BIOL 274 Homework 2

Ruby Krasnow

2024-02-01

Question 1

Suppose a species of fish in a particular lake has a population that is modeled by the Logistic Growth model with $r = 0.3, K = 2500$, and $P_0 = 2500$. Adjust the model and write down an IVP to represent each of the following situations.

a. One hundred fish are harvested each year.
b. One-third of the fish population is harvested annually.
c. The number of fish harvested each year is proportional to the square root of the number of fish in the lake.

We will assume that, in the absence of harvesting, the fish population at time t, $p(t)$, can be modeled by a discrete logistic growth equation with a time step of $\Delta t = 1$ year. Given an initial population size of $P(0) = P_0 = 2500$, we have the following IVP:

$$\begin{cases} \frac{dP}{dt} = 0.3P\left(1 - \frac{P}{2500}\right) \\ P(0) = 2500 \end{cases} \tag{1}$$

Note: Since we are working in discrete time, it might be more accurate to write the change in population as a difference equation, $P_{n+1} - P_n = \Delta P = 0.3P\left(1 - \frac{P}{2500}\right)$. For now we will continue using differential equation notation and apologize to Leibniz.

The initial conditions $P(0) = 2500$ will remain the same for each of the three situations, but we can adjust the differential equation for each scenario as follows:

a. One hundred fish are harvested each year:

$$\frac{dP}{dt} = 0.3P\left(1 - \frac{P}{2500}\right) - 100$$

b. One-third of the fish population is harvested annually:

$$\frac{dP}{dt} = 0.3P\left(1 - \frac{P}{2500}\right) - \frac{P}{3}$$

c. The number of fish harvested each year is proportional to the square root of the number of fish in the lake:

$$\frac{dP}{dt} = 0.3P\left(1 - \frac{P}{2500}\right) - \alpha\sqrt{P}$$

Where α is a constant of proportionality.

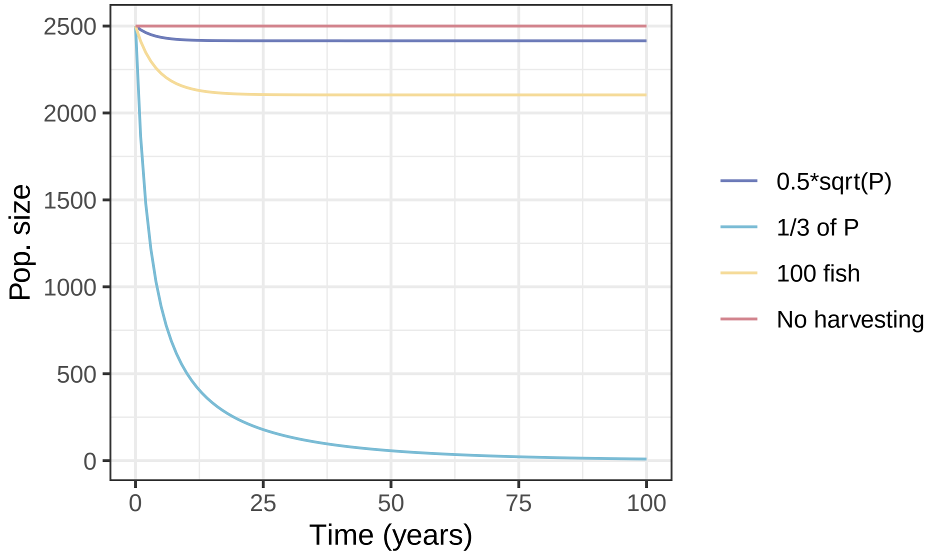
Let's see how the population would change over a 100-year period if no harvesting occurred and under each of these three harvesting strategies! We will use the numerical ODE solver from the `deSolve` package.

```r
library(deSolve) # best package for solving ODEs in R

#First we define our state variables
state_fish <- c(Po = 2500, #original - no harvest
                Pc = 2500, #constant harvest (100 fish/year)
                P3 = 2500, #1/3 of the fish are harvested every year
                Ps = 2500) #fish are harvested prop. to the square root of P

#How many time steps (years) do we want the model to run for?
times_fish <- seq(0, 100, 1)

#Provide values for our parameters. This makes it easy to identify
#how the population might change if we changed the carrying capacity
#or had a different harvesting intensity in scenario C.
params_fish <- c(K=2500, #carrying capacity
                 a=0.5) #constant that scales the harvest intensity in scenario C
```

```r
#Define the model structure and the differential equations
fish_model <- function(t, state, parameters) {

  with(as.list(c(state, parameters)), {
    dPo_dt <- 0.3*Po*(1-(Po/K)) #original - no harvest
    dPc_dt <- 0.3*Pc*(1-(Pc/K))-100 #constant harvest (100 fish/year)
    dP3_dt <- 0.3*P3*(1-(P3/K))-P3/3 #1/3 of the fish are harvested every year
    dPs_dt <- 0.3*Ps*(1-(Ps/K))-a*sqrt(Ps) #fish are harvested prop. to the square root of P

  #output
    return(list(c(dPo_dt, dPc_dt, dP3_dt, dPs_dt)))
  })
}

#Run the model
ode_output_fish <- ode(y = state_fish, t = times_fish, func = fish_model, parms = params_fish)

#Convert model output to a better format for visualization
fish_output <- ode_output_fish %>%
  as.data.frame() %>%
  pivot_longer(cols=2:5, names_to = "eqn", values_to = "P") %>%
  mutate(names_full = case_when(
    eqn =="Po" ~ "No harvesting",
    eqn =="Pc" ~ "100 fish",
    eqn =="P3" ~ "1/3 of P",
    eqn =="Ps"~"0.5*sqrt(P)")
  )
```

Let's see what we got!

```
ggplot()+
  geom_line(data=fish_output, aes(x=time, y=P, color=names_full))+
  theme_bw()+
  labs(x="Time (years)", y="Pop. size", color=NULL)+
  scale_color_manual(values=pnw_palette(name="Sailboat",n=4,type="discrete"))
```



```
#How many fish are left after 100 years?
fish_output %>% slice_tail(n=4)
```

```
## # A tibble: 4 x 4
##    time eqn         P names_full
##   <dbl> <chr>   <dbl> <chr>
## 1   100 Po    2500    No harvesting
## 2   100 Pc    2104.   100 fish
## 3   100 P3       9.21 1/3 of P
## 4   100 Ps    2415.   0.5*sqrt(P)
```

Cool! So in the original formulation, the population doesn't change because our initial conditions had P equal to our carrying capacity, so the $\left(1 - \frac{P}{K}\right)$ term was equal to zero and therefore $\frac{dP}{dt} = 0$.

Harvesting 1/3 of the population each year is not a sustainable management strategy, because we have decreased the population to almost zero.

Harvest scenario 3 actually did pretty well, and the impact on the population would be even smaller if we changed the values of α. If I was doing a whole project on these equations, it would be fun to test different values of K, P_0, and α and test the impacts on the population, dive into how the bifurcation points and stability of the solutions might change, and explore how we could use the model to identify an optimal harvest strategy.

Question 2

Using Separation of Variables, find a solution to the IVP representing the situation in #1b.

Recall that the IVP for the situation in #1b is:

$$\begin{cases} \frac{dP}{dt} = 0.3P\left(1-\frac{2500}{P}\right)-\frac{P}{3} \\ P(0) = 2500 \end{cases} \tag{2}$$

$$\begin{aligned}
\frac{dP}{dt} &= \frac{3P}{10}\left(1-\frac{P}{2500}\right)-\frac{P}{3} \\
dP\,\frac{1}{\frac{3P}{10}\left(1-\frac{P}{2500}\right)-\frac{P}{3}} &= dt \\
dP\,\frac{1}{\frac{3P}{10}-\frac{3P^2}{25000}-\frac{P}{3}} &= dt \\
dP\,\frac{1}{P\left(\frac{3}{10}-\frac{3P}{25000}-\frac{1}{3}\right)} &= dt \\
dP\,\frac{1}{P\left(\frac{9}{30}-\frac{3P}{25000}-\frac{10}{30}\right)} &= dt \\
dP\,\frac{1}{P\left(-\frac{3P}{25000}-\frac{1}{30}\right)} &= dt
\end{aligned}$$

Now we use partial fraction decomposition to rewrite the fraction, assuming there exist some A, B such that

$$\frac{1}{P\left(-\frac{3P}{25000}-\frac{1}{30}\right)} = \frac{A}{-\frac{3P}{25000}-\frac{1}{30}} + \frac{B}{P}$$

$$\begin{aligned}
AP + B\left(-\frac{3P}{25000}-\frac{1}{30}\right) &= 1 \\
AP - \frac{3BP}{25000} - \frac{B}{30} &= 1
\end{aligned}$$

$-\frac{B}{30} = 1$, so $B = -30$.

$$P\left(A-\frac{3B}{25000}\right) = P\left(A+\frac{9}{2500}\right) = 0$$

$A = -\frac{9}{2500}$, so now

$$\begin{gathered}
\frac{A}{-\frac{3P}{25000}-\frac{1}{30}} + \frac{B}{P} = \\
\frac{-9}{-\frac{3P}{10}-\frac{250}{3}} + \frac{-30}{P} \\
\left(\frac{9}{\frac{3P}{10}+\frac{250}{3}} - \frac{30}{P}\right) dP = dt \\
\left(\frac{9}{\frac{9P+2500}{30}} - \frac{30}{P}\right) dP = dt \\
270\int \frac{1}{9P+2500}dP - 30\int \frac{1}{P}\,dP = \int dt \\
30\ln\left(9P+2500\right) - 30\ln(P) = t + C_1 \\
\ln\left(9P+2500\right) - \ln(P) = \frac{t+C_1}{30}
\end{gathered}$$

$$\ln\left(\frac{9P+2500}{P}\right)=\frac{t+C_1}{30}$$
$$\frac{9P+2500}{P}=e^{\frac{t}{30}+\frac{C_1}{30}}=C_2e^{\frac{t}{30}}$$

$$9P+2500=PC_2e^{\frac{t}{30}}$$
$$9P-PC_2e^{\frac{t}{30}}=-2500$$
$$P\left(9-C_2e^{\frac{t}{30}}\right)=-2500$$

$$P=\frac{2500}{C_2e^{\frac{t}{30}}-9}$$

Now using our initial conditions,

$$2500=\frac{2500}{C_2-9}$$

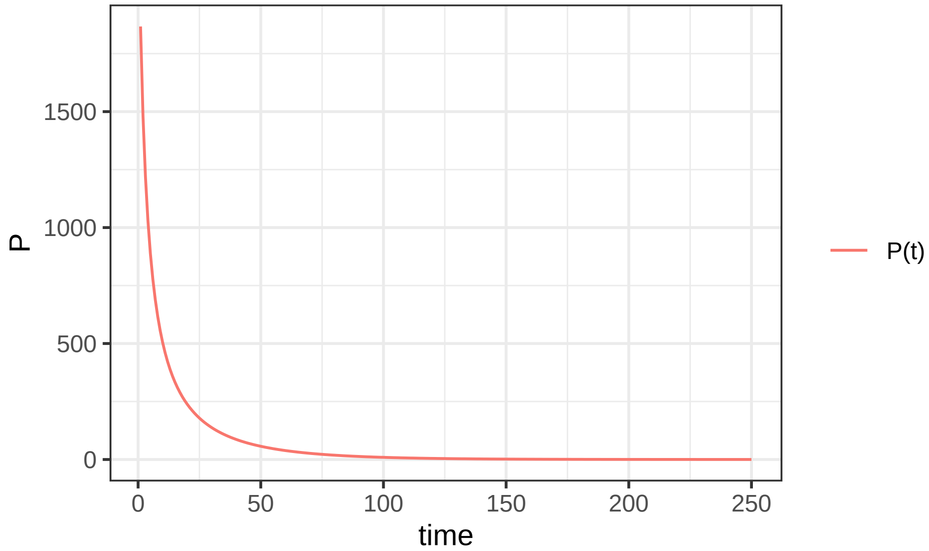
$C_2-9=1$, so $C_2=10$

Our final solution is therefore:

$$P(t)=\frac{2500}{10e^{\frac{t}{30}}-9} \tag{3}$$

Here is what it looks like:

```
t=c(1:250)
y=2500/(10*exp(t/30)-9)

ggplot()+
  geom_line(aes(x=t, y=y, color="P(t)"))+
  theme_bw()+
  labs(x="time", y="P", color=NULL)
```



Let's check and make sure it is actually a solution and satisfies our initial conditions:

We want to show that the derivative with respect to t of (3) is equivalent to $\frac{3P}{10}\left(1-\frac{P}{2500}\right)-\frac{P}{3}$.

Using the chain rule, we differentiate (3) to get:

$$P' = 2500(-1)\left(10e^{\frac{t}{30}} - 9\right)^{-2}\left(10e^{\frac{t}{30}}\right)\left(\frac{1}{30}\right) = \frac{-2500e^{\frac{t}{30}}}{3\left(10e^{\frac{t}{30}} - 9\right)^{2}}$$

Since there are no t's in the original $\frac{dP}{dt}$, we can use our solution to rewrite them as a function of P:

$$P\left(10e^{\frac{t}{30}} - 9\right) = 2500$$

$$10e^{\frac{t}{30}} = \frac{2500}{P} + 9$$

$$e^{\frac{t}{30}} = \frac{250}{P} + \frac{9}{10}$$

Now substituting this into our equation for P', we get

$$\frac{-2500\left(\frac{250}{P} + \frac{9}{10}\right)}{3\left(10\left(\frac{250}{P} + \frac{9}{10}\right) - 9\right)^2} = \frac{-2500\left(\frac{250}{P} + \frac{9}{10}\right)}{3\left(\frac{2500}{P} + 9 - 9\right)^2} = \frac{-2500\left(\frac{250}{P} + \frac{9}{10}\right)}{3\left(\frac{2500^2}{P^2}\right)} = \frac{\frac{-250}{P} - \frac{9}{10}}{3\left(\frac{2500}{P^2}\right)}$$

Now multiply the numerator and denominator by P:

$$\frac{-250 - \frac{9P}{10}}{\frac{7500}{P}} = \left(-250 - \frac{9P}{10}\right)\left(\frac{P}{7500}\right) = \frac{-250P}{7500} - \frac{9P^2}{75000} = P\left(-\frac{3P}{25000} - \frac{1}{30}\right)$$

And see that our initial $\frac{dP}{dt}$,

$$\frac{3P}{10}\left(1 - \frac{P}{2500}\right) - \frac{P}{3} = \frac{3P}{10} - \frac{3P^2}{2500} - \frac{P}{3} = P\left(-\frac{3P}{25000} - \frac{1}{30}\right)$$

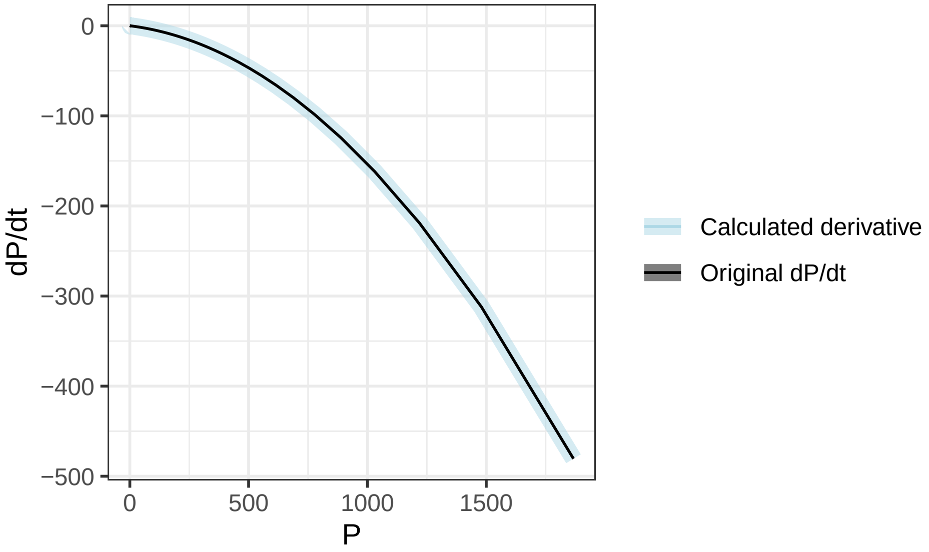
So indeed, our solution satisfies the differential equation.

And checking the initial conditions:

$$P(0) = \frac{2500}{10e^0 - 9} = \frac{2500}{1} = 2500$$

We can also confirm by using R to calculate the derivative of our solution and plotting it alongside the original expression for $\frac{dP}{dt}$:

```r
dy=D(expression(2500/(10*exp(t/30)-9)), "t")

ggplot()+
  geom_line(aes(x=y, y=eval(dy), color="Calculated derivative"), linewidth=3, alpha=0.5)+
  geom_line(aes(x=y, y= 0.3*y*(1-(y/2500))-y/3, color="Original dP/dt"))+
  scale_color_manual(values=c("lightblue", "black"))+
  theme_bw()+
  labs(x="P", y="dP/dt", color=NULL)
```



Question 2 again - for practice with Bernoulli's method

Using Bernoulli's method, find a solution to the IVP representing the situation in #1b.

$$\frac{dP}{dt} = \frac{3P}{10} - \frac{3P^2}{25000} - \frac{1P}{30}$$

Bernoulli's method review: If the differential equation is in the form $y'(x) + p(x)y = g(x)y^n$, we can divide by y^n and then use the substitution $z = y^{1-n}$.

Here,

$$P'(t) + \frac{1P}{30} = -\frac{3P^2}{25000}$$

Divide by P^2:

$$P'(P^{-2}) + \frac{1}{30P} = -\frac{3}{25000}$$

Use the substitution $z = P^{-1}$, $z' = -P^{-2}P'$:

$$-z' + \frac{z}{30} = -\frac{3}{25000}$$

$$z' - \frac{z}{30} = \frac{3}{25000}$$

$$\mu = e^{\int \frac{-1}{30} dt} = e^{\frac{-t}{30}}$$

$$\int \left(e^{\frac{-t}{30}} z\right)' dt = \int \frac{3e^{\frac{-t}{30}}}{25000} dt$$

$$e^{\frac{-t}{30}} z = \frac{-9e^{\frac{-t}{30}}}{2500} + C_1$$

$$z = \frac{-9}{2500} + \frac{C_1}{e^{\frac{-t}{30}}}$$

$$P = 1/z = \frac{1}{\frac{-9}{2500} + C_1 e^{\frac{t}{30}}}$$

$$P(0) = 2500 = \frac{1}{\frac{-9}{2500} + C_1}$$

$$2500\left(\frac{-9}{2500} + C_1\right) = 1$$

$$-9 + 2500C_1 = 1$$

Then $2500C_1 = 10$, so $C_1 = \frac{1}{250}$

$$P(t) = \frac{1}{\frac{-9}{2500} + \left(\frac{1}{250}\right) e^{\frac{t}{30}}} = \frac{2500}{10e^{\frac{t}{30}} - 9}$$

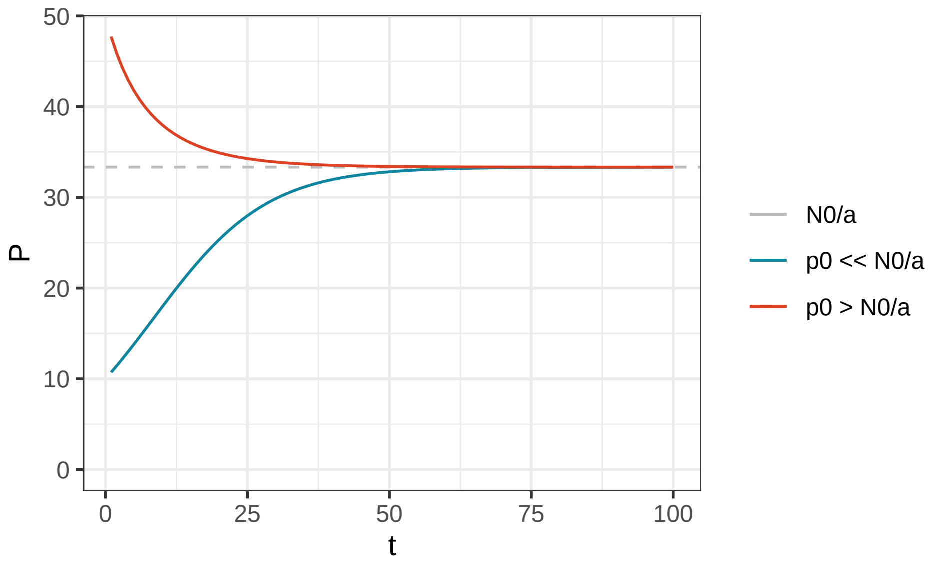
So we did get the same answer as when we used separation of variables.

Question 3

Figure 1.3.1 below is from your textbook. It shows an example of the logistic growth curve when $p(0) << N_0/a$. How would this curve change if $N_0 < aP(0)$?

$$P(t) = \frac{N_0 P_0}{aP_0 + (N_0 - aP_0)\, e^{-N_0 t}}$$

```
N0 = 0.1
a=0.003
p0_small=10
p0_big=50

x<-c(1:100)

ggplot()+
  geom_hline( linetype="dashed", aes(color="N0/a",yintercept=N0/a))+
  geom_line(aes(x=x, y=(N0*p0_small)/(a*p0_small+(N0-a*p0_small)*exp(-N0*x)), color="p0 << N0/a"))+
  geom_line(aes(x=x, y=(N0*p0_big)/(a*p0_big+(N0-a*p0_big)*exp(-N0*x)), color="p0 > N0/a"))+
  theme_bw()+
  labs(x="t", y="P", color=NULL)+
  scale_color_manual(values=c("gray",pnw_palette(name="Bay",n=2,type="discrete")))+
  ylim(0, NA)
```



If $N_0 < aP(0)$, that means that $N_0/a < P(0)$, or equivalently, $P(0) > N_0/a$. This is saying that our initial population size $P(0) = P_0$ is greater than our carrying capacity N_0/a, so the population size will decrease until it reaches the asymptote at $K = N_0/a$.

Euler's Method

For problems 4 and 5, use Euler's Method with the given step size, h, to approximate the solution to the given IVP over the time interval specified. Your answer should include a table of approximate values of the dependent variable. It should also include a sketch of the graph of the approximate solution.

Question 4

$$\frac{dw}{dt} = (3 - w)(w + 1), \quad w(0) = 4, \quad 0 \leq t \leq 5, \quad h = 1.0$$

We start with a table that looks like this:

```
data.frame(t=c(0:5), w=c(4, rep(NA, 5)))
```

```
##   t  w
## 1 0  4
## 2 1 NA
## 3 2 NA
## 4 3 NA
## 5 4 NA
## 6 5 NA
```

The equations needed to use Euler's method are:

$$t_{n+1} = t_n + h$$

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

where $f(t_n, y_n)$ is the slope we calculate by plugging t_n, y_n into our original DE.

Manual computation
At $t = 1, \quad w = 4 + (3 - 4)(4 + 1)1 = 4 + (-5) = -1$
But then for $t = 2, \quad w = -1 + (3 - -1)(-1 + 1)1 = -1 + 0 = -1$
For $t = 3, w = -1 + (3 - -1)(-1 + 1)1 = -1 + 0 = -1$
So once we hit $t = 2$, the slope is always 0, making $y(t_n + 1) = y(t_n) = -1$. The same thing will happen each time we compute w, so using Euler's method with a step size of h=1 has gotten us stuck!

Using a function
Let's define a function to compute Euler's method and see if we can do better with smaller step sizes.

Here we will write our own function, but note that the `deSolve` package includes a function `euler` that can do the same thing with many options for additional customization. There is also a simpler `euler` function in the package `demodelr` that takes fewer parameters.

```
euler_fun <- function(slope_fun, x0, y0, h, N) {
    x <- x0
    y <- y0
    num_steps <- N/h

    for(i in 1:num_steps) {
        y0 <- y0 + h * slope_fun(x0, y0)
        x0 <- x0 + h
        x <- c(x, x0)
        y <- c(y, y0)
    }

    return(data.frame(x = x, y = y, h=h))
}

f <- function(x_n, y_n){
  slope=(3-y_n)*(y_n+1)
  slope
}

euler_fun(slope_fun=f, x0=0, y0=4, h=1, N=5) %>% rename(w=y, t=x)
```

```
##   t  w h
## 1 0  4 1
## 2 1 -1 1
## 3 2 -1 1
## 4 3 -1 1
## 5 4 -1 1
## 6 5 -1 1
```

```
euler_fun(slope_fun=f, x0=0, y0=4, h=0.5, N=5) %>% rename(w=y, t=x)
```

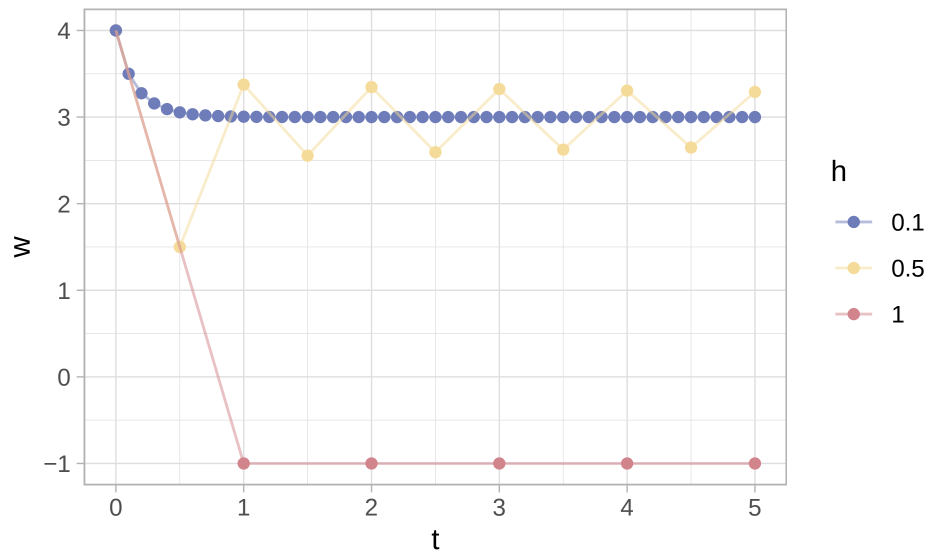
```
##      t        w   h
## 1  0.0 4.000000 0.5
## 2  0.5 1.500000 0.5
## 3  1.0 3.375000 0.5
## 4  1.5 2.554688 0.5
## 5  2.0 3.346161 0.5
## 6  2.5 2.593925 0.5
## 7  3.0 3.323626 0.5
## 8  3.5 2.624007 0.5
## 9  4.0 3.305308 0.5
## 10 4.5 2.648086 0.5
## 11 5.0 3.289992 0.5
```

```
euler_fun(slope_fun=f, x0=0, y0=4, h=0.1, N=5)%>% rename(w=y, t=x)
```

```
##      t        w   h
## 1  0.0 4.000000 0.1
## 2  0.1 3.500000 0.1
## 3  0.2 3.275000 0.1
## 4  0.3 3.157437 0.1
## 5  0.4 3.091984 0.1
## 6  0.5 3.054344 0.1
## 7  0.6 3.032311 0.1
## 8  0.7 3.019282 0.1
## 9  0.8 3.011532 0.1
## 10 0.9 3.006906 0.1
## 11 1.0 3.004139 0.1
## 12 1.1 3.002482 0.1
## 13 1.2 3.001488 0.1
## 14 1.3 3.000893 0.1
## 15 1.4 3.000536 0.1
## 16 1.5 3.000321 0.1
## 17 1.6 3.000193 0.1
## 18 1.7 3.000116 0.1
## 19 1.8 3.000069 0.1
## 20 1.9 3.000042 0.1
## 21 2.0 3.000025 0.1
## 22 2.1 3.000015 0.1
## 23 2.2 3.000009 0.1
## 24 2.3 3.000005 0.1
## 25 2.4 3.000003 0.1
## 26 2.5 3.000002 0.1
## 27 2.6 3.000001 0.1
## 28 2.7 3.000001 0.1
## 29 2.8 3.000000 0.1
## 30 2.9 3.000000 0.1
## 31 3.0 3.000000 0.1
## 32 3.1 3.000000 0.1
## 33 3.2 3.000000 0.1
## 34 3.3 3.000000 0.1
## 35 3.4 3.000000 0.1
## 36 3.5 3.000000 0.1
## 37 3.6 3.000000 0.1
## 38 3.7 3.000000 0.1
## 39 3.8 3.000000 0.1
## 40 3.9 3.000000 0.1
## 41 4.0 3.000000 0.1
## 42 4.1 3.000000 0.1
## 43 4.2 3.000000 0.1
## 44 4.3 3.000000 0.1
## 45 4.4 3.000000 0.1
## 46 4.5 3.000000 0.1
## 47 4.6 3.000000 0.1
## 48 4.7 3.000000 0.1
## 49 4.8 3.000000 0.1
## 50 4.9 3.000000 0.1
## 51 5.0 3.000000 0.1
```

```
euler_output<- c(1, 0.5, 0.1) %>%
  map(\(x) euler_fun(f, x0=0, y0=4, h=x, N=5)) %>%
  bind_rows()
```

```
ggplot(data = euler_output)+
  geom_point(aes(x=x, y=y, color=as.factor(h)))+
  geom_line(aes(x=x, y=y, color=as.factor(h)), alpha=0.5)+
  theme_light()+
  labs(x="t", y="w", color="h")+
  scale_color_manual(values=pnw_palette(name="Sailboat",n=3,type="discrete"))
```



Since this is a separable ODE and we are given initial conditions, we can also find the particular solution explicitly:

$$\frac{dw}{dt} = (3-w)(w+1) = -w^2 + 2w + 3$$

$$\int \frac{dw}{w^2 - 2w - 3} = \int -dt$$

Using partial fraction decomposition for the left-hand side, we can rewrite the integral as:

$$\frac{1}{4}\int \frac{1}{w-3} - \frac{1}{w+1}\, dw = \int -dt$$

$$\frac{1}{4}\left(\ln(w-3) - \ln(w+1)\right) = -t + C_1$$

$$\ln\left(\frac{w-3}{w+1}\right) = -4t + C_2$$

$$\frac{w-3}{w+1} = \frac{w+1-4}{w+1} = 1 + \frac{-4}{w+1} = C_3 e^{-4t}$$
$$\frac{-4}{w+1} = C_3 e^{-4t} - 1$$

Plugging in our initial conditions,

$$\frac{-4}{5} = C_3 - 1 \quad \longrightarrow \quad C_3 = \frac{1}{5}$$

$$\frac{-4}{w+1} = \frac{1}{5} e^{-4t} - 1$$

Solution

$$w(t) = \frac{4}{1 - \frac{1}{5} e^{-4t}} - 1$$

Check solution by differentiating

Direct differentiation of solution

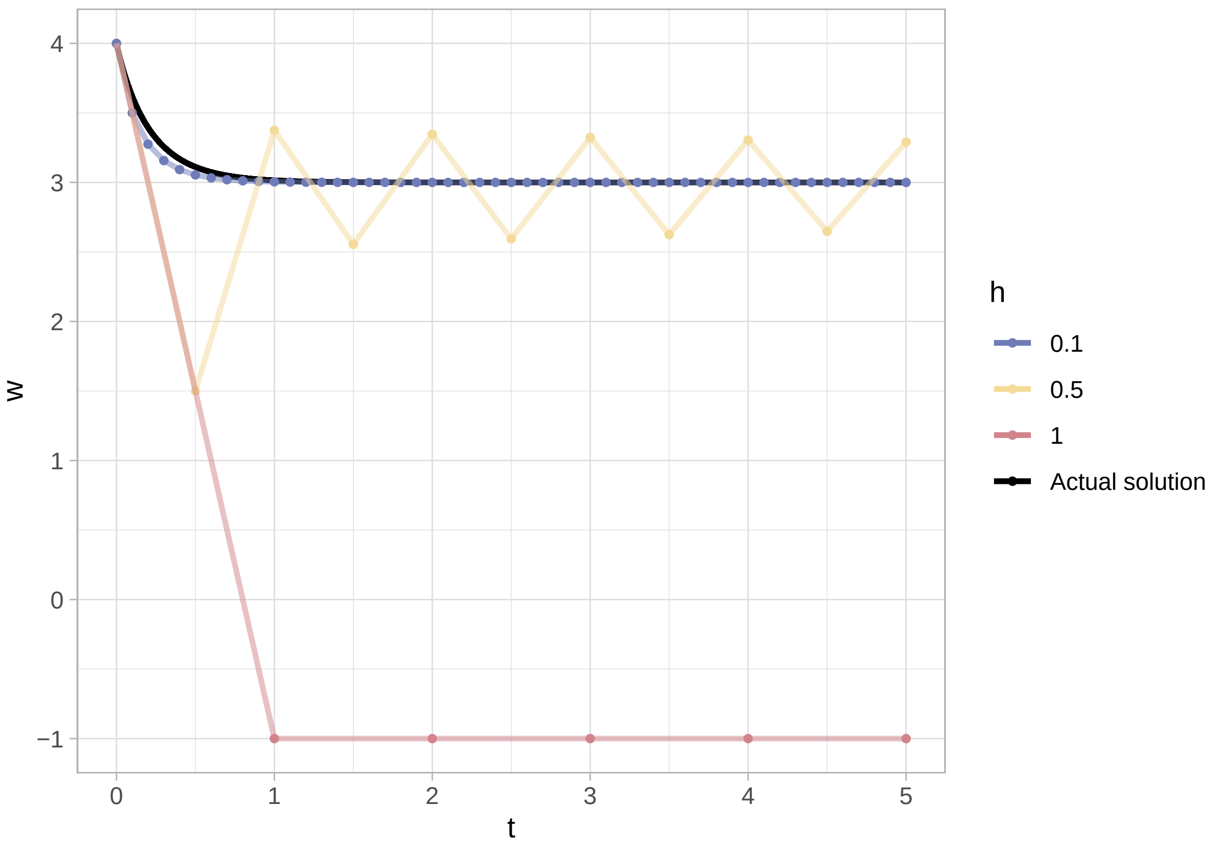
$$\frac{dw}{dt} = -4 \left(1 - \frac{1}{5} e^{-4t}\right)^{-2} \left(\frac{-1}{5} e^{-4t}\right) (-4) = \frac{16\left(-\frac{1}{5} e^{-4t}\right)}{\left(1 - \frac{1}{5} e^{-4t}\right)^2}$$

Plugging solution into original ODE

$$\frac{dw}{dt} = (3-w)(w+1) = \left(3 - \frac{4}{1-\frac{1}{5}e^{-4t}} + 1\right)\left(\frac{4}{1-\frac{1}{5}e^{-4t}} - 1 + 1\right) =$$
$$\left(4 - \frac{4}{1-\frac{1}{5}e^{-4t}}\right)\left(\frac{4}{1-\frac{1}{5}e^{-4t}}\right) =$$
$$\frac{16}{1-\frac{1}{5}e^{-4t}} - \frac{16}{\left(1-\frac{1}{5}e^{-4t}\right)^2} = \frac{16\left(1-\frac{1}{5}e^{-4t}\right) - 16}{\left(1-\frac{1}{5}e^{-4t}\right)^2} =$$
$$\frac{16\left(-\frac{1}{5}e^{-4t}\right)}{\left(1-\frac{1}{5}e^{-4t}\right)^2}$$

So now we can add to our graph the actual solution:

```
actual_eqn <- data.frame(t=seq(from=0, to=5, by=0.0001)) %>% mutate(
w=4/((1-(1/5)*exp(-4*t)))-1)

ggplot()+
  geom_line(data=actual_eqn, aes(x=t, y=w, color="Actual solution"), linewidth=1)+
  geom_point(data = euler_output, aes(x=x, y=y, color=as.factor(h)), size=1)+
  geom_line(data = euler_output, aes(x=x, y=y, color=as.factor(h)), alpha=0.5, linewidth=1)+
  theme_light()+
  labs(x="t", y="w", color="h")+
  scale_color_manual(values=c(pnw_palette(name="Sailboat",n=3,type="discrete"), "black"))
```



We can also see that $\lim_{t\to\infty} w(t) = 3$ by looking at our expression for the actual solution, because e^{-4t} will approach zero, so the denominator will approach $1 - 0$ and the overall solution will approach $\frac{4}{1} - 1 = 3$. Clearly, the step size of $h = 0.1$ very closely approximated the actual solution after around 30 time steps.

Question 5

$$\frac{dy}{dt} = e^{2/y}, \quad y(0) = 2, \quad 0 \leq t \leq 2, \quad h = 0.5$$

Manual computation

$(t_0 = 0, y_0 = 2) \quad \longrightarrow \quad \text{slope} = e^{2/2} = e, y_1 = 2 + \frac{e}{2} \approx 28.7644691$

```
y_1=round(2+exp(1)/2, 4)
y_1
```

```
## [1] 3.3591
```

$(t_1 = 0.5, y_1 \approx 3.3591) \quad \longrightarrow \quad \text{slope} = \exp\left(\frac{2}{3.3591}\right) \approx 1.8137519$

```
y_2=round(3.3591+exp(2/3.3591)/2, 4)
y_2
```

```
## [1] 4.266
```

$(t_2 = 1, y_2 \approx 4.266) \quad \longrightarrow \quad \text{slope} = \exp\left(\frac{2}{4.266}\right) \approx 1.5981125$

```
y_3=round(4.266+exp(2/4.266)/2, 4)
y_3
```

```
## [1] 5.0651
```

$(t_3 = 1.5, y_3 \approx 5.0651) \quad \longrightarrow \quad \text{slope} = \exp\left(\frac{2}{5.0651}\right) \approx 1.4841748$

```
y_4=round(5.0651+exp(2/5.0651)/2, 4)
y_4
```

```
## [1] 5.8072
```

$(t_4 = 2, y_4 \approx 5.8072)$

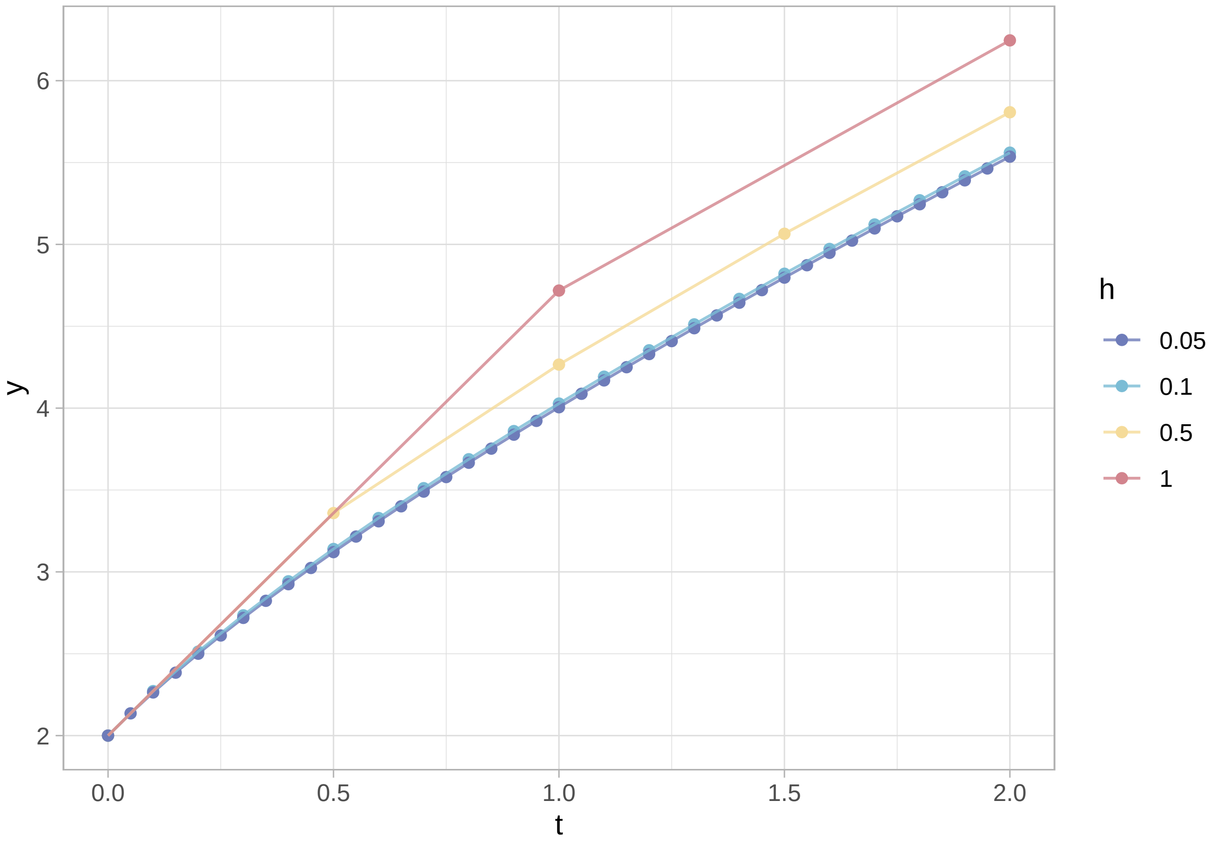
Using our function from earlier,

```
new_f = function(x_n, y_n) {
  slope = exp(2/y_n)
  slope
}

euler_fun(slope_fun = new_f, h=0.5, N=2, x0=0, y0=2)
```

```
##     x        y   h
## 1 0.0 2.000000 0.5
## 2 0.5 3.359141 0.5
## 3 1.0 4.266010 0.5
## 4 1.5 5.065066 0.5
## 5 2.0 5.807155 0.5
```

And we get the same result as our manual computation.

```
euler_output_2<- c(1, 0.5, 0.1, 0.05) %>%
  map(\(x) euler_fun(new_f, x0=0, y0=2, h=x, N=2)) %>%
  bind_rows()

ggplot()+
  geom_point(data = euler_output_2, aes(x=x, y=y, color=as.factor(h)))+
  geom_line(data = euler_output_2, aes(x=x, y=y, color=as.factor(h)), alpha=0.8)+
  theme_light()+
  labs(x="t", y="y", color="h")+
  scale_color_manual(values=pnw_palette(name="Sailboat",n=4,type="discrete"))
```


6
5
4
3
2
y
0.0
0.5
1.0
1.5
2.0
t
h
0.05
0.1
0.5
1