VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

MATHEMATICAL MODELING (CO2011)

Assignment

Dynamics of Love

Students: Đỗ Phạm Khải Nguyên - 2153629.
Hoàng Phan Ngọc Minh - 2053214.
Phạm Lê Quân - 2153749.
Nguyễn Đăng Hoài Nam - 2152181.

HO CHI MINH CITY, NOVEMBER 2022

Contents

Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Đỗ Phạm Khải Nguyên	2153629	- Problems 1, 2, 3, 4	37.5%
2	Hoàng Phan Ngọc Minh	2053214	- Problems 1, 2	25%
2	Phạm Lê Quân	2153749		0%
2	Nguyễn Đăng Hoài Nam	2152181	- Problem 1, 3, 4, 5	37.5%

1 Exercises

1.1 Exercise 1

Solving Initial Problem:

The IVPs Sys. (1) has the form:

$$\begin{cases} R' = aR + bJ \\ J' = cR + dJ \\ R(0) = R_0, J(0) = J_0 \end{cases} \tag{1}$$

This is a linear system of two **ODEs**, where $R_0, J_0 \in \mathbb{R}$ are the initial value of their love and $a, b, c, d \in \mathbb{R}$ represent the interaction of the love of one to the other. And the matrix form is:

$$\begin{cases} u' = Au \\ u(0) = u_0 \end{cases} \tag{2}$$

where $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} R_0 \\ J_0 \end{pmatrix}$.

The above Sys. (1) can be solved and represent the solution in terms of eigenvalues and eigenvectors of matrix A.

Firstly, we consider that $\lambda \in \mathbb{C}$ is called an eigenvalue of a matrix A if there is a vector $\vec{v} \in \mathbb{C}^2$ that nonzero that satisfies:

$$A\vec{v} = \lambda\vec{v}. \tag{3}$$

We can solve for λ by the the characteristic equation of matrix A:

$$det(A - \lambda I) = 0. \tag{4}$$

Which is the solution of the equation:

$$\lambda^2 - (a + d)\lambda + ad - bc = 0.$$

From the values of λ we can determine the eigenvectors $\vec{v}$ by formula:

$$(A - \lambda I)\vec{v} = 0. \tag{5}$$

Base on the solutions of eigenvalues and eigenvectors we can solve for the solution of Sys. (1). From Sys. (1), the second derivative of R can be written as:

$$\begin{cases} R'' = (a + d)R' + (bc - ad)R \\ J'' = (a + d)J' + (bc - ad)J \end{cases} \tag{6}$$

We can solve Sys. (6) by solving the second-order linear equations [1]. We consider the equation:

$$u'' - (a + d)u' + (ad - bc)u = 0$$

Suppose that a solution $u = e^{kt}.\vec{c}$, where k is a determined parameter and $\vec{c}$ is a determined vector. Then $u' = ke^{kt}.\vec{c}$ and $u'' = k^2e^{kt}.\vec{c}$. Substituting these expressions back to the previous equation, we have the equation:

$$(k^2 - (a + d)k + ad - bc)e^{kt}.\vec{c} = 0$$

Since $e^{kt} \neq 0$ and $\vec{c} \neq \vec{0}$, this equation can be written as:

$$k^2 - (a+d)k + ad - bc = 0. \tag{7}$$

Eq. (7) is called the characteristic equation. If we write Eq. (7) as the form of determinant of matrix:

$$det(A - kI) = 0. \tag{8}$$

Compare the Eq. (8) and Eq. (4), we can conclude that the value of variable k is λ. In other words, $u = e^{\lambda t}.\vec{c}$. Furthermore, from Eq. (3) and Sys. (2) we can conclude that $\vec{c} = \vec{v}$, where $\vec{v}$ is the vector that satisfies Eq. (3) or can be called the eigenvectors of matrix A. Then a solution is given by:

$$u = e^{\lambda t}.\vec{v}.$$

In case the 2×2 matrix A is diagonalizable, then it has eigenvalues λ_1, λ_2 with linearly independent eigenvectors $\vec{v}_1$, $\vec{v}_2$.
Solving $\vec{v_1}$ and $\vec{v_2}$ base on eigenvalue and Eq. (5):

$$\vec{v}_1 = \begin{pmatrix} -b \\ a - \lambda_1 \end{pmatrix}, \vec{v}_2 = \begin{pmatrix} d - \lambda_2 \\ -c \end{pmatrix}$$

with the understanding that when $bc = 0, \lambda_1 = d$ and $\lambda_2 = a$. (Outside that case, we can use the eigenvalues in either order.)

The general solution can be divided in 3 cases:
Case 1: If the Eq. (4) has real and different roots:

$$\lambda_1 = \frac{a + d - \sqrt{(a-d)^2 + 4bc}}{2}, \lambda_2 = \frac{a + d + \sqrt{(a-d)^2 + 4bc}}{2} \tag{9}$$

Then the general solution will be:

$$u = c_1 e^{\lambda_1 t}.\vec{v_1} + c_2 e^{\lambda_2 t}.\vec{v_2}. \tag{10}$$

To satisfy the initial condition in Sys. (2), we choose the scalar constants $c_1, c_2 \in \mathbb{C}$ so that:

$$u_0 = c_1.\vec{v}_1 + c_2.\vec{v}_2.$$

c_1, c_2 are unique constants because $\vec{v}_1$, $\vec{v}_2$ are linearly independent and hence form a basis of $\mathbb{C}^2$.
Solving for c_1, c_2:

$$\begin{pmatrix} R_0 \\ J_0 \end{pmatrix} = c_1 \begin{pmatrix} -b \\ a - \lambda_1 \end{pmatrix} + c_2 \begin{pmatrix} d - \lambda_2 \\ -c \end{pmatrix} \tag{11}$$

Let $\psi_1 = a - \lambda_1, \psi_2 = d - \lambda_2$ .From Eq. (9) and Eq. (11):

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \frac{cR_0 + \psi_2 J_0}{\psi_1 \psi_2 - bc} \\ \frac{bJ_0 + \psi_1 R_0}{\psi_1 \psi_2 - bc} \end{pmatrix}$$

Case 2: If A is nondiagonalizable with a repeated eigenvalue $\lambda = \frac{a+d}{2}$, then it has linearly independent generalized eigenvectors $\vec{v}_1, \vec{v}_2$ such that:

$$(A - \lambda I).\vec{v}_1 = 0, (A - \lambda I).\vec{v}_2 = \vec{v}_1.$$

Solving $\vec{v}_2$ with $\vec{v}_1 = \begin{pmatrix} -b \\ a-\lambda \end{pmatrix} = \begin{pmatrix} -b \\ \frac{a-d}{2} \end{pmatrix}$ by $(A-\lambda I).\vec{v}_2 = \vec{v}_1$. So, $\vec{v}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

The solution of the ODE is

$$u = c_1 e^{\lambda t}.\vec{v}_1 + c_2 e^{\lambda t}(t\vec{v}_1 + \vec{v}_2). \tag{12}$$

where constant c_1, c_2 are chosen so that:

$$u_0 = c_1.\vec{v}_1 + c_2.\vec{v}_2.$$

Solving for c_1, c_2:

$$\begin{pmatrix} R_0 \\ J_0 \end{pmatrix} = c_1 \begin{pmatrix} -b \\ \frac{a-d}{2} \end{pmatrix} + c_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tag{13}$$

From Eq. (13):

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \frac{-R_0}{b} \\ \frac{R_0(d-a)}{2b} - J_0 \end{pmatrix}$$

Case 3: If the characteristic equation has complex conjugate roots, let $\alpha = \frac{a+d}{2}, \beta = \frac{\sqrt{-(a-d)^2-4bc}}{2}$,then $\lambda_1 = \alpha + i\beta, \lambda_2 = \alpha - i\beta$ and we can use the Euler's Formula:

$$e^{i\theta} = cos\theta + isin\theta.$$

We can solve to choose a value of $\vec{v}$ by Eq. (5):

$$\vec{v} = \begin{pmatrix} -2b \\ a-d-i\sqrt{-(a-d)^2-4bc} \end{pmatrix}$$

Then the general solution is:

$$u = \begin{pmatrix} R \\ J \end{pmatrix} = e^{\alpha t} \begin{pmatrix} -2b(\cos\beta t)c_1 + -2b(\sin\beta t)c_2 \\ [(a-d)\cos\beta t + 2\beta\sin\beta t]c_1 + [(a-d)\sin\beta t - 2\beta\cos\beta t]c_2 \end{pmatrix}$$

where c_1, c_2 are constant.

Solving c_1, c_2: $\begin{pmatrix} R_0 \\ J_0 \end{pmatrix} = \begin{pmatrix} -2bc_1 \\ (a-d)c_1 - 2\beta c_2 \end{pmatrix}$, we have: $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \frac{-R_0}{2b} \\ \frac{(d-a)R_0 - 2bJ_0}{4b\beta}) \end{pmatrix}$

Re λ_1	**Re λ_2**	**Im $\lvert\lambda_1\rvert$**	**Im$\lvert\lambda_2\rvert$**	**Type**
-	-	+	+	**Spiral Sink (stable)**
+	+	+	+	**Spiral Source (unstable)**
0	0	+	+	**Center**
+	-	0	0	**Saddle**
+	+	0	0	**Nodal Source (unstable)**
-	-	0	0	**Nodal Sink (stable)**
+	0	0	0	**Singular (Repulsive Line)**
-	0	0	0	**Singular (Attractive Line)**

Bảng 1: (Complete) Phase-portrait classification

1.2 Exercise 2

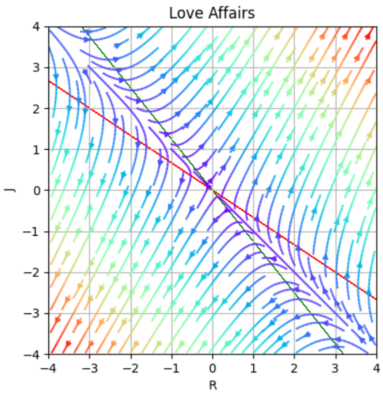
1.2.1 Saddle Node:

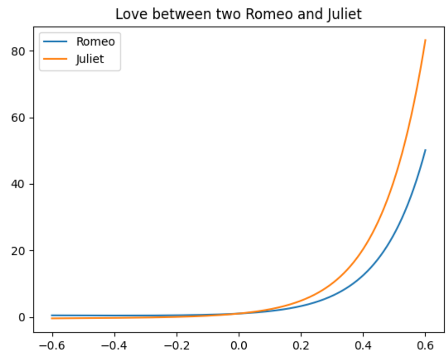
Example: $A = \begin{pmatrix} 2 & 3 \\ 5 & 4 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = 7$, $\lambda_2 = -1$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} -3 \\ -5 \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} 5 \\ -5 \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 1 will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = \frac{-1}{4} e^{7t} . \begin{pmatrix} -3 \\ -5 \end{pmatrix} + \frac{1}{20} e^{-t} . \begin{pmatrix} 5 \\ -5 \end{pmatrix} . \tag{14}$$

From the **Table 1** above this example represent an saddle node and the phase portrait of this example is on the left and love between them is on the right.





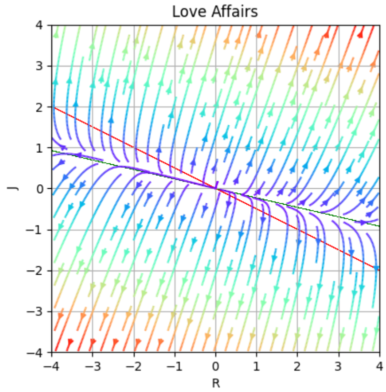
1.2.2 Nodal Source (unstable):

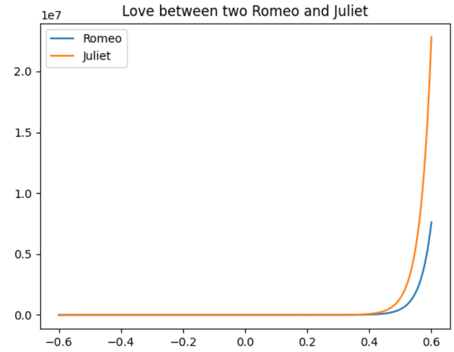
Example: $A = \begin{pmatrix} 4 & 8 \\ 6 & 26 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = 28$, $\lambda_2 = 2$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} -4 \\ 1 \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 1 will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = \frac{5}{13} e^{28t} . \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \frac{-2}{13} e^{2t} . \begin{pmatrix} -4 \\ 1 \end{pmatrix} . \tag{15}$$

From the **Table 1** above this example represent a nodal source (unstable) and the phase portrait of this example is on the left and love between them is on the right.





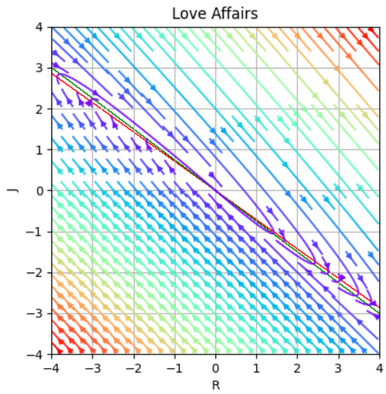
1.2.3 Nodal Sink (stable):

Example: $A = \begin{pmatrix} 5 & 7 \\ -6 & -8 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = -1$, $\lambda_2 = -2$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} -7 \\ 6 \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 1 will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = -2e^{-t}.\begin{pmatrix} -7 \\ 6 \end{pmatrix} + 13.e^{-2t}.\begin{pmatrix} -1 \\ 1 \end{pmatrix}. \tag{16}$$

From the **Table 1** above this example represent a nodal sink (stable) and the phase portrait of this example is on the left and love between them is on the right.



1.2.4 Singular (Repulsive line):

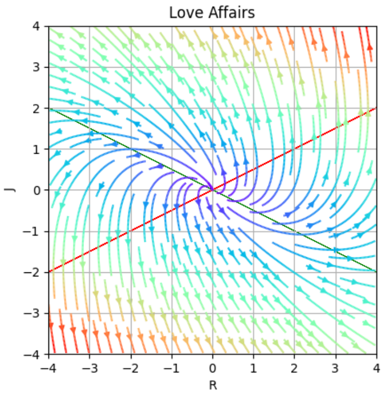
Example: $A = \begin{pmatrix} 1 & -2 \\ 2 & 4 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

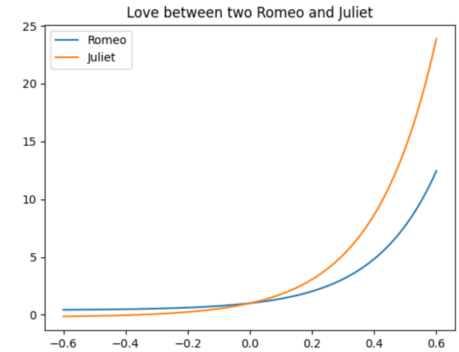
Solving for λ are $\lambda_1 = 5$, $\lambda_2 = 0$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$.

Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 1 will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = \frac{3}{5} e^{5t} . \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \frac{-1}{5} . e^{0t} . \begin{pmatrix} -2 \\ 1 \end{pmatrix} . \tag{17}$$

From the **Table 1** above this example represent a singular (repulsive line) and the phase portrait of this example is on the left and love between them is on the right.





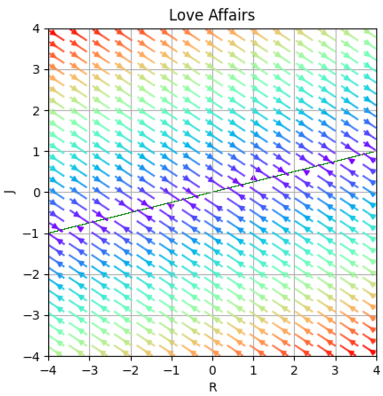
1.2.5 Singular (Attractive line):

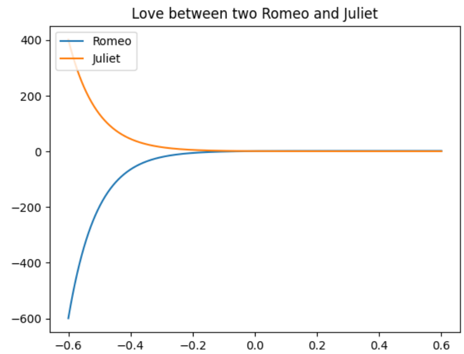
Example: $A = \begin{pmatrix} -3 & 12 \\ 2 & -8 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = 0$, $\lambda_2 = -11$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 1 will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = \frac{5}{11} e^{0t} . \begin{pmatrix} 4 \\ 1 \end{pmatrix} + \frac{3}{11} . e^{-11t} . \begin{pmatrix} -3 \\ 2 \end{pmatrix} . \tag{18}$$

From the **Table 1** above this example represent a singular (attractive line) and the phase portrait of this example is on the left and love between them is on the right.





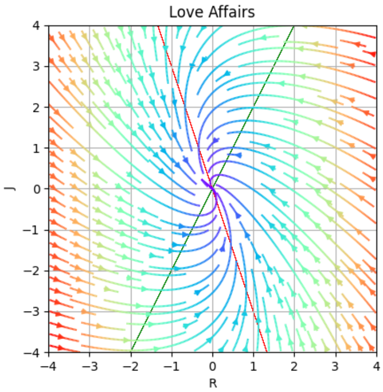
1.2.6 Spiral Sink (stable):

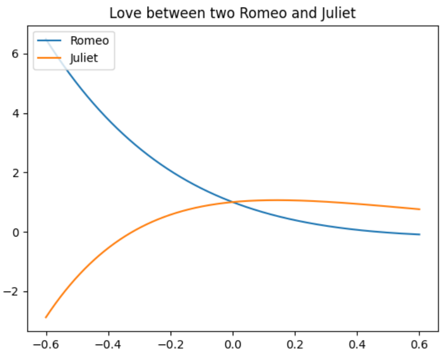
Example: $A = \begin{pmatrix} -3 & -1 \\ 2 & -1 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = -2 +$ i, $\lambda_2 = -2$ - i. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 1 \\ -1-i \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} 1 \\ -1+i \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 3 ($\alpha =$ -2 and $\beta = 1$) = will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = e^{-2t}.\begin{pmatrix} \cos{(t)} - 2\sin{(t)} \\ \cos{(t)} + 3\sin{(t)} \end{pmatrix} \tag{19}$$

From the **Table 1** above this example represent a spiral sink (stable) and the phase portrait of this example is on the left and love between them is on the right.





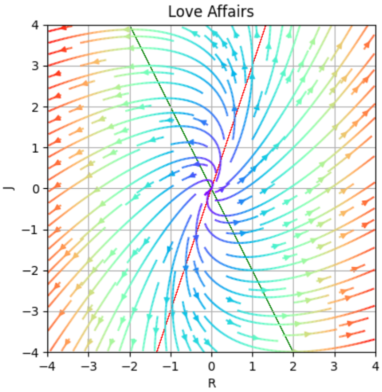
1.2.7 Spiral Source (unstable):

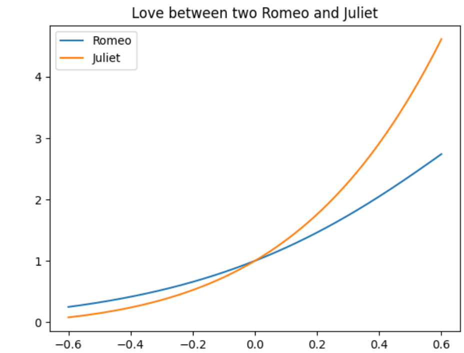
Example: $A = \begin{pmatrix} 3 & -1 \\ 2 & 1 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = 2 +$ i, $\lambda_2 = 2$ - i. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 1 \\ 1-i \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} 1 \\ 1+i \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 3 ($\alpha = 2$ and $\beta = 1$) = will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = e^{2t}.\begin{pmatrix} \cos{(t)} \\ \cos{(t)} + \sin{(t)} \end{pmatrix} \tag{20}$$

From the **Table 1** above this example represent a spiral source (unstable) and the phase portrait of this example is on the left and love between them is on the right.





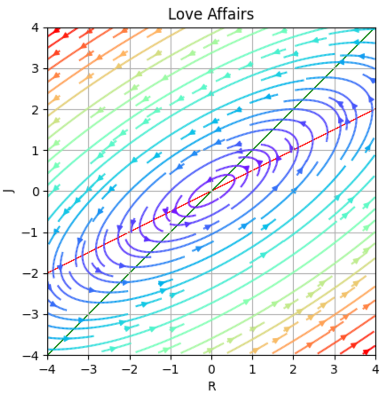
1.2.8 Center:

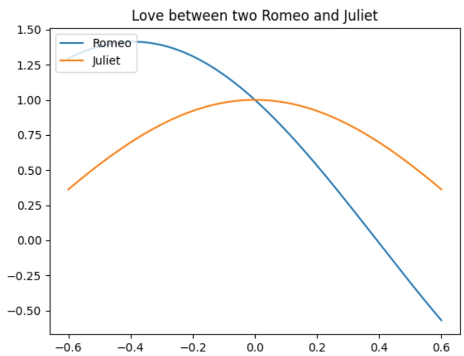
Example: $A = \begin{pmatrix} 2 & -4 \\ 2 & -2 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$ and $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Solving for λ are $\lambda_1 = 2\text{i}$, $\lambda_2 = -2\text{i}$. From λ the value of eigenvectors are $\vec{v}_1 = \begin{pmatrix} 2 \\ 1-i \end{pmatrix}$, $\vec{v}_2 = \begin{pmatrix} 2 \\ 1+i \end{pmatrix}$. Because $\lambda_1 \neq \lambda_2$, then the solution after substitute into Case 3 ($\alpha = 0$ and $\beta = 2$) = will be:

$$\begin{pmatrix} R(t) \\ J(t) \end{pmatrix} = \begin{pmatrix} \cos(2t) - \sin(2t) \\ \cos(2t) \end{pmatrix} \tag{21}$$

From the **Table 1** above this example represent a center and the phase portrait of this example is on the left and love between them is on the right.





1.3 Exercise 3

1.3.1 System of non-homogeneous ODEs

The system of non-homogeneous equation has the form:

$$\begin{cases} R' = aR + bJ + f(t) \\ J' = cR + dJ + g(t) \\ R(0) = R_0, J(0) = J_0 \end{cases} \tag{22}$$

where $f(t)$, $g(t) \neq 0$, $R_0, J_0 \in \mathbb{R}$ are the initial value of their love and $a, b, c, d \in \mathbb{R}$ represent the interaction of the love of one to the other.

The previous problem in Ex. (1) is called a system of homogeneous equations, it happen when $f(t) = 0$ and $g(t) = 0$ .All the results that have been discussed in the homogeneous problem can be reused to solve this non-homogeneous problems.

This can be written in the matrix form is

$$\begin{cases} u' = Au + \vec{h}(t) \\ u(0) = u_0 \end{cases} \tag{23}$$

where $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $\vec{h} = \begin{pmatrix} f(t) \\ g(t) \end{pmatrix}$ and $u_0 = \begin{pmatrix} R_0 \\ J_0 \end{pmatrix}$.

First, we have to define $\phi(t)$ which is an non-singular square matrix that:

$$\phi'(t) = A\phi(t) \text{ for all t } t \in \mathbb{R}$$

this matrix is called fundamental matrix of $u' = Au$.

The solution of the non-homogeneous Sys. (18), when $\phi(t)$ is an fundamental matrix of $u' = Au$ is unique and given by:

$$u(t) = \phi(t)\phi^{-1}(0)u_0 + \int_0^t \phi(t)\phi^{-1}(s)h(s)\, ds$$

To proof this:

$$\begin{aligned} u'(t) &= \phi'(t)\phi^{-1}u_0 + \phi'(t)\int_0^t \phi^{-1}(s)h(s)\, ds + \phi(t)\phi^{-1}(t)h(t) \\ &= A[\phi(t)\phi^{-1}u_0 + \phi(t)\int_0^t \phi^{-1}(s)h(s)\, ds] + h(t) \\ &= Au + h(t). \end{aligned}$$

This is correct for all $t \in \mathbb{R}$.

With $\phi(t) = e^{At}$ (e^{At} is called the exponential of matrix [3]), the solution is given by:

$$u(t) = e^{At}u_0 + e^{At}\int_0^t e^{-As}h(s)\, ds \tag{24}$$

As we can we as the Eq. (19), the condition for this solution to exit is $h(s)$ is integrable.

If f is continuous on [a,b] , or if f has only a finite number of jump discontinuities, then f is integrable on [a,b]; that is, the definite integral exists [2].

The simplest examples for functions that non-integrable are: $\frac{1}{x}$ in the interval [0, t] or $\frac{1}{x^2}$ in any interval that containing 0, for instance, [0, t]. Then, the Sys. (17) has solution when $f(t), g(t)$ are integrable, for example:

Example 1: With $A = \begin{pmatrix} 2 & 3 \\ 5 & 4 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $h(t) = \begin{pmatrix} 3t+2 \\ 4t+5 \end{pmatrix}$.

Which mean:

$$\begin{cases} R' = 2R + 3J + 3t + 2 \\ J' = 5R + 4J + 4t + 5 \\ R_0 = 1, J_0 = 1. \end{cases}$$

Example 2: With $A = \begin{pmatrix} 5 & 7 \\ -6 & -8 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $u_0 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ and $h(t) = \begin{pmatrix} t^2+1 \\ 4t^3 \end{pmatrix}$.

Which mean:

$$\begin{cases} R' = 5R + 7J + t^2 + 1 \\ J' = -6R - 8J + 4t^3 \\ R_0 = 0, J_0 = -1. \end{cases}$$

Example 3: With $A = \begin{pmatrix} 7 & 9 \\ 2 & 6 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $u_0 = \begin{pmatrix} -6 \\ 2 \end{pmatrix}$ and $h(t) = \begin{pmatrix} e^t+3 \\ e^{4t} \end{pmatrix}$.

Which mean:

$$\begin{cases} R' = 7R + 9J + e^t + 3 \\ J' = 2R + 6J + e^{4t} \\ R_0 = -6, J_0 = 2. \end{cases}$$

Example 4: With $A = \begin{pmatrix} 2 & -1 \\ -5 & 9 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $u_0 = \begin{pmatrix} 0 \\ -7 \end{pmatrix}$ and $h(t) = \begin{pmatrix} \cos 3t+2 \\ \sin 5t-1 \end{pmatrix}$.

Which mean:

$$\begin{cases} R' = 2R - J + \cos 3t + 2 \\ J' = -5R + 9J + \sin 5t - 1 \\ R_0 = 0, J_0 = -7. \end{cases}$$

Example 5: With $A = \begin{pmatrix} -10 & -8 \\ 5 & -24 \end{pmatrix}$, $u = \begin{pmatrix} R \\ J \end{pmatrix}$, $u_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $h(t) = \begin{pmatrix} \cos(6t+2)+4 \\ \sin 5t-2 \end{pmatrix}$.

Which mean:

$$\begin{cases} R' = -10R - 8J + \cos(6t+2) + 4 \\ J' = 5R - 24J + \sin 5t - 2 \\ R_0 = 0, J_0 = 0. \end{cases}$$

1.3.2 Nonlinear system of ODEs

To find the condition of the nonlinear system of ODEs to have unique solution:

$$\begin{cases} R' = f(t, R, J) \\ J' = g(t, R, J) \\ R(0) = R_0, J(0) = J_0 \end{cases} \tag{25}$$

We have to have the basic understanding about the flowing theorem.

1.3.2.a Picard–Lindelöf theorem

As an aplication of the contraction mapping principle [4], this theorem provide the condition of the nonlinear system of ODEs to have unique solution.

Theorem 1 *For $y'(x) = f(x, y(x))$ and $y(x_0) = y_0$.*

If the function $f \in C(\mathbb{R}^2, R)$ is Lipschitz continuous and continuous in x, then, for any initial condition $y(x_0) = y_0$, there exists a neighborhood $U(x_0)$ of $x_0 \in \mathbb{R}$ and a unique function $y = y(x)$ defined in $U(x_0)$ satisfying the equation:

$$y'(x) = f(x, y)$$

and the initial condition.

The proof can be obtained by using Banach fixed-point theorem, Picard iteration (finding the solution) and Grönwall's lemma (proving the global uniqueness) [4].

1.3.2.b Lipschitz continuity

In general, every function that has bounded first derivatives is Lipschitz continuous. Lipschitz continuity is the central condition of the Picard–Lindelöf theorem which guarantees the existence and uniqueness of the solution to an initial value problem.

Definition: function $f(t, y)$ satisfies a Lipschitz condition in the variable y on a set $D \subset R^2$ if a constant $L > 0$ exists with

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|$$

whenever $(t, y_1), (t, y_2)$ are in D. L is Lipschitz constant.

1.3.2.c Banach fixed-point theorem

Theorem 2 *Let (X, d) be a non-empty complete metric space. Then a map $T : X \mapsto X$ is called a contraction mapping on X if there exists $q \in [1, 0)$ such that*

$$d(T(x), T(y)) \leq q.d(x, y) \textit{ for all x,y} \in X$$

Then T admits a unique fixed-point x^* in X.

Base on the above theory, these are some example that have unique solution:

Example 1:

$$\begin{cases} R' = R + RJ \\ J' = J + RJ \\ R(0) = 1, J(0) = 0. \end{cases}$$

Example 2:

$$\begin{cases} R' = R + \sqrt{R^2 + J^2} \\ J' = J + \sqrt{R^2 - J} \\ R(0) = 1, J(0) = 1. \end{cases}$$

Example 3:

$$\begin{cases} R' = \sin R + \sin J \\ J' = \cos R - \cos J \\ R(0) = 1, J(0) = -1. \end{cases}$$

Example 4:

$$\begin{cases} R' = \sqrt[3]{6R^3 - 2J^3} \\ J' = \sqrt[3]{8J^3 - 4R^3} \\ R(0) = 3, J(0) = 2. \end{cases}$$

Example 5:

$$\begin{cases} R' = \sin\sqrt{R^2 + 5J} + 5 \\ J' = \cos\sqrt{4J^2 - R} - 9 \\ R(0) = 0, J(0) = 7. \end{cases}$$

1.4 Exercise 4

1.4.1 Proof of error proportion

To prove that $\varepsilon(t_1)$ is proportional to h^2. We have to consider the Taylor series.

The Taylor series of a real or complex-valued function $f(x)$ that is infinitely differentiable at a real or complex number a is the power series:

$$f(x) = \sum_{n=1}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{26}$$

To use Taylor series we have to prove that the Sys. (17) and Sys. (22) have solution that is infinitely differentiable at t_0.

As we have proved in Exercise 3, the condition for the given system to have solution is satisfying the Theorem. (1) which mean they are differentiable at t_0. Because of that, the solution is at least have the second order derivative.

Assume that R, J are infinitely differentiable, from Eq. (26), we have:

$$R(t_1) = R_0 + \frac{R'(t_0).h}{1!} + \frac{R''(t_0).h^2}{2!} + ... + \frac{R^{(n)}(t_0).h^n}{n!}. \tag{27}$$

and

$$J(t_1) = J_0 + \frac{J'(t_0).h}{1!} + \frac{J''(t_0).h^2}{2!} + ... + \frac{J^{(n)}(t_0).h^n}{n!}. \tag{28}$$

where $h = t_1 - t_0$. In the other hand, $R_1 = R_0 + R'(t_0) * h$ and $J_1 = J_0 + J'(t_0) * h$, combine with Eq. (27), Eq. (28), the equation $\epsilon(t_1) = \sqrt{[R(t_1) - R_1]^2 + [J(t_1) - J_1]^2}$ becomes:

$$\epsilon(t_1) = h^2\sqrt{[\frac{R''(t_0)}{2!} + ... + \frac{R^{(n)}(t_0).h^{n-2}}{n!}]^2 + [\frac{J''(t_0)}{2!} + ... + \frac{J^{(n)}(t_0).h^{n-2}}{n!}]^2} \tag{29}$$

Because we have prove that R, J at least have the second derivative so the $\epsilon(t)$ will always proportional to h^2.

1.4.2 Implicit Euler Method

In numerical analysis and scientific computing, the backward Euler method (or implicit Euler method) is one of the most basic numerical methods for the solution of ordinary differential equations. It is similar to the (standard) Euler method, but differs in that it is an implicit method. The backward Euler method has error of order one in time.
Consider the ordinary differential equation

$$\frac{d_y}{d_t} = f(t, y)$$

with initial value $y(t_0) = y_0$. Here the function f and the initial data t_0 and y_0 are known; the function y depends on the real variable t and is unknown. A numerical method produces a sequence $y_0, y_1, y_2, ...$ such that y_k approximates $y(t_0 + kh)$, where h is called the step size.

The backward Euler method computes the approximations using

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}).$$

This differs from the (forward) Euler method in that the forward method uses $f(t_k, y_k)$ in place of $f(t_{k+1}, y_{k+1})$.

The backward Euler method is an implicit method: the new approximation y_{k+1} appears on both sides of the equation, and thus the method needs to solve an algebraic equation for the unknown y_{k+1}. For non-stiff problems, this can be done with fixed-point iteration:

$$y_k^{[0]}+1 = y_k, y_k^{[i+1]}+1 = y_k + hf(t_{k+1}, y_k^{[i]}+1).$$

If this sequence converges (within a given tolerance), then the method takes its limit as the new approximation y_{k+1}.

Alternatively, one can use (some modification of) the Newton–Raphson method to solve the algebraic equation.

1.4.3 Newton's Method

In numerical analysis, Newton's method, also known as the Newton–Raphson method, named after Isaac Newton and Joseph Raphson, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function. The most basic version starts with a single-variable function f defined for a real variable x, the function's derivative f, and an initial guess x_0 for a root of f. If the function satisfies sufficient assumptions and the initial guess is close, then

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

is a better approximation of the root than x_0. Geometrically, $(x_1, 0)$ is the intersection of the x-axis and the tangent of the graph of f at $(x_0, f(x_0))$: that is, the improved guess is the unique root of the linear approximation at the initial point. The process is repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

until a sufficiently precise value is reached. The method will usually converge, provided this initial guess is close enough to the unknown zero, and that $f(x_0) \neq 0$.

1.5 Exercise 5

1.5.1 Approach

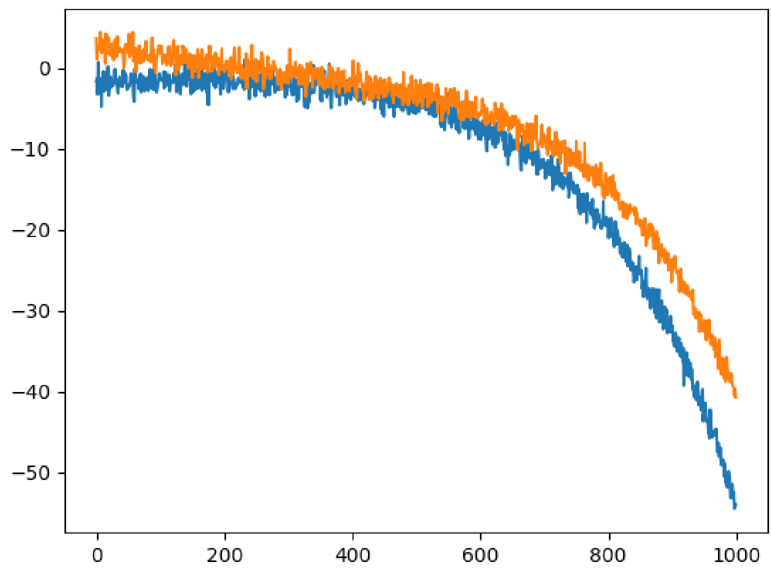
In this exercise, we choose Python for coding because it's easy to approach and has strong tools for Machine Learning and Deeplearing.
First thing first, we change data appropriately into DataFrame using Pandas library and visualize it by matplotlib library to consider the way we approach to solve the problem:

Building DataFrame:

```
import pandas as pd
df = pd.read_excel("Directory") #in the excel file, we deleted the order column
df.column = ['R', 'J']          #label columns
```

Visualizing:

```
import matplotlib.pyplot as plt
plt.plot(df)
plt.show()
```



e

Figure 2

From the graph (Figure 2), we can see that R(t) and J(t) seem to be exponential functions and we are able to use regression for approximating the correct exponential lines and parameters inference. Remember that: this is an initial value problem of ODEs so we can use Euler method [5] to predict the shape of the function from initial values and differential equations.

With $y'(t) = f(t, y(t))$, $y(t_0) = y_0$ and h is size of every each step, Euler method formula: $y_{n+1} = y_n + h.f(t_n, y_n)$

In our problem, $u' = Au$, $u_0 = [-2, 3].T$ and $h = 0.001$ so the formula should be: $u_{t+1} = u_t + h.A.u_t$ but we use z for this formula to distinguish between predicted point z_t from Euler method and real point u_t from dataset. The formula will be: $z_{t+1} = z_t + h.A.z_t$

The formula for prediction contains matrix A and it's also the parameters matrix that we want to infer so we build loss function for regression based on this formula. For every data point, the error of Euler method predicted point and real data point is $|z_t - u_t|$ or $|(z_{t-1} + h.A.z_{t-1}) - u_t|$ with z is the predicted point and u is the real point. Note that $z_0 = [-2, 3]$.

From the error of each data point, we have the Loss function for all data point (Least Squared Error):

$$L(A) = \frac{1}{2N} \sum_{t=1}^{N} \|(z_{t-1} + h.A.z_{t-1}) - u_t\|_2^2$$

$L(A)$ is called the loss function of the Linear Regression problem. We always want the loss (error) to be as small as possible, which means finding the coefficient matrix A so that the value of this loss function is minimum. Value of A make the loss function reach the minimum value called the optimal point (optimal point), denoted:

$$A^* = \underset{A}{\operatorname{argmin}}\, L(A)$$

The most common way to find the solution to an optimization problem is to solve the zero gradient equation! That is, of course, when calculating the derivative and solving the zero-derivative equation are not so complicated. However, we use Gradient Descent [6] instead of solving the zero gradient equation.

Gradient of the Loss function with respect to A:

$$\frac{\partial L(A)}{\partial A} = \frac{1}{N} \sum_{t=1}^{N} h(z_{t-1} + h.A.z_{t-1} - u_t).z^T$$

Formula to update coefficient matrix A:

$$A = A - \eta A \text{ with learning rate } \eta = 0.5$$

Rule: always go in the opposite direction of the derivative.

1.5.2 Implement

Based on the idea of approaching, we implement the algorithm:

```
import numpy as np
from numpy import linalg as LA

A = np.matrix([[1,1],                          #choose a random matrix A
               [1,1]])
```

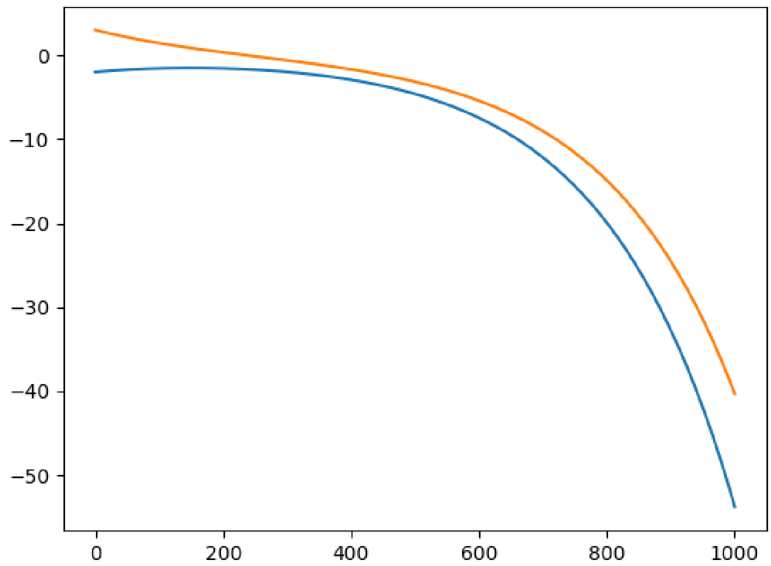
```
U = np.matrix(df)                        #U contains N real data point (row vector)

for i in range(0,50000):                 #epoch = 50000
    grad = np.matrix([[0,0],             #gradient matrix of loss function
                      [0,0]])
    z = np.matrix([[-2,3]]).transpose()  #z0
    for k in range(0,1000):              #calculate sum of gradient of each data point
        newZ = A.dot(z)
        newZ = h*newZ
        newZ = z+newZ                    # newZ = oldZ(z)+h.A.oldZ
        newG = h*(newZ-U[k].transpose()) # gradient of each point
        grad = grad+newG.dot(z.transpose()) # sum gradient of every point
        z = newZ                         # update oldZ(z)
    grad=1/N*grad                        # average of total gradient
    A = A - 0.5*grad                     # update A

print("Final solution:")
print(A)
```

Output:

```
Final solution:
[[ 2.16425972  3.74353052]
 [ 5.67963528 -2.59944283]]
```

To ensure the solution is true, we visualize data generated by this coefficient matrix using Euler method and consider if the new data is approximate to initial data:

```
A = np.matrix([[2.16425972,  3.74353052],
               [5.67963528, -2.59944283]])
ans = np.matrix([[-2,3]])
h = 0.001
for i in range(0,1000):                  #Euler method to predict 1000 point
    new = A.dot(ans[i].transpose())
    new = h*new
    new = new+ans[i].transpose()
    new = new.transpose()                #calculate Zt
    ans = np.append(ans, new, 0)         #store all the Zt to matrix ans
print(ans)
plt.plot(ans)
plt.show()
```



After training

From the graph and the new data, we can jump into conclusion that this regression method has approximated true lines and find right coefficient matrix A.

References

[1] James Stewart (2010).Second-Order Differential Equations.*Calculus: Early Transcendentals, Seventh Edition.* Cengage Learning, pp. 1142.

[2] James Stewart (2010).Second-Order Differential Equations.*Calculus: Early Transcendentals, Seventh Edition.* Cengage Learning, pp. 373.

[3] Russell Herman (2022).Matrix Formulation.*A FIRST COURSE IN DIFFERENTIAL EQUATIONS FOR SCIENTISTS AND ENGINEERS.* University of North Carolina Wilmington, pp. 260-262.

[4] Vladimir A.Zorich (2003).Continuous Mappings.*Mathematical Analysis II*, Universitext, pp. 34-40.

[5] Griffiths, D.F., Higham, D.J. (2010). Euler's Method. In: *Numerical Methods for Ordinary Differential Equations.* Springer Undergraduate Mathematics Series. Springer, London.

[6] Amir Beck(2017), *First-Order Methods in Optimization.* MOS-SIAM Series on Optimization.