Module Code: CSMMS
Assignment report Title: ASSIGNMENT 1
Student Number: 32824514
Actual hours spent for the assignment: 20
Which Artificial Intelligence tools used (if applicable): None

Question 1: Cholesky Decomposition
Part A:

Apparently, $\boldsymbol{A}=\begin{pmatrix}3 & -1 & 0 & -1 & 0\\ -1 & 3 & -1 & 3 & -1\\ 0 & -1 & 3 & 1 & 0\\ -1 & 3 & 1 & 5 & -1\\ 0 & -1 & 0 & -1 & 1\end{pmatrix}=\boldsymbol{A}^T$, A is symmetric, positive definite matrices, which can be factorized into a product $\mathbf{A}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$, thus

$$\boldsymbol{A}=\begin{pmatrix}3 & -1 & 0 & -1 & 0\\ -1 & 3 & -1 & 3 & -1\\ 0 & -1 & 3 & 1 & 0\\ -1 & 3 & 1 & 5 & -1\\ 0 & -1 & 0 & -1 & 1\end{pmatrix} \tag{1.1}$$

$$=\boldsymbol{L}\boldsymbol{L}^T=\begin{pmatrix}l_{11} & 0 & 0 & 0 & 0\\ l_{21} & l_{22} & 0 & 0 & 0\\ l_{31} & l_{32} & l_{33} & 0 & 0\\ l_{41} & l_{42} & l_{43} & l_{44} & 0\\ l_{51} & l_{52} & l_{53} & l_{54} & l_{55}\end{pmatrix}\begin{pmatrix}l_{11} & l_{21} & l_{31} & l_{41} & l_{51}\\ 0 & l_{22} & l_{32} & l_{42} & l_{52}\\ 0 & 0 & l_{33} & l_{43} & l_{53}\\ 0 & 0 & 0 & l_{44} & l_{54}\\ 0 & 0 & 0 & 0 & l_{55}\end{pmatrix}$$

$$\begin{pmatrix}l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} & l_{11}l_{41} & l_{11}l_{51}\\ \nearrow & l_{21}^2+l_{22}^2 & l_{21}l_{31}+l_{22}l_{32} & l_{21}l_{41}+l_{22}l_{42} & l_{21}l_{51}+l_{22}l_{52}\\ \nearrow & \nearrow & l_{31}^2+l_{32}^2+l_{33}^2 & l_{31}l_{41}+l_{32}l_{42}+l_{33}l_{43} & l_{31}l_{51}+l_{32}l_{52}+l_{33}l_{53}\\ \nearrow & \nearrow & \nearrow & l_{41}^2+l_{42}^2+l_{43}^2+l_{44}^2 & l_{41}l_{51}+l_{42}l_{52}+l_{43}l_{53}+l_{44}l_{54}\\ \nearrow & \nearrow & \nearrow & \nearrow & l_{51}^2+l_{52}^2+l_{53}^2+l_{54}^2+l_{55}^2\end{pmatrix} \tag{1.2}$$

Some items in the matrix were omitted as it is symmetric.
Matching each item from (1.1) to (1.2) gives equations:

$$\begin{aligned} &l_{11}^2=3\\ &l_{11}l_{21}=-1\\ &l_{11}l_{31}=0\\ &\ldots\\ &l_{51}^2+l_{52}^2+l_{53}^2+l_{54}^2+l_{55}^2=1\end{aligned} \tag{1.3}$$

Though there are 15 equations and 15 variables in System of Equations (1.3), the solving process is simple as following the order of equations from top to bottom and using solved values. Solving equation group (1.3), supposing all square root are not negative $l_{11}\geq 0$, gives

$$\boldsymbol{L}=\begin{pmatrix}\sqrt{3} & 0 & 0 & 0 & 0\\ -\frac{\sqrt{3}}{3} & \frac{2\sqrt{6}}{3} & 0 & 0 & 0\\ 0 & -\frac{\sqrt{6}}{4} & \frac{\sqrt{42}}{4} & 0 & 0\\ -\frac{\sqrt{3}}{3} & \frac{2\sqrt{6}}{3} & \frac{4\sqrt{42}}{21} & \frac{\sqrt{210}}{21} & 0\\ 0 & -\frac{\sqrt{6}}{4} & -\frac{\sqrt{42}}{28} & \frac{\sqrt{210}}{35} & \frac{\sqrt{10}}{5}\end{pmatrix}$$

Part B:
The following R codes is Cholesky Decomposition function:

```
# 1.b Cholesky Decomposition
# 1.b.1 import required libraries
library(Matrix)
```

```
require(Matrix)
# 1.b.2 define CholeskyDecomposition function
CholeskyDecomposition <- function(A) {
  if (!isSymmetric.matrix(A)) {
    print('[ERROR] parameter A should be a n*n symmetric matrix')
    return()
  }
  n <- nrow(A)
  for (i in 1:n) {
    if (A[i, i] < 0) {
      print(sprintf('[ERROR] the leading minor of order %d is not positive', i))
      return()
    }
  }
  L <- Matrix(0, n, n)
  for (i in 1:n) {
    square <- A[i, i]
    if (1 <= i) {
      for (j in 1:i) {
        square <- square - (L[i, j]^2)
      }
    }
    if (square < 0) {
      print(sprintf('[ERROR] the leading minor of order %d is not positive', i))
      return()
    }
    L[i, i] <- sqrt(square)
    if ((i + 1) <= n) {
      for (j in (i + 1):n) {
        product <- A[i, j]
        if (1 <= (j - 1)) {
          for (k in 1:(j - 1)) {
            product <- product - L[i, k] * L[j, k]
          }
        }
        if ((L[i, i] == 0) & product != 0) {
          print(sprintf('[ERROR] the leading minor of order %d is not positive',
i))
          return()
        }
        L[j, i] <- product / L[i, i]
      }
    }
  }
```

```
  L
}

# 1.b.3 test answers in part(a)
CholeskyDecomposition(matrix(c(3, -1, 0, -1, 0,
                               -1, 3, -1, 3, -1,
                               0, -1, 3, 1, 0,
                               -1, 3, 1, 5, -1,
                               0, -1, 0, -1, 1), nrow = 5, byrow = TRUE))

# 5 x 5 sparse Matrix of class "dtCMatrix"
# [1,]  1.7320508  .          .         .         .
# [2,] -0.5773503  1.6329932  .         .         .
# [3,]  .         -0.6123724  1.620185 .         .
# [4,] -0.5773503  1.6329932  1.234427 0.6900656 .
# [5,]  .         -0.6123724 -0.231455 0.4140393 0.6324555
```

Question 2: Singular Value Decomposition

Part A:

Step1: Left-singular vectors as the eigenbasis of AA^T.

We start by computing

$$\boldsymbol{AA}^T = \begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.1}$$

The characteristic polynomial is

$$p_A(\lambda) = \det(\boldsymbol{AA}^T - \lambda \boldsymbol{I}) = \begin{vmatrix} 5-\lambda & 1 & 1 & 1 \\ 1 & 1-\lambda & 1 & 1 \\ 1 & 1 & 1-\lambda & 1 \\ 1 & 1 & 1 & 1-\lambda \end{vmatrix}$$

$$= \lambda^2(6-\lambda)(2-\lambda) \tag{2.2}$$

Giving the roots $\lambda_1 = 6$, $\lambda_2 = 2$ and $\lambda_3 = \lambda_4 = 0$, which are the eigenvalues of $\boldsymbol{AA}^T$.

For $\lambda_1 = 6$,

$$\boldsymbol{AA}^T\boldsymbol{x} = \begin{bmatrix} 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = 6 * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{2.3}$$

we find a unit eigenvector of $\boldsymbol{AA}^T$: $p_1 = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} \end{bmatrix}^T$.

Similarly, for $\lambda_2 = 2$, we get another unit eigenvector: $p_2 = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}^T$.

For $\lambda_3 = \lambda_4 = 0$, we obtain

$$\boldsymbol{AA}^T\boldsymbol{x} = \begin{bmatrix} 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \boldsymbol{0} \tag{2.4}$$

We solve this system and obtain $\boldsymbol{x} = \begin{bmatrix} 0 \\ x_2 \\ x_3 \\ -x_2 - x_3 \end{bmatrix}$, thus we construct two unit

orthogonal eigenvectors: $p_3 = \frac{1}{\sqrt{6}}[0 \quad 1 \quad 1 \quad -2]^T$and $p_4 = \frac{1}{\sqrt{2}}[0 \quad 1 \quad -1 \quad 0]^T$.

Thus, the singular values and left-singular vectors through the eigenvalue decompositions of $\boldsymbol{AA}^T$ is

$$\boldsymbol{AA}^T = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & 0 \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & -\frac{\sqrt{6}}{3} & 0 \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \end{bmatrix}$$

$$= \boldsymbol{PDP}^T \tag{2.5}$$

And we obtain the left-singular vectors as the columns of $\boldsymbol{P}$ so that

$$\boldsymbol{U} = \boldsymbol{P} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & 0 \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & -\frac{\sqrt{6}}{3} & 0 \end{bmatrix}. \tag{2.6}$$

Step 2: Singular-value matrix.

As the singular values σ_i are the square roots of the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$, we obtain them from D. Since rk($\boldsymbol{A}$) = 2, there are only two nonzero singular values: $\sigma_1 = \sqrt{6}$ and $\sigma_2 = \sqrt{2}$. The singular value matrix must be the same size as $\boldsymbol{A}$, and we obtain

$$\Sigma = \begin{bmatrix} \sqrt{6} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.7}$$

Step 3: Right-singular vectors as the eigenbasis of $\boldsymbol{A}^T\boldsymbol{A}$.

Similarly to step 2, we start by computing

$$\boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} \tag{2.8}$$

The characteristic polynomial is

$$p_A(\lambda) = \det(\boldsymbol{A}^T\boldsymbol{A} - \lambda\boldsymbol{I}) = \begin{vmatrix} 4-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix} = (6-\lambda)(2-\lambda) \tag{2.9}$$

Giving the roots $\lambda_1 = 6$ and $\lambda_2 = 2$, which are the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$.
We compute the singular values and right-singular vectors through the eigenvalue decompositions of $\boldsymbol{A}^T\boldsymbol{A}$, which is given as

$$\boldsymbol{A}^T\boldsymbol{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \boldsymbol{PDP}^T \tag{2.10}$$

And we obtain the right-singular vectors as the columns of P so that

$$\boldsymbol{V} = \boldsymbol{P} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \tag{2.11}$$

Thus,

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = U\Sigma\mathrm{V}^{\mathrm{T}} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & 0 \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{3}}{6} & \frac{1}{2} & -\frac{\sqrt{6}}{3} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{6} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Part B:
R codes show below:

```
# 2.b.1 import required libraries
library(Matrix)
require(Matrix)
# 2.b.2 compare
A2 <- matrix(c(1,2,
               1,0,
               1,0,
               1,0), nrow=4, byrow=TRUE)
svd(A2)
# $d
# [1] 2.449490 1.414214
# $u
#            [,1]     [,2]
# [1,] -0.8660254 0.5
# [2,] -0.2886751 -0.5
# [3,] -0.2886751 -0.5
# [4,] -0.2886751 -0.5
# $v
#            [,1]          [,2]
# [1,] -0.7071068   -0.7071068
# [2,] -0.7071068    0.7071068
```

Comparing to my answer in part (a), though eigenvectors u_1 and u_2 in U matrix are positive and negative opposites, both of them are right eigenvectors. The eigenvectors u_3 and u_4 in U matrix are ignored as they are meaningful for the calculations of eigenvectors, but not for the SVD. Actually, the elements of them always multiply by 0 of matrix Σ. Same to matrix U, eigenvectors in V matrix are positive and negative opposites.

Question 3: Differential Equations

Part A:

For differential equation

$$\frac{d^3y}{dt^3}+\frac{d^2y}{dt^2}-\frac{dy}{dt}-y=e^{-t}, \tag{3.1}$$

the characteristic equation is

$$r^3+r^2-r-1=0. \tag{3.2}$$

We solve by factoring:

$(r+1)^2(r-1)=0,$

then the characteristic function has roots $r_1=r_2=-1$, and $r_3=1$. The roots $r_1=r_2=-1$ contributes $c_1e^{-t}+c_2te^{-t}$ to the solution, while the root $r_3=1$ contributes c_3e^t. So a transient solution of the given differential equation is

$$y_t=c_1e^{-t}+c_2te^{-t}+c_3e^t \tag{3.3}$$

The function is $f=e^{-t}$, we should begin with a trial function y_s whose derivative involves e^{-t}. A reasonable guess is $y_s=Ae^{-t}$, but it is part of the transient solution in (3.3), as well as $y_s=Ate^{-t}$. Then we should try $y_s=At^2e^{-t}$, for which

$y_s'=2Ate^{-t}-At^2e^{-t},$

$y_s''=2Ae^{-t}-4Ate^{-t}+At^2e^{-t}=2Ae^{-t}-4Ate^{-t}+y_p,$

$y_s'''=-6Ae^{-t}+4Ate^{-t}+y_p'.$

Substitution into the original differential equation yields

$-6Ae^{-t}+4Ate^{-t}+2Ae^{-t}-4Ate^{-t}=-4Ae^{-t}=e^{-t}.$

So that $A=-\frac{1}{4}$.

Consequently, a steady solution is $y_s=-\frac{1}{4}t^2e^{-t}$. (3.4)

And the general solution is $y=y_t+y_s=c_1e^{-t}+c_2te^{-t}+c_3e^t-\frac{1}{4}t^2e^{-t}$. (3.5)

Then the given initial conditions yield the linear equations

$y(0)=\left(c_1e^{-t}+c_2te^{-t}+c_3e^t-\frac{1}{4}t^2e^{-t}\right)(0)=c_1+c_3=0,$

$y'(0)=\left((c_2-c_1)e^{-t}+c_3e^t-(c_2+\frac{1}{2})te^{-t}+\frac{1}{4}t^2e^{-t}\right)(0)=c_2-c_1+c_3=1,$

$y''(0)=\left((c_1-c_2)e^{-t}+c_3e^t-\left(c_2+\frac{1}{2}\right)e^{-t}+(c_2+1)te^{-t}-\frac{1}{4}t^2e^{-t}\right)(0)=$

$c_1-2c_2+c_3-\frac{1}{2}=-\frac{5}{2}$

in the coefficients c_1, c_2, and c_3. We solve the linear equations gives $c_1=c_3=0$, and $c_2 = 1$. Thus the desired particular solution is

$y = te^{-t} - \frac{1}{4}t^2e^{-t}$.

Part B:

R codes show below:

```
# 3.b Differential Equations
# 3.b.1 define initial conditions
initial.3 <- c(y=0, v=1, u=-2.5)
# 3.b.2 define differential equations
derivs.3 <- function (t,Ov,parms){
  with(as.list(c(Ov,parms)), {
    dy <- v
    dv <- u
    du <- exp(-t) + y + v - u
    list(c(dy, dv, du))
  })
}
# 3.b.3 define time interval
times <- seq(from = 0, to = 5, by = 0.001)
# 3.b.4 call ode function and get output
ode.out3 <- ode(y = initial.3, times = times, func = derivs.3, parms = NULL)
# 3.b.5 plot a curve for ode results
plot(ode.out3[,"time"], ode.out3[,"y"],
      type = "l", xlab="time", ylab="y", col="green", lwd=2)

# 3.b.6 Analytic solution
y3<-function(t) (t - 0.25 * t^2)*exp(-t)
curve(y3, 0, 5, lty=2, add=T, col="red")
```

The output images:

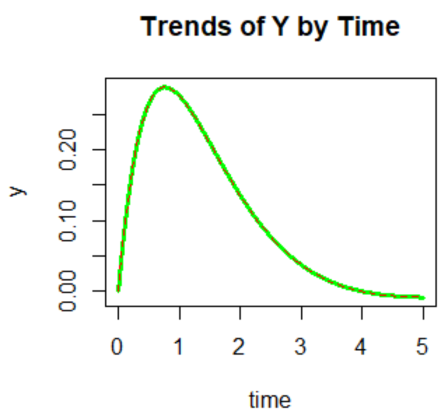


Figure 1

Question 4: Fish and Humans

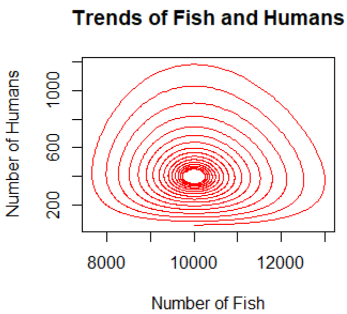
Part A:

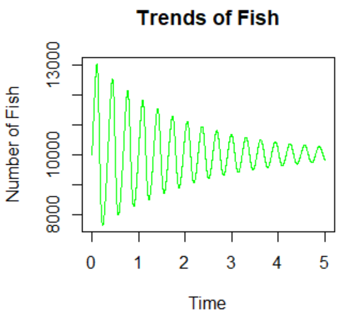
Part B:

R code shown below:

```
# 4.b Differential Equations
# 4.b.1 import required libraries
library(deSolve)
# 4.b.2 define initial conditions
initial.4 <- c(x=10000, y=60)
# 4.b.3 define parameters
parameters.4 <- c(a=5, b=0.01, c=100, d=0.01, g=0.0001)
# 4.b.4 define differential equations
derivs.4 <- function (t, Ov, params){
    with(as.list(c(Ov,params)), {
        dx <- a*x - g*x^2 - b*x*y
        dy <- -c*y + d*x*y
        list(c(dx, dy))
    })
}
# 4.b.5 define time interval
times.4 <- seq(from=0, to=5, by=0.001)
# 4.b.6 call ode function and get output
ode.out4 <- ode(y=initial.4, times=times.4, func=derivs.4, parms=parameters.4)
# 4.b.7 plot a curve for ode results
par(mfrow = c(1, 3))
plot(ode.out4[,"x"], ode.out4[,"y"], type="l",col="red",
          main='Trends of Fish and Humans', xlab="Number of Fish", ylab="Number
of Humans")
plot(ode.out4[,"time"], ode.out4[,"x"], type="l", col="green",
          main='Trends of Fish', xlab="Time", ylab="Number of Fish")
plot(ode.out4[,"time"], ode.out4[,"y"], type="l", col="blue",
          main='Trends of Humans', xlab="Time", ylab="Number of Humans")
```

The output images:





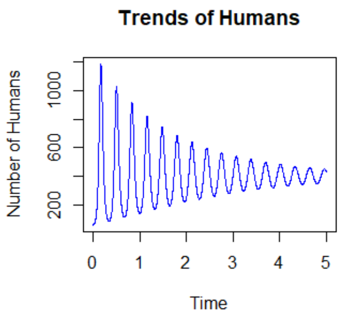


Figure 2 Figure 3 Figure 4

The Figure 2 shown that:

- At the beginning there were 60 people and 10000 fish. With the fewest predators, fish had the fastest reproduction rate and reached its peak at about 13000. Meanwhile, the number of humans also increased significantly to around 400.
- Then, having the most food and a relatively large population, the number of population still grown rapidly and reached its peak at about 1200. At the same time, with the growing number of predators, the number of fish jumped to 10000.
- With the largest population and decreasing food, the number of humans dropped to 400. And the number of fish decreased continually reached the shallow of lower 8000.
- Even worse, with the minimal food, the number of humans continually decreased to about 80 (bigger than the initial 60). Meanwhile, with less predators, the number of fish had a restorative growth to 10000.
- And then humans and fish re-enter the first step and the cycle begins again. But this time, with a larger population, the number of fish grown slower, as well as other changes were more gradual. Finally, the growths of fish and humans reached a relative equilibrium at about 10000 fish and 400 people.

Part C:

To find the equilibrium, let's consider the critical points of the equations:

$$5 * x - 0.0001 * x^2 - 0.01 * x * y = x * (5 - 0.0001 * x - 0.01 * y) = 0 \quad (4.1)$$
$$-100 * y + 0.01 * x * y = y * (-100 + 0.01 * x) = 0 \ .$$

A critical point (x, y) must satisfy that either

$$x = 0 \ or \ 5 - 0.0001 * x - 0.01 * y = 0 \quad (4.2.a)$$

And either

$$y = 0 \ or - 100 + 0.01 * x = 0 \quad (4.2.b)$$

If $x = 0$ and $y \neq 0$, then the second equation in (4.2.b) gives $x = 10000$, which is contradictory. If $y = 0$ and $x \neq 0$, then the second equation in (4.2.a) gives $x = 50000$. If $x \neq 0$ and $y \neq 0$, then we solve the system of equations:

$$5 - 0.0001 * x - 0.01 * y = 0, \ -100 + 0.01 * x = 0$$

for $x = 10000, y = 400$. There are 3 critical points, (0, 0), (50000, 0) and (10000, 400), for the equations. For the given scenario, the system achieves equilibrium at $x = 10000, y = 400$.

Part D:

R code shows below:

```
# 4.d find the minimum value of b, at which the human population dies out
# 4.d.1 iterate values of b to find the minimum
for (b.x in seq(from=0.2, to=0.4, by=0.001)) {
   # update parameters
   parameters.4 <- c(a=5, b=b.x, c=100, d=0.01, g=0.0001)
   # call ode function and get output
   ode.out4 <- ode(y=initial.4, times=times.4, func=derivs.4, parms=parameters.4)
   # find the minimum value of y
   my.4 <- min(ode.out4[,"y"])
```

```
    if (as.integer(b.x*1000) %% 10 == 0) {
        print(sprintf('b: %.3f -> min y: %f', b.x, my.4))
    }
    # if the minimum value of y less than 1, print result and break the loop
    if (my.4 < 1) {
        print(ode.out4[ode.out4[,'y']<=1, c('x','y')])
        plot(ode.out4[,"x"], ode.out4[,"y"], type="l",col="red",
                main='Trends of Fish and Humans', xlab="Number of Fish",
ylab="Number of Humans")
        print(sprintf('The minimum value of b is %.3f.', b.x))
        break
    }
}
```

Output:

```
                 x              y
[1,]    9883.251 0.9999307
[2,]    9920.026 0.9989476
[3,]    9956.903 0.9983330
[4,]    9993.882 0.9980873
[5,] 10030.961 0.9982112
[6,] 10068.139 0.9987058
[7,] 10105.416 0.9995728
[1] "The minimum value of b is 0.296."
```

As the result shown, the minimum value of b is 0.296, at which the human population dies out (y is about 0.9980873).

Question 5: Multivariate Optimization

Part A:

Run the following R codes and produce a plot of f(x, y):

```
# 5.a Part A: plot f(x, y)
# 5.a.1 define function f and expression f
f5 <- function(x, y) {
    (x^2 + y - 11)^2 + (x + y^2 - 7)^2
}
f5.exp <- expression(f(x, y) == (x^2 + y - 11)^2 + (x + y^2 - 7)^2)
# 5.a.2 define ranges of variables and calculate function results -4.5 <= x,y <= 4.5
x5 <- seq(-6, 6, length.out = 50)
y5 <- seq(-6, 6, length.out = 50)
f5.out <- outer(x5, y5, f5)
# 5.a.3 plot
persp(x5, y5, f5.out, theta = 30, phi = 30, expand = 0.5,
            col = 'lightblue', ltheta = 120, shade = 0.5,
            ticktype = 'detailed', xlab = 'x',
            ylab = 'y', zlab = 'f(x,y)', main='Graph of f(x,y)', sub=f5.exp)
```

The plot below shows there are 4 stationary points below the surface.

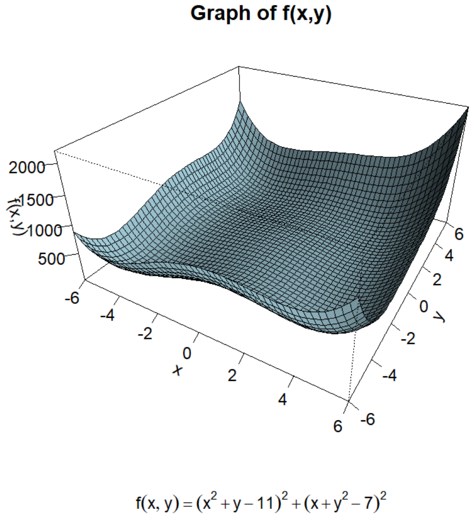


Figure 5

Part B:

To approximate f(x, y), a useful method is Newton Raphson, which is:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - [\nabla^2 f(\boldsymbol{x}_{k-1})]^{-1} \nabla f(\boldsymbol{x}_{k-1}),\ k = 1,2 \dots \quad (5.1)$$

The R codes shown below:

```
# 5.b To approximate f(x, y) by Newton Raphson method
# 5.b.1 define a multivariate differentiation function
MDeriv<-function(expr, x.name){
   #Get 1st partial derivatives
   deriv1<-lapply(x.name, function(x,f) D(f,x), f=expr)
   #Construct vector labels
   temp<-paste0("d",x.name)
   names(deriv1)<-temp
   #Initialize a list to store 2nd partial derivatives
   deriv2<-list()
   #Get 2nd partial derivatives
   for (i in 1:length(x.name)) {
      deriv2[[i]]<-lapply(x.name, function(x,f) D(f,x), f=deriv1[[i]])
   }
   #Construct vector of labels for 2nd partial derivatives
   names2<-c(sapply(temp, paste0, x.name))
   deriv2<-unlist(deriv2)
   names(deriv2)<-names2
   c(deriv1,deriv2)
}

# 5.b.2 define a Newton Raphson function
optim.NewtonRaphson<-function(expr,name,start){
   # Get partial derivatives
```

```
  mderiv <- MDeriv(expr, name)
  # first derivative vector
  grad <- function(xp) {
    data <- as.list(c(x <- xp[1], y <- xp[2]))
    c(with(data, eval(mderiv$dx)), with(data, eval(mderiv$dy)))
  }
  # the reverse of the second derivative matrix
  hess.rev <- function(xp) {
    data <- as.list(c(x<-xp[1],y<-xp[2]))
    with(data, eval(mderiv$dxx))
    mtx <- Matrix(c(with(data, eval(mderiv$dxx)),
                    with(data, eval(mderiv$dxy)),
                    with(data, eval(mderiv$dyx)),
                    with(data, eval(mderiv$dyy))),
                  nrow = 2, byrow = T)
    solve(mtx)
  }
  # Newton-Raphson, Want to find x satisfying grad(x)=0
  x.old<-start # starting value for x
  n.iter <- 1 # count the number of iterations
  while(n.iter <= 50){
    x.new <- x.old - hess.rev(x.old) %*% grad(x.old)
    tol<-sqrt(sum((x.new-x.old)^2))
    # convergence criteria
    if (tol <= 10^-10) {
      break
    }
    # footprints
    # if (n.iter%%5==0) print(x.new)
    x.old <- x.new
    n.iter <- n.iter + 1
  }
  c(value=with(as.list(c(x<-x.new[1],y<-x.new[2])), eval(expr)),
    optimum=x.new,
    gradient=grad(x.new),
    n.iterations=n.iter)
}
# Iterate several points to search critical points
for (start.5 in list(
  c(-4,-4),
  c(-4,4),
  c(4,-4),
  c(4,4)
  )){
```

```
  opt.out.5 <- optim.NewtonRaphson(expression((x^2 + y - 11)^2 + (x + y^2 -
7)^2),
                              c('x','y'),
                              start.5)
  print('----------------------solutition--------------------------')
  print(c(value=opt.out.5$val, optimum=opt.out.5$optimum))
}
```

To run the codes above and get the critical points:

```
The value is 7.888609e-31 at point (-3.779310, -3.283186).
The value is 7.888609e-31 at point (-2.805118, 3.131313).
The value is 0 at point (3.584428, -1.848127).
The value is 0 at point (3, 2).
```

Part C:

The first derivative of f(x,y) is

$$\frac{df}{dx} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} = \begin{bmatrix} 4*x^3 + 4*x*y - 42*x + 2*y^2 - 14 \\ 2*x^2 + 4*x*y + 2*y + 4*y^3 - 28*y - 22 \end{bmatrix} \tag{5.2}$$

The second derivative of f(x,y) is

$$\frac{d^2f}{dx^2} = \begin{bmatrix} \frac{d^2f}{dx^2} & \frac{d^2f}{dxdy} \\ \frac{d^2f}{dydx} & \frac{d^2f}{dy^2} \end{bmatrix} = \begin{bmatrix} 12*x^2 + 4*y - 42 & 4*(x+y) \\ 4*(x+y) & 4*x + 12*y^2 - 26 \end{bmatrix} \tag{5.3}$$

To find the critical points and calculate eigen values of hess matrix will be calculate by following codes in R.

```
# 5.c Find all the critical points
# 5.c.1 Write the function with parameters in a vector
f.vec.5 <- function(vec) (vec[1]^2+vec[2]-11)^2 + (vec[1]+vec[2]^2-7)^2
# 5.c.2 To get the second derivative matrix
mderiv <- MDeriv(expression((x^2 + y - 11)^2 + (x + y^2 - 7)^2),c('x', 'y'))
# 5.c.3 Iterate start points to find the critical points
for (start.5 in list(
  c(-4, -4),
  c(-4, 4),
  c(4, -4),
  c(4, 4)
  )){
  # call optim function and get a Optimal solution
  opt.out.5 <- optim(start.5,f.vec.5)
  data <- as.list(c(x=opt.out.5$par[1],y=opt.out.5$par[2]))
  # wrap and calculate hess matrix
  hess <- Matrix(c(with(data, eval(mderiv$dxx)),
                 with(data, eval(mderiv$dxy)),
                 with(data, eval(mderiv$dyx)),
```

```
                                with(data, eval(mderiv$dyy))),
                            nrow = 2, byrow = T)
    print(c(value=opt.out.5$val, point=opt.out.5$par,
eigen.values=eigen(hess)$values))
}
```

To run the codes above and get the critical points and eigen values:

value	x	y	eigen.val1	eigen.val2
1.005040e-07	-3.779347	-3.283172	133.878	70.1429
6.008897e-07	-2.805129	3.131435	80.55922	64.84158
1.162862e-06	3.584429	-1.848408	105.4177	28.70327
9.557439e-07	3.000108	1.999750	8.228795	25.70724

As the results shown, there are 4 critical points, and all of their eigen values are positive, thus all of them are local minimas.

Question 6: Back Propagation

Part A:

Though there are vectors such as $\boldsymbol{\alpha}_i^T$ and $\boldsymbol{x}$, their dot product is an express of variables x_1, x_2 and x_3, as well as $\boldsymbol{\beta}_i^T$ and $\boldsymbol{z}$. Thus $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are multivariable functions of x_1, x_2, and x_3.

For $l = k = 1$:

$$\frac{\partial f_k(\boldsymbol{x})}{\partial T_l} = \frac{\partial f_1(\boldsymbol{x})}{\partial T_1} = \frac{\partial(\frac{e^{T_1}}{e^{T_1}+e^{T_2}})}{\partial T_1} = \frac{e^{T_1}}{e^{T_1}+e^{T_2}} - \frac{e^{T_1}}{(e^{T_1}+e^{T_2})^2} * e^{T_1} = \frac{e^{T_1} * e^{T_2}}{(e^{T_1}+e^{T_2})^2}$$

$$= \frac{e^{T_1}}{e^{T_1}+e^{T_2}} * \frac{e^{T_2}}{e^{T_1}+e^{T_2}} = \frac{e^{T_1}}{e^{T_1}+e^{T_2}} * \left(1 - \frac{e^{T_1}}{e^{T_1}+e^{T_2}}\right) = f_1(\boldsymbol{x}) * (1 - f_1(\boldsymbol{x}))$$

$$= f_k(\boldsymbol{x}) * (1 - f_l(\boldsymbol{x})) \quad (6.1)$$

For $k = 1$ and $l \neq k (l = 2)$:

$$\frac{\partial f_k(\boldsymbol{x})}{\partial T_l} = \frac{\partial f_1(\boldsymbol{x})}{\partial T_2} = \frac{\partial\left(\frac{e^{T_1}}{e^{T_1}+e^{T_2}}\right)}{\partial T_2} = -\frac{e^{T_1}}{(e^{T_1}+e^{T_2})^2} * e^{T_2} = -\frac{e^{T_1}}{e^{T_1}+e^{T_2}} * \frac{e^{T_2}}{e^{T_1}+e^{T_2}}$$

$$= -f_1(\boldsymbol{x}) * f_2(\boldsymbol{x}) = -f_k(\boldsymbol{x}) * f_l(\boldsymbol{x}) \quad (6.2)$$

Similarly, it is easy to prove the symmetrical situation of $k = 2$.

Thus, $\frac{\partial f_k(x)}{\partial T_l} = \begin{cases} f_k(\boldsymbol{x}) * (1 - f_l(\boldsymbol{x}))\ if\ l = k; \\ -f_k(\boldsymbol{x}) * f_l(\boldsymbol{x}) \quad if\ l \neq k, \end{cases} \quad for\ k = 1,2.$ (6.3)

Part B:

Pat Bi:

As $T_1 = (\beta_{1,1}\ \beta_{2,1})\binom{z_1,}{z_2} = \beta_{1,1} * z_1 + \beta_{2,1} * z_2$,

thus, $\frac{\partial T_1}{\partial \beta_{1,1}} = z_1$. (6.4)

At part A, we already calculated $\frac{\partial f_k(x)}{\partial T_l}$ for $l = 1,2$ and $k = 1,2$,

thus,

$$\frac{\partial f_1}{\partial \beta_{1,1}} = \frac{\partial f_1}{\partial T_1} * \frac{\partial T_1}{\partial \beta_{1,1}} = f_1(\boldsymbol{x}) * \left(1 - f_1(\boldsymbol{x})\right) * z_1,$$

$$\frac{\partial f_2}{\partial \beta_{1,1}} = \frac{\partial f_2}{\partial T_1} * \frac{\partial T_1}{\partial \beta_{1,1}} = -f_2(\boldsymbol{x}) * f_1(\boldsymbol{x}) * z_1. \qquad (6.5)$$

Additionally, as $f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) = \frac{e^{T_1}}{e^{T_1}+e^{T_2}} + \frac{e^{T_2}}{e^{T_1}+e^{T_2}} = 1,\ f_1(\boldsymbol{x}) = 1 - f_2(x),$ (6.6)

the equation (6.3) can simplify to

$$\frac{\partial f_1(x)}{\partial T_1} = \frac{\partial f_2(x)}{\partial T_2} = f_1(\boldsymbol{x}) * f_2(\boldsymbol{x}) \text{ and } \frac{\partial f_1(x)}{\partial T_2} = \frac{\partial f_2(x)}{\partial T_1} = -f_1(\boldsymbol{x}) * f_2(\boldsymbol{x}). \qquad (6.7)$$

Given $R = -[(1-y) * log\ f_1(\boldsymbol{x}) + y * log\ f_2(\boldsymbol{x})]$,

So $\frac{\partial R}{\partial f_1(x)} = -\frac{1-y}{f_1(x)}$ and $\frac{\partial R}{\partial f_2(x)} = -\frac{y}{f_2(x)}$. (6.8)

Then, $\frac{\partial R}{\partial T_1} = \frac{\partial R}{\partial f_1(x)} * \frac{\partial f_1(x)}{\partial T_1} + \frac{\partial R}{\partial f_2(x)} * \frac{\partial f_2(x)}{\partial T_1}$

$$= -\frac{1-y}{f_1(x)} * f_1(\boldsymbol{x}) * f_2(\boldsymbol{x}) + \left(-\frac{y}{f_2(x)}\right) * \left(-f_1(\boldsymbol{x}) * f_2(\boldsymbol{x})\right)$$

$$= -[(1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})] \qquad (6.9)$$

And $\frac{\partial R}{\partial T_2} = \frac{\partial R}{\partial f_1(x)} * \frac{\partial f_1(x)}{\partial T_2} + \frac{\partial R}{\partial f_2(x)} * \frac{\partial f_2(x)}{\partial T_2}$

$$= -\frac{1-y}{f_1(x)} * (-f_1(\boldsymbol{x}) * f_2(\boldsymbol{x})) + \left(-\frac{y}{f_2(x)}\right) * f_1(\boldsymbol{x}) * f_2(\boldsymbol{x})$$

$$= (1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x}). \qquad (6.10)$$

Set $ex1 = (1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})$, thus simplify (6.9) and (6.10) to

$$\frac{\partial R}{\partial T_1} = -ex1 \text{ and } \frac{\partial R}{\partial T_2} = ex1. \qquad (6.11)$$

Given $T_1 = (\beta_{1,1}\ \beta_{2,1})\binom{z_1}{z_2}$ and $\boldsymbol{\beta}_1 = \binom{\beta_{1,1}}{\beta_{2,1}}$

Thus $\frac{\partial T_1}{\partial \boldsymbol{\beta}_1} = \binom{z_1}{z_2}$. (6.12)

Similarly, we can get $\frac{\partial T_2}{\partial \boldsymbol{\beta}_2} = \binom{z_1}{z_2}$. (6.13)

Thus,

$$\frac{\partial R}{\partial \beta_{1,1}} = \frac{\partial R}{\partial T_1} * \frac{\partial T_1}{\partial \beta_{1,1}} = -\mathrm{ex1} * \mathrm{z}_1 \text{ and } \frac{\partial R}{\partial \beta_{2.1}} = \frac{\partial R}{\partial T_1} * \frac{\partial T_1}{\partial \beta_{2,1}} = -\mathrm{ex1} * \mathrm{z}_2. \qquad (6.14)$$

Then $\frac{\partial R}{\partial \boldsymbol{\beta}_1} = -ex1\binom{z_1}{z_2} = (-1)^1[(1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})]\binom{z_1}{z_2}$ (6.15)

Similarly, for $\boldsymbol{\beta}_2$, we get

$$\frac{\partial R}{\partial \beta_{1,2}} = \frac{\partial R}{\partial T_2} * \frac{\partial T_2}{\partial \beta_{1,2}} = \mathrm{ex1} * \mathrm{z}_1 \text{ and } \frac{\partial R}{\partial \beta_{2,2}} = \frac{\partial R}{\partial T_2} * \frac{\partial T_2}{\partial \beta_{2,2}} = \mathrm{ex1} * \mathrm{z}_2. \qquad (6.16)$$

Then $\frac{\partial R}{\partial \boldsymbol{\beta}_2} = ex1\binom{z_1}{z_2} = (-1)^2[(1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})]\binom{z_1}{z_2}$ (6.17)

Combine (6.15) and (6.17), we get

$$\frac{\partial R}{\partial \boldsymbol{\beta}_k} = (-1)^k[(1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})]\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \text{ for } k=1, 2. \quad (6.18)$$

Part Bii:

Given $T_1 = (\beta_{1,1}\ \beta_{2,1})\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ and $\boldsymbol{\beta}_1 = \begin{pmatrix} \beta_{1,1} \\ \beta_{2,1} \end{pmatrix}$

Thus $\frac{\partial T_1}{\partial \boldsymbol{z}} = \begin{pmatrix} \beta_{1,1} \\ \beta_{2,1} \end{pmatrix} = \boldsymbol{\beta}_1$. (6.19)

Similarly, we get $\frac{\partial T_2}{\partial \boldsymbol{z}} = \begin{pmatrix} \beta_{1,2} \\ \beta_{2,2} \end{pmatrix} = \boldsymbol{\beta}_2$. (6.20)

As proved equation (6.11) shown $\frac{\partial R}{\partial T_1} = -ex1$ and $\frac{\partial R}{\partial T_2} = ex1$, thus

$$\frac{\partial R}{\partial z_1} = \frac{\partial R}{\partial T_1} * \frac{\partial T_1}{\partial z_1} + \frac{\partial R}{\partial T_2} * \frac{\partial T_2}{\partial z_1} = -ex1 * \beta_{1,1} + ex1 * \beta_{1,2}$$
$$= \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{1,k}. \quad (6.21)$$

As $z_1 = \frac{e^{\alpha_1^T x}}{1+e^{\alpha_1^T x}} = 1 - \frac{1}{1+e^{\alpha_1^T x}}$, $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, and $\boldsymbol{\alpha}_1 = \begin{pmatrix} \alpha_{1,1} \\ \alpha_{2,1} \\ \alpha_{3,1} \end{pmatrix}$,

thus $\frac{\partial z_1}{\alpha_{1,1}} = \frac{1}{\left(1+e^{\alpha_1^T x}\right)^2} * e^{\alpha_1^T x} * x_1$. (6.22)

Set $ex2 = \frac{e^{\alpha_1^T x}}{\left(1+e^{\alpha_1^T x}\right)^2}$, equation (6.22) can simplify to $\frac{\partial z_1}{\alpha_{1,1}} = ex2 * x_1$. (6.23)

Thus $\frac{\partial R}{\partial \alpha_{1,1}} = \frac{\partial R}{\partial z_1} * \frac{\partial z_1}{\partial \alpha_{1,1}} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{1,k} * ex2 * x_1$.

Similarly, we can get $\frac{\partial R}{\partial \alpha_{2,1}} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{1,k} * ex2 * x_2$

and $\frac{\partial R}{\partial \alpha_{3,1}} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{1,k} * ex2 * x_3$.

Thus $\frac{\partial R}{\partial \boldsymbol{\alpha}_1} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{1,k} * ex2 * \boldsymbol{x}$. (6.24)

Considering $\alpha_{1,1}$ and $\alpha_{1,2}$, they have the similar relationship to function $z_1$and z_2, respectively. For $z_1$and z_2, they also have similar relationship to function T_1 and T_2, just need to substitute $\beta_{1,i}$ to $\beta_{2,i}$ (for i = 1, 2). It works through elements of $\boldsymbol{\alpha}_1$and $\boldsymbol{\alpha}_2$, thus

$$\frac{\partial R}{\partial \boldsymbol{\alpha}_2} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{2,k} * ex2 * \boldsymbol{x}. \quad (6.25)$$

Combine equation (6.24) and (6.25), we get,

$$\frac{\partial R}{\partial \boldsymbol{\alpha}_s} = \sum_{k=1}^{2}(-1)^k * ex1 * \beta_{s,k} * ex2 * \boldsymbol{x}$$

$$= \sum_{k=1}^{2}(-1)^k * [(1-y) * f_2(\boldsymbol{x}) - y * f_1(\boldsymbol{x})] * \beta_{s,k} * \frac{e^{\alpha_1^T x}}{\left(1+e^{\alpha_1^T x}\right)^2} * \boldsymbol{x}. \quad (6.26)$$