used for future plots, and `hold on` tells Matlab that we'll be drawing multiple items on this figure. The `for` loop starts by using the dummy variable `v`, and `[-5:5]` tells Matlab to run through values `v = -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5` (ie. step size 1; if we wanted a different step size, say 0.5, we would write `[-5:0.5:5]`). The `fplot` command is the same as above, except that the last argument `[-1,3]` tells Matlab to plot on the `t`-values $[-1, 3]$.

TASK 7: Use Matlab to plot the family of solutions to $y' = 0.1y(4-y)$ with $y(0) = -3, -2, \ldots, 7$.

Unfortunately, the way Matlab outputs implicit solutions means that we can't simply wrap the `dsolve` in an `fplot` like we do for explicit solutions. Instead, we solve the equation ourselves (ie. using `dsolve` in Matlab), and then plot the result using `fimplicit`, which is Matlab's tool for plotting implicit equations. For example, if we solve the initial value problem $(y^2 - y)y' = t$ with $y(2) = 1$ (it's separable), we get $2y^3 - 3y^2 = 3t^2 - 13$. We can plot this as:

```
syms y t;
fimplicit(2*y^3-3*y^2 == 3*t^2 - 13)
```

TASK 8: The initial value problem $(y^2 + 1)y' = \cos(t)$ with $y(0) = 1$ has solution $y^3 + 3y = 3\sin t + 4$. Use Matlab to plot this solution.

For a family of solutions, we need to find the general solution (without the initial condition), and then use different constants. The equation $(y^2 - y)y' = t$ has solution $2y^3 - 3y^2 = 3t^2 + 6C$, so for $C = -3, -2, -1, 0, 1, 2, 3$, we can do:

```
syms y t;
figure;
hold on
for C = [-3:3]
fimplicit(2*y^3-3*y^2 == 3*t^2 + 6*C)
end
hold off
```

TASK 9: The differential equation $(y^2 + 1)y' = \cos(t)$ with $y(0) = 1$ has solution $y^3 + 3y = 3\sin t + 3C$. Use Matlab to plot a family of solutions for $C = -10, -9, \ldots, 9, 10$.

Direction Fields

Plotting direction fields in Matlab is harder than simpler plots, because Matlab has no built in method for it. Here's how to do it for $y' = y - t^2$.

```
clear all
syms t y;
[T, Y] = meshgrid(-5:0.5:5, -5:0.5:5);
S = Y - T.^2;
L = sqrt(1 + S.^2);
quiver(T, Y, 1./L, S./L, 0.5)
```

Essentially, the `meshgrid` command creates a matrix of t and y values (between -5 and 5 in steps of size 0.5), and assigns them to `T` and `Y`. The `quiver` command draws an arrow starting at the coordinates given by the first two arguments (here `T` and `Y`), and in the direction given by the third and fourth arguments (here `1./L` and `S./L`). The `S` and `L` are as follows: if an arrow has slope S, then it should go 1 unit horizontally and S units vertically. Such an arrow has length $L = \sqrt{1 + S^2}$, and so to scale the arrow to have length 1, it should go $1/L$ units horizontally and S/L units vertically. The `0.5` argument of `quiver` scales everything down to half size.

The operations `./` and `.^` (and also `.*`) are required by Matlab because we're dealing with matrices, not numbers. Don't think too hard about it.

The only lines you'll have to change is the line defining `S` and the arguments of `meshgrid`.

TASK 10: Use Matlab to sketch the direction field for $y' = (1 - y)\sin t$ for integer t and y values between -10 and 10, with segments of length 0.5.

TASK 11: Produce another copy of the picture from TASK 10, and plot on top of it a reasonable family of solutions.

Numerical Methods

In order to apply Euler's Method to a differential equation of the form $y' = f(t, y)$, we need to know how to upt f into Matlab. We do this as follows for $y' = y^2y$:

```
syms f(t,y)
f(t,y) = t^2*y;
```

Now if we have the initial condition $y(1) = 2$, and we wish to do a single step of Euler's Method to approximate $y(1.1)$, we use the formula $y(1.1) \approx y(1) + 0.1 * y'(1) = 2 + 0.1 * f(1, 2)$, and we calculate:

```
2 + 0.1*f(1,2)

ans =
11/5
```

For a second step, to calculate $y(1.2)$, we can now use the output $y(1.1) \approx 11/5$. However, we'd like to automate this with a `for` loop. We start by telling Matlab our step size h, number of iterations n, starting values t and y (ie. t_0 and y_0 for $y(t_0) = y_0$). For $y' = t^2 y$ and $y(1) = 2$, with $n = 10$ steps of size $h = 0.1$, we can approximate $y(2)$ by:

```
syms f(t,y);
f(t,y) = t^2*y;
t = 1; y = 2; h = 0.1; n = 10;
for s = [1:n]
y = vpa(y + h*f(t,y));
t = t+h;
end
y
```

Instead of defining $t_1, t_2, \ldots$ and $y_1, y_2, \ldots$, we're just changing the value of t and y at every loop: y is set to the value of $y + hf(t, y)$ (using the previous values of y and t to calculate the next value of y), and then t is set to $t + h$. The command `vpa` (standing for variable precision arithmetic) makes sure that `y` is returned as a decimal.

TASK 12: Program Euler's Method into Matlab and use it to approximate $y(5)$ using 20 steps of size 0.1 from the initial value problem $y' = \frac{t^2+y}{y}$ with $y(3) = -1$.

TASK 13: Look up how to use `ode45` in Chapter 8 of the Matlab textbook or online, and plot the solution of $y' = t^2 + y^2$ with $y(0) = 1$ on the interval $[0, 0.5]$. Find the (approximate) value of $y(0.5)$.

MATH 246 – Matlab Project 2 – due November 4, 11:59pm

- **What to turn in:** You will need to write a script (.m file) and then "publish" it in Matlab as a PDF in order to turn it in on Gradescope.
- **What is a script?** A script is just a sequence of Matlab commands, with each command on its own line. If you click "Run", Matlab will execute all of the commands in the script, in order. The output will be produced in the Command Window, as if you entered each command individually. Your script should include all the necessary commands to solve the problems in the assignment. Scripts are useful because you can easily make edits to the commands if you make a mistake, or if you want to change something. To open a new script, click on 'New Script' in the top menu bar.
- **What is publishing?** When you "publish" a script, Matlab will produce a nice-looking document that displays all of the code in your script **as well as** the output produced by the commands after they were executed.
  - You may need to go into the options in Matlab to change the format of the published document from HTML to PDF. Please be sure to submit a PDF document. This can be done by going to Publish → Edit Publishing Options → Output File Format.
  - In the published document, Matlab displays ALL of the commands and then ALL of the output afterwards. For a long script, this results in something that is very difficult to read. Here is how you separate your code problem-by-problem in order to make it better: Before each numbered problem, type %% in the script to make a heading. This signifies to Matlab that it is a new portion of code, and Matlab separates it accordingly when published. Any text after the %% reads as a heading; remember to put a space between it and any text!

Do this	Not this
%% Problem 1	%%Problem 1

    After you publish, it should look like
    * Code for Problem 1
    * Output for Problem 1
    * Code for Problem 2
    * Output for Problem 2
    * ⋮
  - You will need to include some answers to questions that are not Matlab commands (e.g. you may have to explain why something happened). These questions are marked with a ⋆, and they require you to type an answer that is not just a Matlab command. You should have Matlab print these answers using the `disp` or `fprintf` commands in your script. You may need to manually add line breaks if your explanation is long. If the grader cannot read what you wrote because the text was chopped off, you won't get full credit!
  - You can see an example .m file and its output in the Files section on Canvas.
- **Rules for working in groups:** You are encouraged to collaborate on this project, if it is a true collaboration: no slackers allowed. You can't learn a programming language without typing it yourself. Projects may be turned in by groups of no more than three students, and in Gradescope for group submissions, there will be an option to include both group member's names. **If you don't include the names of all group members, not everyone will get credit.** It is possible to add names to the group even after submission.

Matlab Project 2

Matrices and Determinants

Matrices can be stored in Matlab using bracket notation with semicolons separating rows and either commas or spaces separating columns within those rows:

```
A = [1 2 3 ; -4 0 7]
```

```
A =
1 2 3
-4 0 7
```

If a matrix is square then the determinant can be found using the `det` command:

```
B = [1 -1 0 ; -7 -1 5 ; 8 1 0];
det(B)
```

```
ans =
   -45
```

TASK 1: Find the determinant of the matrix

$$\begin{pmatrix} 3 & 2 & 8 & -1 \\ 0 & -1 & -3 & 5 \\ 2 & 2 & -2 & 7 \\ 5 & 1 & 0 & 1 \end{pmatrix}. \tag{1}$$

Matrices can have symbolic variables in it and it will still work.

TASK 2: Declare the variable b as symbolic, and find the values of b where the determinant of the following matrix is zero using `det` wrapped in `solve`:

$$\begin{pmatrix} 3 & b & 2 \\ b & b & 5 \\ 1 & -1 & 5 \end{pmatrix}.$$

Factoring Polynomials

When hunting for fundamental sets we often have to factor the characteristic polynomial. Matlab can make this process easy once we understand what the output looks like. Here is the factorization of $x^3 - x^2 - 8x + 12$:

```
syms x
factor(x^3-x^2-8*x+12)
```

```
ans =
[x + 3, x - 2, x - 2]
```

We see that Matlab gives us the factors, with multiplicity if appropriate, as a list. If we want Matlab to work with complex numbers, we need to tell it explicitly:

```
factor(x^2+4,'FactorMode','Complex')
```

```
ans =
[x + 2.0i, x - 2.0i]
```

TASK 3: Factor the polynomial $x^2 - 7x + 10$.

TASK 4: Factor the polynomial $x^8 - 18x^6 - 4x^5 - 51x^4 + 612x^3 - 216x^2 + 648x - 4860$ and include complex terms.

Solving Higher Order Linear Differential Equations

Matlab can easily handle higher order linear differential equations. The key thing to know is to set the unknown function up properly and use the `dsolve` command. For a simple differential equation like $y'' - 3y' - 4y = 0$, we simply do something like:

```
syms y(t)
dsolve(diff(y,2)-3*diff(y)-4*y==0)
```

```
ans =
C1*exp(-t) + C2*exp(4*t)
```

If we want to give it some initial values it's notationally easier to prepare the derivatives before using `dsolve`, and it makes things look prettier. Notice below that Matlab will happily accept the definitions of `Dy` and `D2y`, even though `y` itself is symbolic and not known. Here's the above differential equation with the initial conditions $y(1) = -1$ and $y'(1) = 3$ added to make it an initial value problem.

```
syms y(t)
Dy = diff(y);
D2y = diff(y,2);
dsolve(D2y-3*Dy-4*y==0,y(1)==-1,Dy(1)==3)
```

```
ans =
(2*exp(4*t)*exp(-4))/5 - (7*exp(-t)*exp(1))/5
```

TASK 5: Solve the initial value problem $y'' - 3y' + 10y = 0$ with $y(0) = 1$ and $y'(0) = 17$.

TASK 6: Find the general solution to the differential equation $D^5y - 9D^4y + 36D^3y - 108D^2y + 243Dy - 243y = 0$.

It makes no difference to Matlab whether the system is nonhomogeneous.

TASK 7: Solve the differential equation $y'' - y' - 2y = \cos(t)$.

TASK 8: Solve the initial value problem $y''' + 3y'' - 4y' = t + \cos(t)$ with $y(0) = 0$ and $y'(0) = 1$ and $y''(0) = 3$.

Laplace Transforms

Matlab can calculate Laplace Transforms using the `laplace` command, as long as you understand exactly what to plug in. Basically we have to give it the function and both the "input" and "output" variables. We usually go from t to s and so we'd do something like this:

```
clear all
syms s t
laplace(exp(3*t),t,s)
```

```
ans =
1/(s - 3)
```

Here's a harder example, for a function that isn't in our table. It's wrapped in `simplify` to make it look nicer.

```
simplify(laplace(t^3*sin(t)*exp(3*t),t,s))
```

```
ans =
(24*(s^3 - 9*s^2 + 26*s - 24))/(s^2 - 6*s + 10)^4
```

TASK 9: Find and simplify $\mathcal{L}\left(t^3 \sin(2t)\right)$.

TASK 10: Find and simplify $\mathcal{L}\left(\left(2t^2 - t\right)\cos(5t)\right)$.

Matlab can also take inverse Laplace Transforms with the `ilaplace` command. For example suppose you know that $\mathcal{L}(y) = \frac{s+1}{s^2+9}$ and you want to find out what y is. Notationally, you could think of this as $y = \mathcal{L}^{-1}\left(\frac{s+1}{s^2+9}\right)$, and Matlab can compute this with:

```
ilaplace((s+1)/(s^2+9),s,t)
```

```
ans =
cos(3*t) + sin(3*t)/3
```

TASK 11: Find the function y which satisfies $\mathcal{L}(y) = \dfrac{1}{(s-2)^3}$.

TASK 12: Find the function y which satisfies $\mathcal{L}(y) = \dfrac{s^2+s+1}{s^3-5s^2-2s}$.

MATH 246 – Matlab Project 3 – due December 4, 11:59pm

- **What to turn in:** You will need to write a script (.m file) and then "publish" it in Matlab as a PDF in order to turn it in on Gradescope.
- **What is a script?** A script is just a sequence of Matlab commands, with each command on its own line. If you click "Run", Matlab will execute all of the commands in the script, in order. The output will be produced in the Command Window, as if you entered each command individually. Your script should include all the necessary commands to solve the problems in the assignment. Scripts are useful because you can easily make edits to the commands if you make a mistake, or if you want to change something. To open a new script, click on 'New Script' in the top menu bar.
- **What is publishing?** When you "publish" a script, Matlab will produce a nice-looking document that displays all of the code in your script **as well as** the output produced by the commands after they were executed.
  - You may need to go into the options in Matlab to change the format of the published document from HTML to PDF. Please be sure to submit a PDF document. This can be done by going to Publish → Edit Publishing Options → Output File Format.
  - In the published document, Matlab displays ALL of the commands and then ALL of the output afterwards. For a long script, this results in something that is very difficult to read. Here is how you separate your code problem-by-problem in order to make it better: Before each numbered problem, type %% in the script to make a heading. This signifies to Matlab that it is a new portion of code, and Matlab separates it accordingly when published. Any text after the %% reads as a heading; remember to put a space between it and any text!

Do this	Not this
%% Problem 1	%%Problem 1

    After you publish, it should look like
    - Code for Problem 1
    - Output for Problem 1
    - Code for Problem 2
    - Output for Problem 2
    - ⋮
  - You will need to include some answers to questions that are not Matlab commands (e.g. you may have to explain why something happened). These questions are marked with a ⋆, and they require you to type an answer that is not just a Matlab command. You should have Matlab print these answers using the `disp` or `fprintf` commands in your script. You may need to manually add line breaks if your explanation is long. If the grader cannot read what you wrote because the text was chopped off, you won't get full credit!
  - You can see an example .m file and its output in the Files section on Canvas.
- **Rules for working in groups:** You are encouraged to collaborate on this project, if it is a true collaboration: no slackers allowed. You can't learn a programming language without typing it yourself. Projects may be turned in by groups of no more than three students, and in Gradescope for group submissions, there will be an option to include both group member's names. **If you don't include the names of all group members, not everyone will get credit.** It is possible to add names to the group even after submission.

Matlab Project 3

More Matrices

We already learned how to store and take the determinant of matrices. You can multiply matrices `A` and `B` by inputting `A*B`.

TASK 1: Store the following matrices and calculate the products AB and BA.

$$A = \begin{pmatrix} 1 & 2 & -3 \\ 0 & 4 & 4 \\ -2 & 5 & 1 \end{pmatrix} \text{ and } B = \begin{pmatrix} -4 & 0 & 1 \\ 2 & 2 & -2 \\ 7 & -5 & 2 \end{pmatrix}.$$

For finding eigenvalues and eigenvectors it can be nicer to declare the matrix as symbolic. Matlab treats it a bit differently and will often return nicer answers. All we do is wrap the matrix in `sym`. Here we find the eigenvalues and eigenvectors of a 2×2 matrix:

```
C = sym([3 -1;4 -2]);
[evect,eval] = eig(C)
```

```
evect =
[ 1/4, 1]
[ 1, 1]
eval =
[ -1, 0]
[ 0, 2]
```

This is a bit confusing at first. The first part, `evect`, contains the eigenvectors as columns. The second part, `eval` contains the corresponding eigenvectors as entries in a diagonal matrix. So $\lambda = -1$ corresponds to $\begin{pmatrix} 1/4 \\ 1 \end{pmatrix}$ and $\lambda = 2$ corresponds to $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

TASK 2: Find the eigenvalues and eigenvectors of the matrix $\begin{pmatrix} 2 & -5 \\ 1 & -2 \end{pmatrix}$.

TASK 3: Find the eigenvalues and eigenvectors of the matrix $\begin{pmatrix} -1 & -1 & 0 \\ 5 & -3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$.

Solving Systems of Differential Equations

Matlab's `dsolve` command can solve systems of linear differential equations. For example if $x(t)$ and $y(t)$ satisfy the system

$$\begin{aligned} x' &= 2x + y, \\ y' &= x + 2y, \end{aligned}$$

then we can solve this in Matlab with:

```
syms x(t) y(t);
[xsoln,ysoln] = dsolve(diff(x) == 2*x+y, diff(y) == x+2*y)
```

```
xsoln =
C2*exp(3*t) - C1*exp(t)
ysoln =
C1*exp(t) + C2*exp(3*t)
```

Note that the solution for x must be the first of the two solutions, because `x` comes alphabetically before `y`.

TASK 4: Find the general solution of the system of differential equations: $x' = 3x - y$ and $y' = 4x - 2y$.

Matlab can also solve an initial value problem.

```
syms x(t) y(t);
[xsoln,ysoln] = dsolve(diff(x) == 2*x+y, diff(y) == x+2*y,x(0)==1,y(0)==-2)
```

```
xsoln =
(3*exp(t))/2 - exp(3*t)/2
ysoln =
- exp(3*t)/2 - (3*exp(t))/2
```

TASK 5: Find the solution to the differential equation from TASK 4 with initial conditions $x(0) = 2$ and $y(0) = -3$.

The system $x' = 2x+y$ and $y' = x+2y$ can be written in matrix form as $X' = AX$, where $X = X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ and $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$. We can put this into Matlab in this format:

```
syms x(t) y(t);
X=[x;y];
A = [2 1;1 2];
[xsoln,ysoln] = dsolve(diff(X) == A*X)
```

```
xsoln =
C2*exp(3*t) - C1*exp(t)
ysoln =
C1*exp(t) + C2*exp(3*t)
```

TASK 6: Solve the system of differential equations $x' = 2x - 5y$ and $y' = -x + 2y$ using matrix notation.

Non-Linear Systems

Matlab's `dsolve` often won't handle non-linear systems because most such systems don't have "nice" solutions. However Matlab's `ode45` command can give numerical (approximate) solutions if an initial value is given. By an approximate solution, we mean a collection of points whose graph is approximately a solution.

We'll try out `ode45` first on linear systems. Consider $x' = 2y$, $y' = -2x$ with $x(0) = 1$, $y(0) = 0$. The solution to this initial value problem is $x(t) = \cos(2t)$ and $y(t) = -\sin(2t)$, so if we plot this for t in $[0, \frac{\pi}{2}]$, we will get a semicircle.

To plot this in Matlab, we will use a single Matlab variable `x` that stands for the vector $\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$, and we will reference its components as `x(1)` and `x(2)`. Since $\mathbf{x}'(t) = \begin{pmatrix} 2y \\ -2x \end{pmatrix}$, we define the right-hand side as:

```
f = @(t,x) [2*x(2);-2*x(1)];
```

This is called a function handle: the letters after the `@` tells Matlab that what follows is a function of `t` and `x`. We now use `ode45`, passing it `f`, the range of t values, and the initial values (which is now a matrix $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$).

```
f = @(t,x) [2*x(2);-2*x(1)];
[t,xsoln] = ode45(f,[0 pi/2],[1,0]);
```

The output of `ode45` will be a vector of `t` values and a matrix of corresponding `x` values for our approximate solution (the first column corresponds to the approximate solution to $x(t)$, and the second to the approximate solution to $y(t)$). We plot `xsoln` to get a phase portrait of the approximate solution, which (after making the axes equal) looks like a semicircle!

```
f = @(t,x) [2*x(2);-2*x(1)];
[t,xsoln] = ode45(f,[0 pi/2],[1,0]);
plot(xsoln(:,1),xsoln(:,2))
axis equal
```

This method works for non-linear systems as well. For example, for $x' = 4y - y^3$, $y' = x$ with $x(0) = 1$, $y(0) = 0$, we get:

```
f = @(t,x) [4*x(2)-x(2)^3;x(1)];
[t,xsoln] = ode45(f,[0 2*pi],[1,0]);
plot(xsoln(:,1),xsoln(:,2))
axis equal
```

If you try this, you'll see that the ends of the curve don't meet up, even though they should. This is because `ode45` is only an approximation to the solution, and there is some error involved.

TASK 7: Plot an approximate solution to the initial value problem $x' = 4y - y^3$, $y' = x$ with $x(0) = 0$, $y(0) = 1$. By playing around with the t values for which you graph your solution, find an approximation of the period of the solution (that is, the smallest value $r\pi$ such that the solution starts looping around itself). Your answer should be $r\pi$, where r is a decimal that is correct up to two decimal places.

TASK 8: Plot an approximate solution for $0 \le t \le 4$ to the initial value problem $x' = 2 - y$, $y' = 4 - x^2$ for $x(0) = 0$, $y(0) = 6$.

We want to be able to plot phase portraits more generally, and we do that with the `quiver` command in Matlab. There is no built-in command for a phase portrait, unfortunately, but `quiver` will draw us vectors at an array of points with given directions: `quiver` takes four arguments, two for the base of the arrow, and two for the direction of the arrow. We need to get arrays of points using `meshgrid` (the input here is values from -1 to 1 with step size 0.05), and operations (multiplication, division, powers) using this arrays are written with an extra dot preceding the operation, so `.*`, `./`, and `.^` (don't ask why). Here is the code to plot the phase portrait of the system $x' = 2x - y$, $y' = 4x - xy$:

```
[x,y] = meshgrid(-1:0.05:1, -1:0.05:1);
xprime = 2*x-y;
yprime = 4*x - x.*y;
quiver(x,y,xprime,yprime)
axis equal
```

TASK 9: Plot a phase portrait for $-0.5 \le x, y \le 0.5$ for the system $x' = -x - 2x^2y + y$, $y' = -x - y$. Based on your plot, describe the stability of the stationary solution $x(t) = 0$, $y(t) = 0$.