BIOL 274 Homework 8

Ruby Krasnow

2024-04-06

Visualizing Solutions of Linear Systems and Runge-Kutta

Question 1

Consider the epidemic model presented in class.

a.

For the parameters given, we found one single equilibrium point. Determine whether this equilibrium point is stable or unstable and sketch the phase plane near the equilibrium point.

Let X be the number of susceptible individuals at time t and Y be the number of infected individuals (who can transmit the disease). The epidemic model presented in class takes the following form:

$$\frac{dX}{dt} = -\mu XY + \rho$$

$$\frac{dY}{dt} = \mu XY - \nu Y$$

where $\mu = 0.025,\ \rho = 5$, and $\nu = 0.5$. We determined that given these parameters, the model has a single equilibrium point at $\left(\frac{\nu}{\mu}, \frac{\rho}{\nu}\right) = (20, 10)$.

We will determine the local stability of this equilibrium point by first computing the Jacobian matrix at $(20, 10)$.

Given a system of the type $\dot{x} = f(x,y), \quad \dot{y} = g(x,y)$, the Jacobian is defined as the following matrix:

$$J(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}$$

For our system, the Jacobian is

$$J(X,Y) = \begin{bmatrix} -\mu Y & -\mu X \\ \mu Y & -\mu X - \nu \end{bmatrix}$$

We then evaluate the Jacobian at our equilibrium point:

$$J(20,10) = \begin{bmatrix} -0.025(10) & -0.025(20) \\ 0.025(10) & 0.025(20) - 0.5 \end{bmatrix} = \begin{bmatrix} -0.25 & -0.5 \\ 0.25 & 0 \end{bmatrix}$$

Our next step is to find the eigenvalues of this matrix.

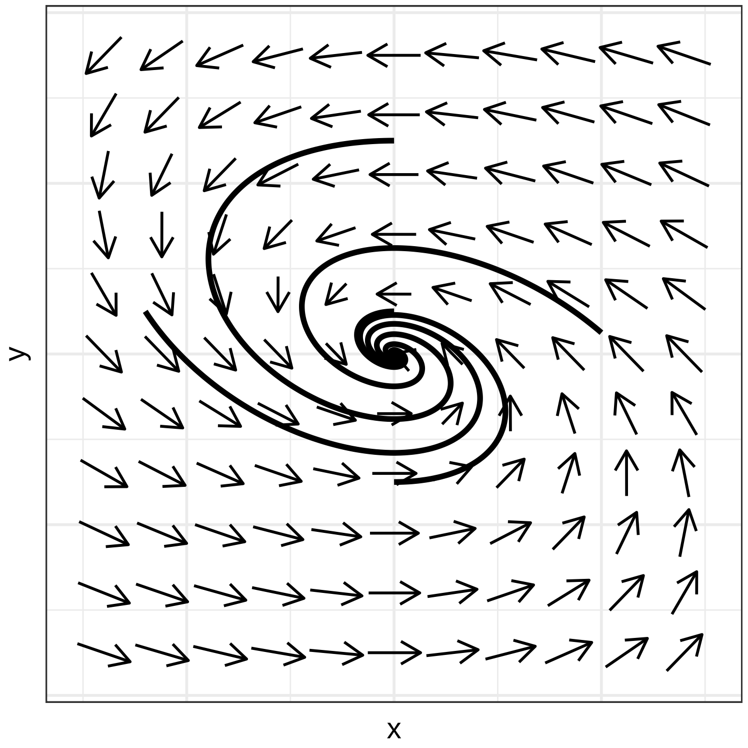
$$\det \begin{bmatrix} -0.25 - \lambda & -0.5 \\ 0.25 & 0 - \lambda \end{bmatrix} = 0$$

$$(-0.25-\lambda)(-\lambda)+0.125=0$$

$$\lambda^2+0.25\lambda+0.125=0$$

$$\frac{-b\pm\sqrt{b^2-4ac}}{2a} \quad\longrightarrow\quad \frac{-0.25\pm\sqrt{0.0625-4(0.125)}}{2}=\frac{-0.25\pm\sqrt{-4.375}}{2}\approx -0.125\pm 0.331\,i$$

We found complex eigenvalues with a negative real part, which means that we have a (stable) spiral sink. Near our equilibrium point, the phase portrait looks like:



b.

Modify the model given in class, assuming that the Susceptible population exhibits exponential growth with an intrinsic growth rate of 0.10 in the absence of any infected individuals.

The differential equation for the infected population will remain the same, so we will still have

$$\frac{dY}{dt}=\mu XY-\nu Y$$

However, we need to modify our DE for the susceptible population so that in the absence of infected individuals (i.e., when $Y=0$), the population will exhibit exponential growth. We set $r=0.1$ and write our new equation as

$$\frac{dX}{dt}=-\mu XY+rX$$

When $Y=0$, this becomes $\frac{dX}{dt}=rX$. We know this simple first-order ODE has the solution $X(t)=X_0e^{rt}$, meaning the population will grow exponentially over time with an intrinsic growth rate of 0.1, as desired.

c.

Find all equilibria of your new Epidemic model from part (b) and determine their stability by sketching the phase plane near each equilibrium.

To find the equilibrium points of our new epidemic model, we need to find all of the points where

$$\frac{dX}{dt} = -\mu XY + rX = 0 \tag{1}$$

$$\frac{dY}{dt} = \mu XY - \nu Y = 0 \tag{2}$$

Let's start with the derivative of Y, which is the same as when we analyzed it in class.

$$\frac{dY}{dt} = Y(\mu X - \nu) = 0$$

This means that either $Y = 0$ or $\mu X - \nu = 0$. But if $Y = 0$, we must have $\frac{dX}{dt} = 0 + rX = 0$, and since we defined $r = 0.1$, X must be zero and we have found the logical but boring equilibrium point $(0, 0)$.

The other possibility is $\mu X - \nu = 0$, which we can rearrange to be $X = \frac{\nu}{\mu}$. Plugging this into (1), we get

$$\frac{dX}{dt} = -\mu\frac{\nu}{\mu}Y + r\frac{\nu}{\mu} = 0$$
$$\frac{dX}{dt} = -\nu Y + r\frac{\nu}{\mu} = 0$$
$$\nu Y = r\frac{\nu}{\mu}$$
$$Y = \frac{r}{\mu}$$

Therefore, our second equilibrium point is

$$\left(\frac{\nu}{\mu}, \frac{r}{\mu}\right) = \left(\frac{0.5}{0.025}, \frac{0.1}{0.025}\right) = (20, 4)$$

Now we look at $\frac{dX}{dt} = -\mu XY + rX = X(-\mu Y + r) = 0$ Thus, either $X = 0$ or $-\mu Y + r = 0$. If $X = 0$, our equation (2) says that we must have $Y = 0$, so we found the same equilibrium point $(0, 0)$ as before. If $-\mu Y + r = 0 \quad \Rightarrow \quad \mu Y = r \quad \Rightarrow \quad Y = \frac{r}{\mu}$

$$\frac{dY}{dt} = \mu XY - \nu Y = 0$$

$$\frac{dY}{dt} = \mu X\frac{r}{\mu} - \nu\frac{r}{\mu} = 0$$
$$Xr - \nu\frac{r}{\mu} = 0$$

We can divide by r because it is a non-zero constant:

$$X - \frac{\nu}{\mu} = 0$$

So we again we have the equilibrium point

$$\left(\frac{\nu}{\mu}, \frac{r}{\mu}\right) = (20, 4)$$

For our new system, the Jacobian is

$$J(X,Y) = \begin{bmatrix} -\mu Y + r & -\mu X \\ \mu Y & -\mu X - \nu \end{bmatrix} = \begin{bmatrix} -\mu Y + r & -\mu X \\ \mu Y & -\mu X - \nu \end{bmatrix}$$

We then evaluate the Jacobian at our equilibrium point:

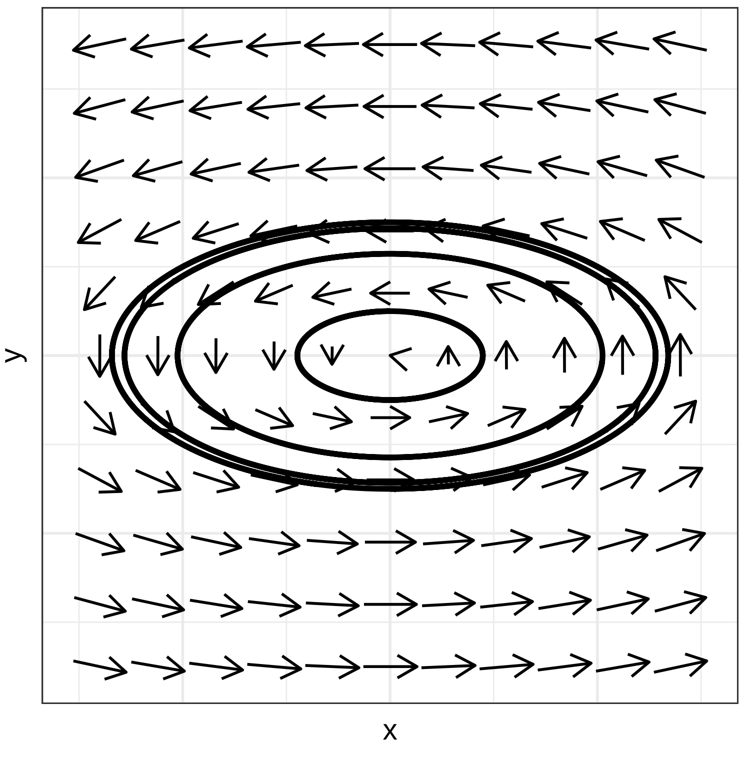
$$J(20,4) = \begin{bmatrix} -0.025(4) + 0.1 & -0.025(20) \\ 0.025(4) & 0.025(20) - 0.5 \end{bmatrix} = \begin{bmatrix} 0 & -0.5 \\ 0.1 & 0 \end{bmatrix}$$

Our next step is to find the eigenvalues of this matrix.

$$\det \begin{bmatrix} 0 - \lambda & -0.5 \\ 0.1 & 0 - \lambda \end{bmatrix} = 0$$

$$\lambda^2 + 0.05 = 0$$

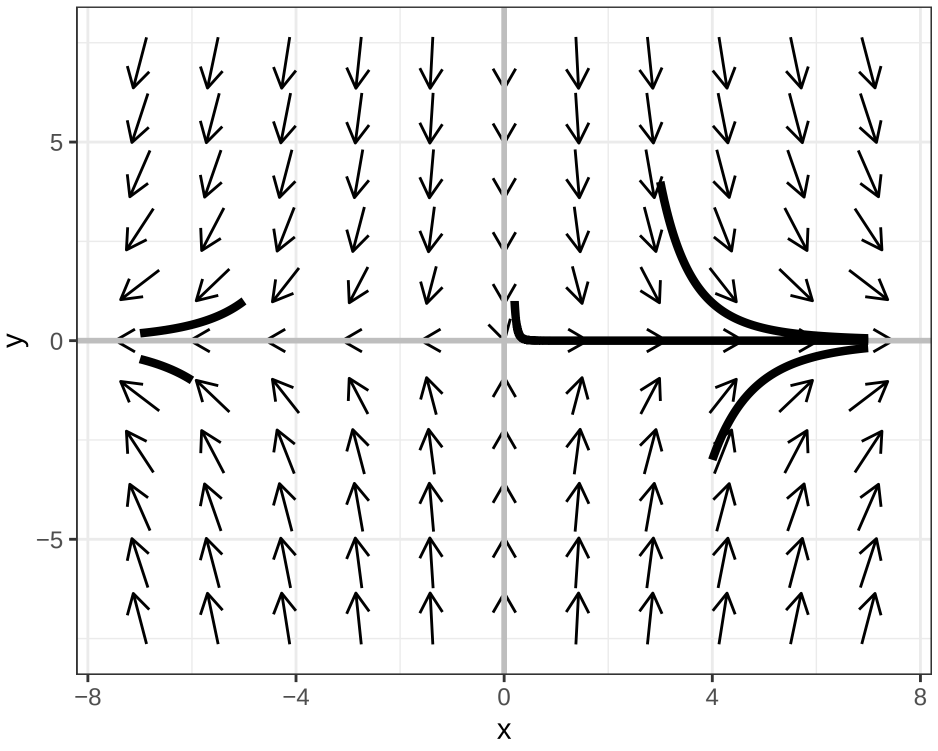
So $\lambda \approx \pm 0.223i$. Since we have purely imaginary complex eigenvalues, the phase plane will look like a center, corresponding to periodic solutions:



We then evaluate the Jacobian at our other equilibrium point, (0,0):

$$J(0,0) = \begin{bmatrix} -0.025(0) + 0.1 & -0.025(0) \\ 0.025(0) & 0.025(0) - 0.5 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & -0.5 \end{bmatrix}$$

We can see that our eigenvalues for this matrix are the values along the diagonal, $\lambda_1 = 0.1$, $\lambda_2 = -0.5$. Using our knowledge of linear algebra, we know that the eigenvector associated with λ_1 is just $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and the eigenvector associated with λ_2 is $\begin{bmatrix}1\\0\end{bmatrix}$. We have one positive and one negative eigenvalue, which means we will have an (unstable) saddle near the equilibrium at (0,0), with straight-line solutions tending away from the origin on the x-axis (because of the positive eigenvalue) and towards the origin on the y-axis (because of the negative eigenvalue).



Question 2

Consider the IVP from class:

$$\frac{dy}{dt} = -2ty^2, \quad y(0) = 1$$

over the interval $0 \le t \le 2$.

a.

Calculate the Runge-Kutta approximation to the solution with $n = 4$ steps.

Given an initial value problem $\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$, the Runge-Kutta method is as follows:

Pick a step-size $h > 0$ and define:

$$y_{n+1} = y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right),$$
$$t_{n+1} = t_n + h$$

for $n = 0, 1, 2, 3, ...$ using

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right), \\
k_3 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right), \\
k_4 &= f(t_n + h, y_n + hk_3)\,.
\end{aligned}$$

We will first compute the result manually using this process, then check our results with two different R packages:

```
manual<- tribble(
  ~val,  ~n_1,        ~n_2,          ~n_3,         ~n_4,
  "t",  0.5,        1,           1.5,         2,
  "k1", 0,         -0.63740945,-0.49940322,-0.28490054,
  "t_n+h/2",0.25,       0.75,        1.25,        1.75,
  "y_n+h*(k1)/2", 1,          0.6390269,  0.37485072, 0.23694178,
  "k2",-0.5,        -0.61253307,-0.35128265,-0.19649492,
  "y_n+h*(k2)/2", 0.875,      0.64524599, 0.41188086, 0.25904318,
  "k3",-0.3828125, -0.62451359,-0.42411461,-0.2348618,
  "y_n+h*(k3)", 0.80859375, 0.48612247, 0.28764422, 0.19073601,
  "k4",-0.65382385,-0.47263011,-0.24821759,-0.14552091,
  "y",  0.79837926, 0.49970152, 0.30816691, 0.20040567)

# from manual computation of intermediate values
as.data.frame(manual)
```

```
##              val        n_1        n_2        n_3        n_4
## 1              t  0.5000000  1.0000000  1.5000000  2.0000000
## 2             k1  0.0000000 -0.6374095 -0.4994032 -0.2849005
## 3        t_n+h/2  0.2500000  0.7500000  1.2500000  1.7500000
## 4   y_n+h*(k1)/2  1.0000000  0.6390269  0.3748507  0.2369418
## 5             k2 -0.5000000 -0.6125331 -0.3512826 -0.1964949
## 6   y_n+h*(k2)/2  0.8750000  0.6452460  0.4118809  0.2590432
## 7             k3 -0.3828125 -0.6245136 -0.4241146 -0.2348618
## 8     y_n+h*(k3)  0.8085938  0.4861225  0.2876442  0.1907360
## 9             k4 -0.6538238 -0.4726301 -0.2482176 -0.1455209
## 10             y  0.7983793  0.4997015  0.3081669  0.2004057
```

```
# from demodelr package
demodelr_out<- as.data.frame(demodelr::rk4(system_eq=c(dy ~ -2*t*(y^2)),
              initial_condition = c(y=1),
              deltaT = 0.5,
              n_steps = 5))
demodelr_out
```

```
##     t         y
## 1 0.0 1.0000000
## 2 0.5 0.7983793
## 3 1.0 0.4997015
## 4 1.5 0.3081669
## 5 2.0 0.2004057
```

```
# from deSolve paackage
rk_fun <- function(t, x, parms) {
  with(as.list(parms), {
    dx <- -1*a*t*(x[1])^2
    list(dx) }) }

time <- seq(from=0, to=2, length.out=5)
deSolve_out <- as.data.frame(deSolve::rk4(c(y=1), time, rk_fun, c(a=2)))
deSolve_out
```

```
##   time         y
## 1  0.0 1.0000000
## 2  0.5 0.7983793
## 3  1.0 0.4997015
## 4  1.5 0.3081669
## 5  2.0 0.2004057
```

In all three cases, we found $y(2) \approx 0.2004057$.

b.

Calculate the total error e_4 associated with this approximation.

We define the total error after k steps to be

$$e_k = |y(t_k) - y_k|$$

Since we know the true solution to our ODE is $y(t) = \frac{1}{1+t^2}$, we know that the value of the function at $t = 2$ should be $y(2) = \frac{1}{1+4} = 0.2$

```
rk_est <- deSolve_out %>% filter(time==2) %>% pull(y) #our rk4 approx.
abs(0.2-rk_est) #abs. difference between approx. and real values
```

```
## [1] 0.0004056722
```

Thus, RK4 with $\Delta t = 0.5$ gives us $e_4 = 4.057 \times 10^{-4}$.

c.

How many steps are necessary to approximate the solution with an error of less than 0.0001? Make sure to justify your answer.

In class, we found that RK4 with $\Delta t = 0.2$ gives us $e_{10} = 1.095 \times 10^{-5}$ and with $\Delta t = 0.1$ gives us $e_{20} = 6.541 \times 10^{-7}$.

We know that

$$e_n \leq C\Delta t \tag{3}$$

,

so given the error associated with two different time steps t_1 and t_2, we have the following relationship:

$$e_{n_1} \leq C(\Delta t_1)^4, \quad e_{n_2} \leq C(\Delta t_2)^4 \quad \implies \quad \frac{e_{n_1}}{e_{n_2}} \leq \left(\frac{\Delta t_1}{\Delta t_2}\right)^4 \quad \implies \quad \left(\frac{e_{n_1}}{e_{n_2}}\right)^{1/4} \leq \frac{\Delta t_1}{\Delta t_2}$$

Which we can rearrange as

$$\Delta t_1 \geq \left(\frac{e_{n_1}}{e_{n_2}}\right)^{1/4} \Delta t_2$$

Given an interval $0 \leq t \leq T$ and a known time step Δt_1, you need $n_1 = \frac{T}{\Delta t_1}$ time steps, so if we know the error e_{n_2} associated with a time step t_2, we can find the number of steps needed to approximate a target error e_{n_1} with the formula:

$$n_1 \geq \left(\frac{e_{n_1}}{e_{n_2}}\right)^{-1/4} \frac{T}{\Delta t_2}$$

We can write a function to do this in R:

```r
find_n <- function(e1, e2, t2, T) {
  out <- ((e1/e2)^(-1/4))*T*(1/t2)
  out
}
```

```r
e4<- 0.0004056722 # The error we calculated in part (b) above
e10<-1.095e-5     # From class slides
e20<- 6.541e-7    # From class slides

# Checking that our function works
test4 <- paste(round(find_n(e1=e4, e2=e10, t2=0.2, T=2),2),
      round(find_n(e1=e4, e2=e20, t2=0.1, T=2),2), sep=",")

test10<- paste(round(find_n(e1=e10, e2=e4, t2=0.5, T=2),2),
      round(find_n(e1=e10, e2=e20, t2=0.1, T=2),2), sep=",")

test20 <- paste(round(find_n(e1=e20, e2=e10, t2=0.2, T=2),2),
     round(find_n(e1=e20, e2=e4, t2=0.5, T=2),2), sep=",")

out <- paste(test4, test10, test20, sep="\n")
cat(out)
```

```
## 4.05,4.01
## 9.87,9.89
## 20.23,19.96
```

```r
en <- 0.0001 # Target error

# Applying the function to our target error and all three known error/time step combinations
out2 <- paste(find_n(e1=en, e2=e10, t2=0.2, T=2),
              find_n(e1=en, e2=e20, t2=0.1, T=2),
              find_n(e1=en, e2=e4, t2=0.5, T=2), sep="\n")
cat(out2)
```

```
## 5.75245897121819
## 5.68775699137658
## 5.67680273628098
```

Therefore, we would need 6 time steps to approximate our solution with an error of less than 0.001.

We can see that we obtain the same result by finding C directly and plugging in our values to (3):

```r
c1<- 0.0004056722/(0.5)^4 #0.006490755
c2 <- 1.095e-5/(0.2)^4 #0.00684375
c3 <- 6.541e-7/(0.1)^4 #0.006541
c<- mean(c(c1, c2, c3))
2/((en/c)^(1/4))
```

```
## [1] 5.705967
```