ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

MÔ HÌNH HÓA TOÁN HỌC (CO2011)

Bài tập lớn

Dynamics of Love

GVHD: Nguyễn Tiến Thịnh
Nguyễn An Khương
Nguyễn Văn Minh Mẫn
Mai Xuân Toàn
Trần Hồng Tài
SV thực hiện: Trần Bảo Phúc – 2114452
Nguyễn Ngọc Phú – 2114417
Dương Phúc Thắng – 2112327
Nguyễn Trọng Tín – 2115011

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 11/2020

Mục lục

1 Danh sách thành viên và phân chia công việc

No.	Fullname	Student ID	Problems	Percentage of work
1	Nguyễn Trọng Tín	2115011	Bài tập 1: Giải hệ phương trình vi phân	20%
3	Nguyễn Ngọc Phú	2114417	Bài tập 2, viết phần Report bài 1	30%
3	Trần Bảo Phúc	2114452	Bài tập 3	25%
4	Dương Phúc Thắng	2112327	Bài tập 4	25%

2 Bài tập 1

2.1 Phương trình vi phân bậc 1 dạng tổng quát

Giả sử $F = F(t, x, y)$ là một hàm đã được định nghĩa với 3 biến và ta cần tìm hàm $u(t)$ sao cho:

$$F(t, u'(t), u(t)) = 0 \tag{1}$$

trên một miền thuộc trục t. Dạng tổng quát của phương trình này là:

$$\frac{du}{dt} = f(t, u) \tag{2}$$

Giả sử hàm f trong phương trình (2) không phụ thuộc vào u, nghiệm của phương trình trên là:

$$u(t) = \int^{t} f(t)dt + C \tag{3}$$

với C là một hằng số tùy ý.

2.2 Bài toán liên quan đến điều kiện ban đầu (Initial value problem - IVP)

Nghiệm giải ra trong ví dụ (3) không chỉ bao gồm 1 mà gồm một họ các nghiệm, mỗi nghiệm ứng với một giá trị của hằng số C. Ta có thể xác định được nghiệm duy nhất của hệ phương trình nếu có thể xác định được cụ thể giá trị của hằng số C. Một cách để thực hiện việc này là ràng buộc hàm $u(t)$ sao cho $u(t)$ không chỉ thỏa điều kiện của phương trình vi phân mà còn phải thỏa điều kiện ban đầu u_0 tại một giá trị t_0 cho trước. Ta định nghĩa ***Bài toán điều kiện ban đầu như sau:***

$$\begin{cases} \frac{du}{dt} = f(t, u) \\ u(t_0) = u_0 \end{cases} \tag{4}$$

Ở đây hàm f và điều kiện ban đầu u_0, t_0 được cho trước.

2.3 Hệ phương trình vi phân bậc nhất

2.3.1 Định nghĩa

Hệ phương trình vi phân bậc nhất có dạng:

$$\begin{cases} u_1' = f(u_1, u_2, ..., u_n, t) \\ u_2' = f(u_1, x_2, ..., u_n, t) \\ \vdots \\ u_n' = f(u_1, u_2, ..., u_n, t) \end{cases} \tag{5}$$

với t có giá trị trong một khoảng cho trước. Hệ phương trình vi phân bậc nhất với bài toán điều kiện ban đầu có dạng tương tự (5) và bổ sung thêm các điều kiện ban đầu $u_1(t_0) = c_1$, $u_2(t_0) = c_2$,..., $u_n(t_0) = c_n$.
Nghiệm của hệ phương trình vi phân bậc nhất là tập các hàm $u_1 = f_1(t)$, $u_2 = f_2(t)$,..., $u_n = f_n(t)$ thỏa tất cả các phương trình của hệ. Nghiệm của hệ phương trình vi phân bậc nhất với bài toán điều kiện ban đầu ngoài ra còn phải thỏa thêm các điều kiện ban đầu.

2.3.2 Hệ tuyến tính

Hệ phương trình vi phân tuyến tính bậc nhất là hệ phương trình vi phân bậc nhất dưới dạng:

$$\begin{cases} u_1' = a_{11}(t)u_1(t) + a_{12}(t)u_2(t) + ... + a_{1n}(t)u_n(t) + b_1(t) \\ u_2' = a_{21}(t)u_1(t) + a_{22}(t)u_2(t) + ... + a_{2n}(t)u_n(t) + b_2(t) \\ \vdots \\ u_n' = a_{n1}(t)u_1(t) + a_{n2}(t)u_2(t) + ... + a_{nn}(t)u_n(t) + b_n(t) \end{cases} \tag{6}$$

Trong đó $a_{ij}(t)$, $b_i(t)$ là các hàm xác định với $i \in [1, n]$ và $j \in [1, n]$; $u_i(t)$ là hàm cần tìm với $i \in [1, n]$.

2.3.3 Dạng ma trận của hệ tuyến tính

Ma trận hệ số của hệ (6) gồm:

$$A = \begin{pmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{pmatrix}, \vec{b}(t) = \begin{pmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_n(t) \end{pmatrix}, \vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{pmatrix}, \vec{u'} = \begin{pmatrix} u_1' \\ u_2' \\ \vdots \\ u_n' \end{pmatrix}$$

Khi đó hệ tuyến tính có thể được viết dưới dạng ma trận như sau:

$$\vec{u'} = A(t)\vec{u}(t) + b(\vec{t}) \tag{7}$$

Hệ (7) được gọi là thuần nhất nếu $\vec{b}(t) = \vec{0}$.
Bài toán điều kiện ban đầu cho hệ (11) là tìm vector hàm $\vec{u}(t)$ thỏa hệ trên một khoảng cho trước và các điều kiện ban đầu $\vec{u}(t_0) = (u_1(t_0), u_2(t_0), ..., u_n(t_0))^T$.

2.3.4 Hệ phương trình vi phân bậc một thuần nhất với hệ số hằng

Ta xét hệ

$$\vec{u'} = A\vec{u} \tag{8}$$

với t thuộc miền I, A là một ma trận $n \times n$ với các phần tử là hằng số.
Nếu λ là trị riêng của A với vector riêng $\vec{v}$, đặt $\vec{u} = e^{\lambda t}\vec{v}$. Khi đó $\vec{u'} = e^{\lambda t}\lambda\vec{v}$. Ta có:

$$A\vec{u} = e^{\lambda t}A\vec{v} = e^{\lambda t}\lambda\vec{v} = \vec{u'} \tag{9}$$

Vậy $e^{\lambda t}\vec{v}$ là một nghiệm của (9).
Giả sử ma trận A có n vector riêng độc lập tuyến tính $\vec{v_1}, \vec{v_2}, ...\vec{v_n}$ và λ_k là trị riêng tương ứng với vector riêng $\vec{v_k}$. Khi đó nghiệm tổng quát là:

$$\vec{u} = c_1e^{\lambda_1 t}\vec{v_1} + c_2e^{\lambda_2 t}\vec{v_2} + ... + c_ne^{\lambda_n t}\vec{v_n} \tag{10}$$

với $c_1, c_2, ..., c_n$ là các hằng số tùy ý.
Khi đó, tập nghiệm cơ bản là $e^{\lambda_1 t}\vec{v_1}$, $e^{\lambda_2 t}\vec{v_2}$,...,$e^{\lambda_n t}\vec{v_n}$. Ta gọi $U(t)$ là một *ma trận cơ bản* có các cột tương ứng với các phần tử trong tập nghiệm cơ bản.
Vậy nghiệm tổng quát của phương trình (8) là $u(t) = U(t)\vec{c}$, với $\vec{c} = (c_1, c_2, ..., c_n)^T$ và nghiệm thỏa điều kiện ban đầu $\vec{u}(t_0) = \vec{u_0}$ là $u(t) = U(t)U(t_0)^{-1}\vec{u_0}$.

2.4 Hệ động lực

2.4.1 Định nghĩa

Các mô hình toán học của các hệ thống khoa học thường cho ra các phương trình vi phân mà ở đó thời gian là biến độc lập. Các hệ này xuất hiện trong các lĩnh vực như thiên văn học, sinh học, hóa học, kinh tế học, kỹ thuật, vật lý học. Hệ các phương trình vi phân:

$$\vec{u'} = f(u) \tag{11}$$

tạo ra một ***hệ động lực học***. Trong đó cần chú ý những vấn đề sau:

- Các hành vi liên quan đến tính chất của hệ được thể hiện trên một khoảng thời gian dài. Hay nói cách khác, cần xem xét đến tính động lực khi biến thời gian $t \to +\infty$.
- Sự thay đổi của nghiệm thu được từ hệ động lực khi các dữ liệu đầu vào thay đổi. Xét hệ (11) cho tất cả các phương trình vi phân với giá trị ban đầu $u(t_0) = u_0$, nghiệm không chỉ là một hàm theo biến t mà còn là một hàm theo biến $u_0 : u = \phi(t, u_0)$
- Họ các phương trình vi phân. Các phương trình của một họ thường được phân biệt bởi giá trị của một thông số, ví dụ như λ. Vế phải của hệ phương trình (11) được viết lại thành $f(u, \lambda)$. Nghiệm của hệ giờ đây phụ thuộc thêm vào λ.

Những điều cần lưu ý trên thể hiện các vấn đề trong quá trình ứng dụng hệ động lực học, tại đây phụ thuộc vào các tham số và cần phải được xem xét trên nhiều điều kiện ban đầu khác nhau. Như vậy, hệ động lực là một hệ thống thay đổi theo thời gian dựa vào một tập các ràng buộc nhất định để xác định cách chuyển trạng thái trong hệ thống. Nói cách khác, hệ động lực là một *không gian trạng thái (state space)* cùng với cách biến đổi của không gian đó.
Một hệ động lực gồm 2 phần:

- *Vector trạng thái (state vector)* mô tả chính xác các trạng thái của hệ thống.
- *Hàm trạng thái (state function)* mô tả cách chuyển trạng thái .

Vector trạng thái có thể được mô tả bởi:

$$\vec{u}(t) = [u_1(t), u_2(t), ...u_n(t)]$$

Hàm trạng thái có thể được mô tả bởi:

$$f_1(u_1, u_2, ...u_n), f_2(u_1, u_2, ...u_n), ..., f_n(u_1, u_2, ...u_n)$$

2.4.2 Phân loại

1. ***Tự động và bất tự động***
   Hệ các phương trình vi phân như (11) được gọi là một hệ ***tự động (autonomous)*** vì vế phải của hệ không phụ thuộc vào biến thời gian.
   Ngoài ra có một dạng hệ tổng quát hơn, ***bất tự động (non-autonomous)***:

   $$x' = f(t, x) \tag{12}$$

   Từ hệ phương trình vi phân (12) gồm n phương trình này có thể đưa về một hệ tự động gồm $n + 1$ phương trình bằng cách thêm vào hệ phương trình $x'_{n+1} = 1$ và thay $t = x_{n+1}$ ở n phương trình còn lại.

2. ***Rời rạc và liên tục***
   Hệ động lực ***liên tục*** là một hệ thay đổi trạng thái trong không gian trạng thái của hệ một cách liên tục. Hệ động lực ***rời rạc*** là một hệ thay đổi trạng thái trong không gian trạng thái của hệ một cách rời rạc.

2.5 Mô hình hệ động lực trong Bài tập lớn

Trong bài tập lớn này, chúng ta đề cập đến một mô hình autonomous đơn giản có hệ số hằng và điều kiện ban đầu. Mô hình này như sau:

$$\begin{cases} R' = aR + bJ \\ J' = cR + dJ \\ R(0) = R_0 \\ J(0) = J_0 \end{cases} \tag{13}$$

Trong đó:
$R : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$ (*hàm đại diện tình yêu của Romeo dành cho Juliet).*
$J : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$(*hàm đại diện tình yêu của Juliet dành cho Romeo).*
a, b, c, $d \in \mathbb{R}$: mô tả sự tương tác tình yêu của một người đến người còn lại.
R_0, J_0: lần lượt là tình yêu của Romeo dành cho Juliet và của Juliet dành cho Romeo tại thời điểm ban đầu.
Hệ phương trình vi phân trên có nghiệm duy nhất khi và chỉ khi hệ có điều kiện ban đầu. Ta sẽ giải cụ thể sau đây.

2.6 Giải hệ phương trình

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

Ta có ma trận hệ số $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$$det(A - \lambda I) = \begin{pmatrix} a-\lambda & b \\ c & d-\lambda \end{pmatrix} = (a-\lambda)(d-\lambda) - bc = \lambda^2 - (a+d)\lambda + ad - bc = 0 \tag{14}$$

Ta có $\Delta = (a+d)^2 - 4(ad-bc)$. Ta xét 3 trường hợp:

- Trường hợp 1: $\Delta > 0$, phương trình có hai nghiệm thực phân biệt $\lambda_1 < \lambda_2$. Khi đó, nghiệm của hệ là:

$$\begin{cases} R(t) = e^{\lambda_1 t}\frac{R_0(\lambda_2 - a) - J_0 b}{\lambda_2 - \lambda_1} - e^{\lambda_2 t}\frac{R_0(\lambda_1 - a) - J_0 b}{\lambda_2 - \lambda_1} \\ J(t) = e^{\lambda_1 t}\frac{[R_0(\lambda_2 - a) - J_0 b](\lambda_1 - a)}{b(\lambda_2 - \lambda_1)} - e^{\lambda_2 t}\frac{[R_0(\lambda_1 - a) - J_0 b](\lambda_2 - a)}{b(\lambda_2 - \lambda_1)} \end{cases}$$

- Trường hợp 2: $\Delta < 0$, phương trình có hai nghiệm phức $\lambda_{1,2} = \alpha \pm \beta i$. Khi đó, nghiệm của

hệ là:

$$\begin{cases} R(t) = e^{\alpha t}\left(R_0 \cos \beta t + \frac{J_0 b - (\alpha - a)R_0}{\beta} \sin \beta t\right) \\ J(t) = e^{\alpha t}\left(J_0 \cos \beta t + \frac{J_0 b(\alpha - a) - (\alpha - a)^2 R_0 - \beta^2 R_0}{b\beta} \sin \beta t\right) \end{cases}$$

- Trường hợp 3: $\Delta = 0$, phương trình có nghiệm kép $\lambda_1 = \lambda_2 = \lambda$. Khi đó, nghiệm của hệ là:

$$\begin{cases} R(t) = e^{\lambda t}\big((J_0 b - (\lambda - a)R_0)t + R_0\big) \\ J(t) = e^{\lambda t}\left(J_0 + \big(J_0(\lambda - a) - \frac{(\lambda - a)^2 R_0}{b}\big)t\right) \end{cases}$$

2.7 Bảng phân loại phase portrait

Phase portrait của hệ (13) được xây dựng bằng cách vẽ đồ thị ***vector field*** $\vec{x'} = (R', J')^T = (f(R, J), g(R, J))^T$ đối với các biến độc lập R, J trong $RJ - plane$. Một ***trajectory*** trong vector field đại diện cho đồ thị của một bài toán có điều kiện ban đầu xác định. Việc xây dựng phase portrait cho phép chúng ta tìm tất cả các đồ thị khác nhau của một hệ động lực với các điều kiện ban đầu khác nhau.
Điểm cố định của hệ (13) được xác định như sau:

$$\begin{cases} R' = 0 \\ J' = 0 \end{cases}$$

Việc phân loại các điểm cố định này sẽ cho ta các phase portrait tương ứng. Ta sẽ căn cứ vào 2 thông số $\det(A) = ad - bc$ và $trace(A) = a + d$ ở phương trình trị riêng (14).

- Trường hợp 1: $\det(A) < 0$
  Vì $\det(A) = \lambda_1 \lambda_2 < 0$ nên phương trình trị riêng có 2 nghiệm thực phân biệt λ_1, λ_2 trái dấu và điểm cố định là duy nhất. Điểm cố định là điểm yên ngựa ***(Saddle node)***, không ổn định ***(Unstable)***.

- Trường hợp 2: $\det(A) = 0$
  Ở trường hợp này, phương trình trị riêng có 2 nghiệm thực λ_1, λ_2 và điểm cố định không là duy nhất mà tạo thành 1 đường thẳng (line) hay toàn bộ mặt phẳng (plane). Cụ thể gồm 3 trường hợp sau:
  - Nếu $trace(A) < 0$: ***Line of stable fixed points***.
  - Nếu $trace(A) = 0$: ***Plane of fixed points***.
  - Nếu $trace(A) > 0$: ***Line of unstable fixed points***.

- Trường hợp 3: $\det(A) > 0$
  Ở trường hợp này, điểm cố định là duy nhất. Ta xét các trường hợp con:
  - Nếu $trace(A) < -\sqrt{4\det(A)}$
    Phương trình trị riêng có 2 nghiệm thực phân biệt và cùng âm $\lambda_1 < 0, \lambda_2 < 0$. Điểm cố định là ***Stable node*** và có tính chất hút (attracting sink) .
  - Nếu $trace(A) = -\sqrt{4\det(A)}$
    Phương trình trị riêng có 2 nghiệm thực bằng nhau $\lambda_1 = \lambda_2 < 0$, tại đây ta lại xét 2 trường hợp nhỏ hơn:

  - Nếu tồn tại một vector riêng được xác định duy nhất thì điểm cố định là ***Stable degenerate node***.
  - Nếu không tồn tại một vector riêng được xác định duy nhất thì điểm cố định là ***Stable start***.
- Nếu $-\sqrt{4\det(A)} < trace(A) < 0$
  Phương trình trị riêng có 2 nghiệm phức liên hợp $\lambda_{1,2} = \alpha \pm \beta i$ và phần thực $\alpha < 0$. Điểm cố định là ***Stable spiral*** và có tính chất hút (attracting sink).
- Nếu $trace(A) = 0$
  Phương trình trị riêng có 2 nghiệm thuần ảo $\lambda_{1,2} = \pm\beta i$. Điểm cố định là ***Center***.
- Nếu $0 < trace(A) < \sqrt{4\det(A)}$
  Phương trình trị riêng có 2 nghiệm phức liên hợp $\lambda_{1,2} = \alpha \pm \beta i$ và phần thực $\alpha > 0$. Điểm cố định là ***Unstable spiral*** và có tính chất đẩy (repelling source).
- Nếu $trace(A) = \sqrt{4\det(A)}$
  Phương trình trị riêng có 2 nghiệm thực bằng nhau $\lambda_1 = \lambda_2 > 0$, tại đây ta lại xét 2 trường hợp nhỏ hơn:
  - Nếu tồn tại một vector riêng được xác định duy nhất thì điểm cố định là ***Unstable degenerate node***.
  - Nếu không tồn tại một vector riêng được xác định duy nhất thì điểm cố định là ***Unstable start***.
- Nếu $\sqrt{4\det(A)} < trace(A)$
  Phương trình trị riêng có 2 nghiệm thực phân biệt và cùng dương $\lambda_1 > 0, \lambda_2 > 0$. Điểm cố định là ***Unstable node*** và có tính chất đẩy (repelling source).

Sau khi phân loại các điểm cố định, ta tổng hợp lại thành một bảng phân loại phase portarit tương ứng như sau:

Re λ_1	Re λ_2	|Im λ_1|	| Im λ_2|	Type
+	-	0	0	Saddle
-	-	0	0	Stable
+	+	0	0	Unstable
-	-	+	+	Stable spiral
0	0	+	+	Center
+	+	+	+	Unstable spiral

3 Bài tập 2

3.1 Eager Beaver

Điều kiện của a, b trong phong cách này là $a > 0$ và $b > 0$. Ta xét 2 ví dụ cùng ma trận hệ số nhưng điều kiện ban đầu khác nhau.

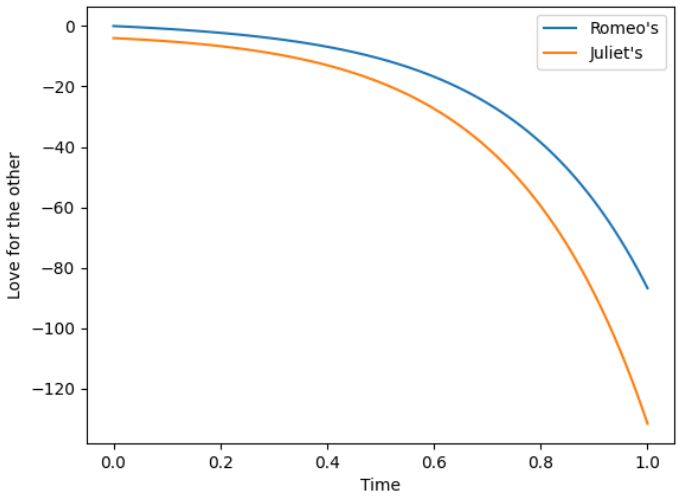
3.1.1 Ví dụ 1

$$\begin{cases} R' = R + 2J \\ J' = 3R + 2J \\ R(0) = 0 \\ J(0) = -4 \end{cases}$$

Theo bài 1, ta có $\Delta = 25 > 0$, hệ có 2 trị riêng phân biệt $\lambda_1 = -1, \lambda_2 = 4$. Áp dụng công thức nghiệm chính xác ở trường hợp 1:

$$\begin{cases} R(t) = \frac{8}{5}e^{-t} - \frac{8}{5}e^{4t} \\ J(t) = -\frac{8}{5}e^{-t} + -\frac{12}{5}e^{4t} \end{cases}$$

Biểu diễn R (eager beaver) và J (eager beaver) trên cùng một đồ thị theo biến thời gian t (hình 1):



Hình 1: *The love between two eager beaver*

Giải thích các đoạn code trong *Python* vẽ đồ thị ví dụ trên (các đoạn code này được ta định nghĩa trong file **trajectory.py** và sẽ được dùng để vẽ **phase portrait**) như sau:

- Nạp các modules cần thiết, bao gồm: **numpy** (module toán học xử lý ma trận và mảng), **matplotlib.pyplot**(module đồ họa vẽ đồ thị), **math** và khai báo ma trận hệ số (hình 2).

```python
import numpy as np
import matplotlib.pyplot as plt
import math

pi = math.pi
a = 1
b = 2
c = 3
d = 2
R0 = 0
J0 = -4
```

Hình 2: *Modules và khai báo*

```python
def delta(a, b, c, d):
    return pow((a + d), 2) - 4 * (a * d - b * c)


def lamda(a, b, c, d):
    deltaValue = delta(a, b, c, d)
    if deltaValue > 0:
        deltaSqrt = math.sqrt(deltaValue)
        lamda1 = (a + d - deltaSqrt) / 2
        lamda2 = (a + d + deltaSqrt) / 2
        return [lamda1, lamda2]
    elif deltaValue == 0:
        x = (a + d) / 2
        return x
    else:
        deltaValue = abs(deltaValue)
        deltaSqrt = math.sqrt(deltaValue)
        a1 = (a + d) / 2
        b1 = deltaSqrt / 2
        return [a1, b1]
```

Hình 3: Δ *và* λ

- Viết hàm tính Δ và λ (hình 3).
- Viết các hàm phụ tính các hằng số trong công thức tương ứng với 3 trường hợp (hình 4).

```python
def F1(a, b, R0, J0, k1, k2):
    D = k2 - k1
    D1 = R0 * (k2 - a) - J0 * b
    D2 = -R0 * (k1 - a) + J0 * b
    return [D1 / D, D2 / D]


def F2(a, b, R0, J0, k):
    x = J0 * b - (k - a) * R0
    return x


def F3(a, b, R0, J0, m, n):
    x = ((b * J0) - ((m - a) * R0)) / n
    return x
```

Hình 4: *Các hàm phụ trợ tính hệ số*

- Hàm tính giá trị của R (hình 5).

```python
def fR(t, a, b, c, d, R0, J0):
    deltaValue = delta(a, b, c, d)
    if deltaValue > 0:
        k1 = lamda(a, b, c, d)[0]
        k2 = lamda(a, b, c, d)[1]
        c1 = F1(a, b, R0, J0, k1, k2)[0]
        c2 = F1(a, b, R0, J0, k1, k2)[1]
        R = c1 * math.exp(k1 * t) + c2 * math.exp(k2 * t)
        return R
    elif deltaValue == 0:
        k = lamda(a, b, c, d)
        c2 = F2(a, b, R0, J0, k)
        R = R0 * math.exp(k * t) + c2 * math.exp(k * t) * t
        return R
    else:
        m = lamda(a, b, c, d)[0]
        n = lamda(a, b, c, d)[1]
        x = F3(a, b, R0, J0, m, n)
        R = math.exp(m * t) * (R0 * math.cos((n * t)) + x * math.sin((n * t)))
        return R
```

Hình 5: *Hàm tính giá trị R theo t*

- Hàm tính giá trị của J (hình 6).

```python
def fJ(t, a, b, c, d, R0, J0):
    deltaValue = delta(a, b, c, d)
    if deltaValue > 0:
        k1 = lamda(a, b, c, d)[0]
        k2 = lamda(a, b, c, d)[1]
        c1 = F1(a, b, R0, J0, k1, k2)[0]
        c2 = F1(a, b, R0, J0, k1, k2)[1]
        R = fR(t, a, b, c, d, R0, J0)
        dRdt = c1 * k1 * math.exp(k1 * t) + c2 * k2 * math.exp(k2 * t)
        J = (1 / b) * (dRdt - a * R)
        return J
    elif deltaValue == 0:
        k = lamda(a, b, c, d)
        c2 = F2(a, b, R0, J0, k)
        J = (1 / b) * math.exp(k * t) * (J0 * b + (k - a) * c2 * t)
        return J
    else:
        m = lamda(a, b, c, d)[0]
        n = lamda(a, b, c, d)[1]
        c2 = F3(a, b, R0, J0, m, n)
        R = fR(t, a, b, c, d, R0, J0)
        dRdt = m * math.exp(m * t) * (R0 * math.cos((n * t)) + c2 * math.sin((n * t))) + \
            math.exp(m * t) * (- (R0 * n * math.sin((n * t))) +
                               (c2 * n * math.cos((n * t))))
        J = (1 / b) * (dRdt - a * R)
        return J
```

Hình 6: *Hàm tính giá trị J theo t*

- Lấy 100 giá trị của biến thời gian t trong khoảng $[0, 1]$ để vẽ đồ thị (hình 7).

```python
t = np.linspace(0, 1, 100)
Rt = np.vectorize(fR)
Jt = np.vectorize(fJ)
plt.plot(t, Rt(t, a, b, c, d, R0, J0))
plt.plot(t, Jt(t, a, b, c, d, R0, J0))
plt.xlabel("Time")
plt.ylabel("Love for the other")
plt.legend(["Romeo's", "Juliet's"])
plt.show()
```

Hình 7: *Vẽ đồ thị*

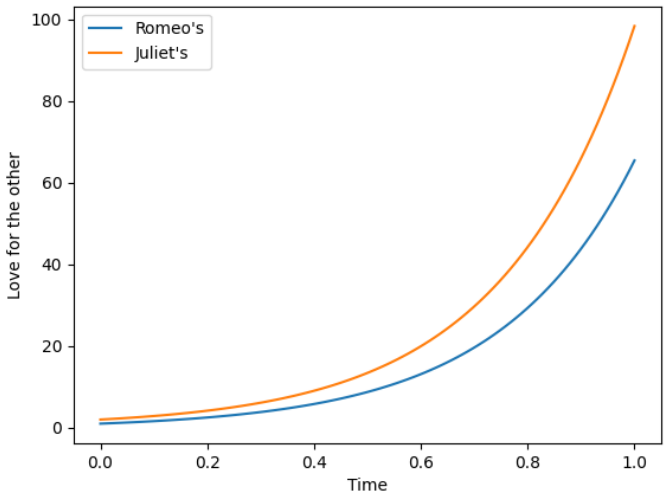
3.1.2 Ví dụ 2

$$\begin{cases} R' = R + 2J \\ J' = 3R + 2J \\ R(0) = 1 \\ J(0) = 2 \end{cases}$$

Tương tự ví dụ 1 nhưng $R_0 = 1, J_0 = 2$ Áp dụng công thức nghiệm chính xác ở trường hợp 1:

$$\begin{cases} R(t) = -\frac{1}{5}e^{-t} + \frac{6}{5}e^{4t} \\ J(t) = \frac{1}{5}e^{-t} + \frac{9}{5}e^{4t} \end{cases}$$

Đồ thi biểu diễn như sau (hình 8):



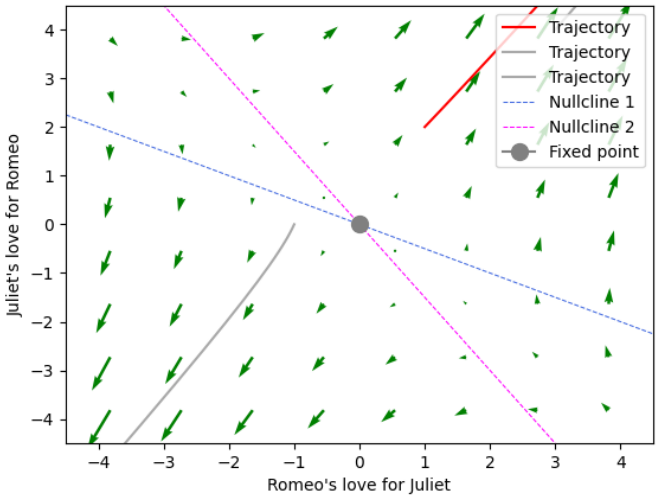
Hình 8: *The love between two eager beaver*

3.1.3 Phase portrait

Ứng với hệ phương trình trong 2 ví dụ trên với các điều kiện ban đầu bất kì $\begin{cases} R' = R + 2J \\ J' = 3R + 2J \end{cases}$

ta có phase portrait trong mặt phẳng RJ được biểu diễn trong hình 9
Giải thích các đoạn code trong *Python* vẽ phase portrait:

- Nạp các modules cần thiết, bao gồm **numpy**, **matplotlib.pyplot**, package **odeint** trong module **scipy.integrate** (dùng để giải hệ phương trình vi phân), các thông số a, b, c, d, R_0, J_0 trong module **trajectory** (module này đã được ta định nghĩa). Ta sẽ vẽ các trajectory ứng với các bộ giá trị ban đầu $[R_0, J_0], [R_0 + 2, J_0 + 2], [R_0 - 2, J_0 - 2]$ (hình 10).
- Viết hàm biểu diễn hệ phương trình vi phân (hình 11).



Hình 9: *The phase portrait of the love between two eager beaver*

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from trajectory import a, b, c, d, R0, J0
Sts = [[R0, J0], [R0 + 2, J0 + 2], [R0 - 2, J0 - 2]]
```

Hình 10: *Modules, pakage và khai báo các bộ giá trị ban đầu*

```
def ivpSys(s, t, a, b, c, d):
    R, J = s
    dRdt = a * R + b * J
    dJdt = c * R + d * J
    return [dRdt, dJdt]
```

Hình 11: *Hàm biểu diễn hệ phương trình vi phân*

```
y1 = np.linspace(-6, 6, 12)
y2 = np.linspace(-6, 6, 12)
Y1, Y2 = np.meshgrid(y1, y2)
t = 0
u, v = np.zeros(Y1.shape), np.zeros(Y2.shape)
NI, NJ = Y1.shape
for i in range(NI):
    for j in range(NJ):
        x = Y1[i, j]
        y = Y2[i, j]
        yvector = ivpSys([x, y], t, a, b, c, d)
        u[i, j] = yvector[0]
        v[i, j] = yvector[1]
Q = plt.quiver(Y1, Y2, u, v, color='g')
```

Hình 12: *Vẽ các vector biểu diễn R' và J'*

- Vẽ các vector biểu diễn R' và J' tại thời điểm ban đầu $t = 0$ của một tập gồm 144 điểm rời rạc (R_0, J_0) phân bố đều nhau trên mặt phẳng RJ (hình 12).

- Vẽ các trajectory với 200 giá trị của biến thời gian t từ $(0, 5)$ ứng với 3 bộ giá trị ban đầu: (R_0, J_0) (màu đỏ) và $(R_0 + 2, J_0 + 2), (R_0 - 2, J_0 - 2)$ (màu xám) (hình 13).

```
tspan = np.linspace(0, 5, 200)
for elementSt, St in enumerate(Sts):
    sol = odeint(ivpSys, St, tspan, args=(a, b, c, d))
    if elementSt == 0:
        color = "r"
    else:
        color = "darkgray"
    plt.plot(sol[:, 0], sol[:, 1], color, label='Trajectory')
```

Hình 13: *Vẽ các trajectory*

- Ký hiệu các chú thích và vẽ phase portrait (hình 14).

```
x = np.linspace(-4.5, 4.5, 100)


def fx(x, a, b):
    return -a * x / b


plt.plot(x, fx(x, a, b), linestyle='dashed', linewidth=0.75,
         color='royalblue', label='Nullcline 1')
plt.plot(x, fx(x, c, d), linestyle='dashed', linewidth=0.75,
         color='magenta', label='Nullcline 2')

plt.plot(0, 0, "red", marker="o", markersize=10.0,
         color='grey', label='Fixed point')

plt.xlabel("Romeo's love for Juliet")
plt.ylabel("Juliet's love for Romeo")
plt.legend()
plt.xlim([-4.5, 4.5])
plt.ylim([-4.5, 4.5])
plt.show()

```

Hình 14: *Vẽ phase portrait*

3.2 Narcissistic Nerd

Điều kiện của a, b trong phong cách này là $a > 0$ và $b < 0$. Ta xét 2 ví dụ cùng ma trận hệ số nhưng điều kiện ban đầu khác nhau. Đoạn code trong *Python* dùng để vẽ các đồ thị và phase portrait tương tự mục Eager Beaver, chỉ thay đổi tham số a, b, c, d, R_0, J_0.

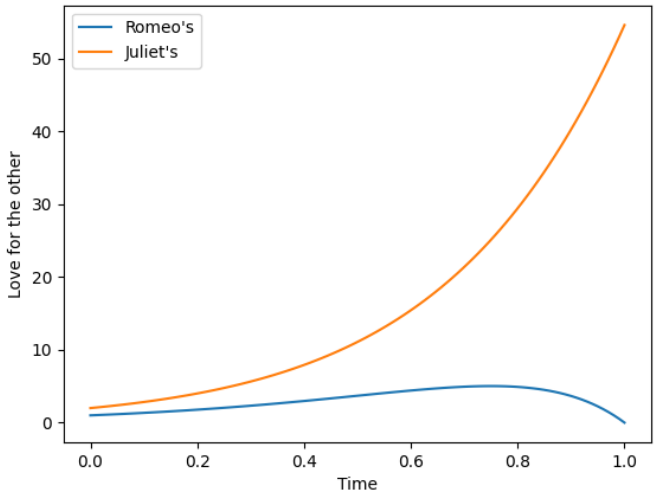
3.2.1 Ví dụ 1

$$\begin{cases} R' = 5R - J \\ J' = R + 3J \\ R(0) = 1 \\ J(0) = 2 \end{cases}$$

Theo bài 1, ta có $\Delta = 0$, hệ có 1 trị riêng duy nhất $\lambda = 4$. Áp dụng công thức nghiệm chính xác ở trường hợp 3:

$$\begin{cases} R(t) = e^{4t}(-t+1) \\ J(t) = e^{4t}(-t+2) \end{cases}$$

Biểu diễn R (narcissistic nerd) và J (eager beaver) trên cùng một đồ thị theo biến thời gian t (hình 15):



Hình 15: *The love between a narcissistic nerd and a eager beaver*

3.2.2 Ví dụ 2

$$\begin{cases} R' = 5R - J \\ J' = R + 3J \\ R(0) = 1 \\ J(0) = 3 \end{cases}$$

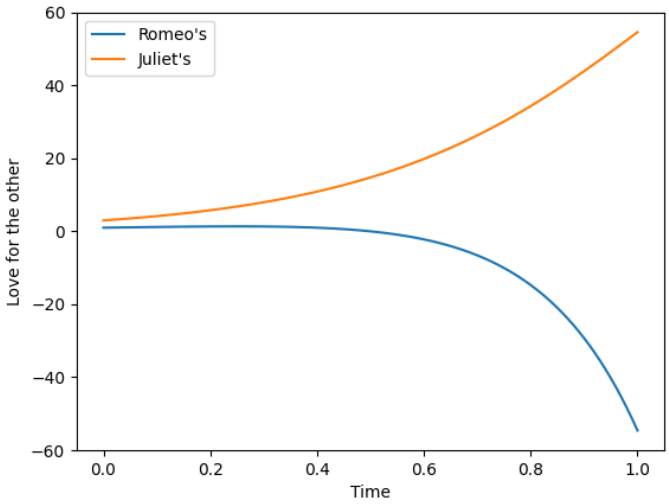
Tương tự ví dụ 1 nhưng $R_0 = 1, J_0 = 3$ Áp dụng công thức nghiệm chính xác ở trường hợp 3:

$$\begin{cases} R(t) = e^{4t}(-2t+1) \\ J(t) = e^{4t}(-2t+3) \end{cases}$$

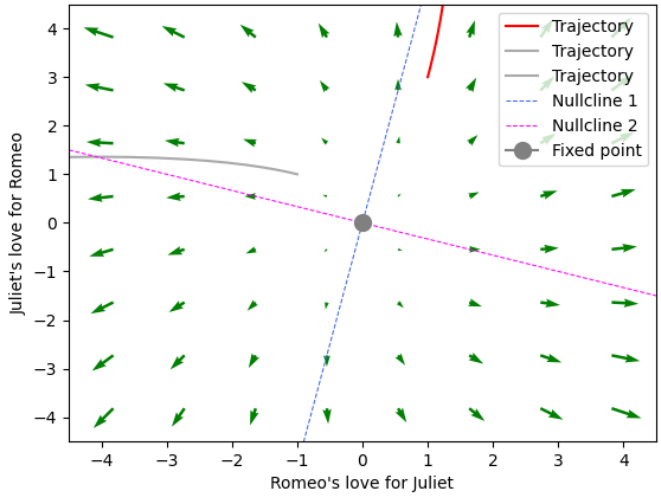
Đồ thị biểu diễn ở hình 16.

3.2.3 Phase portrait

Ứng với hệ phương trình trong 2 ví dụ trên với các điều kiện ban đầu bất kì $\begin{cases} R' = 5R - J \\ J' = R + 3J \end{cases}$ ta có phase portrait trong mặt phẳng RJ được biểu diễn trong hình 17.



Hình 16: *The love between a narcissistic nerd and a eager beaver*



Hình 17: *The phase portrait of the love between a narcissistic nerd and a eager beaver*

3.3 Cautious Lover

Điều kiện của a, b trong phong cách này là $a < 0$ và $b > 0$. Ta xét 2 ví dụ cùng ma trận hệ số nhưng điều kiện ban đầu khác nhau. Đoạn code trong *Python* dùng để vẽ các đồ thị và phase portrait tương tự mục Eager Beaver, chỉ thay đổi tham số a, b, c, d, R_0, J_0.

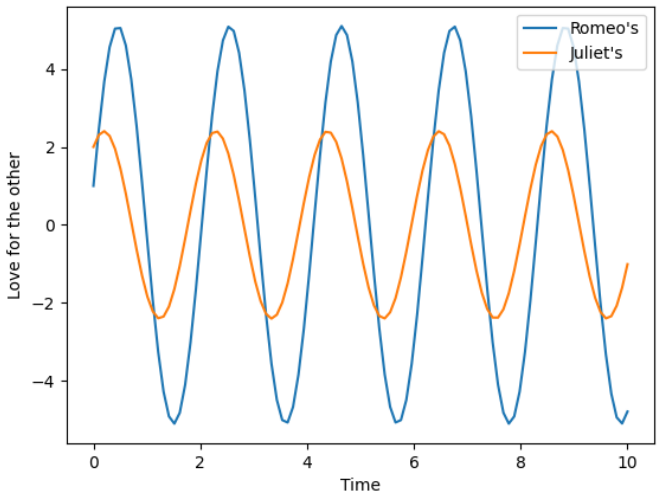
3.3.1 Ví dụ 1

$$\begin{cases} R' = -3R + 9J \\ J' = -2R + 3J \\ R(0) = 1 \\ J(0) = 2 \end{cases}$$

Theo bài 1, ta có $\Delta = -36 < 0$, hệ có 2 trị riêng phức liên hợp $\lambda_{1,2} = \pm 3i$. Áp dụng công thức nghiệm chính xác ở trường hợp 2:

$$\begin{cases} R(t) = \cos 3t + 5 \sin 3t \\ J(t) = 2 \cos 3t + \frac{4}{3} \sin 3t \end{cases}$$

Biểu diễn R (cautious lover) và J (narcissistic nerd) trên cùng một đồ thị theo biến thời gian t (hình 18):



Hình 18: *The love between a cautious lover and a narcissistic nerd*

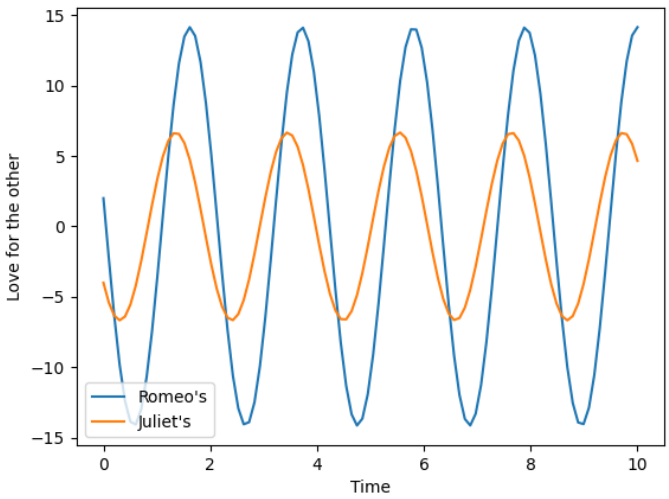
3.3.2 Ví dụ 2

$$\begin{cases} R' = -3R + 9J \\ J' = -2R + 3J \\ R(0) = 2 \\ J(0) = -4 \end{cases}$$

Tương tự ví dụ 1 nhưng $R_0 = 2, J_0 = -4$ Áp dụng công thức nghiệm chính xác ở trường hợp 2:

$$\begin{cases} R(t) = 2 \cos 3t - 14 \sin 3t \\ J(t) = -4 \cos 3t - \frac{16}{3} \sin 3t \end{cases}$$

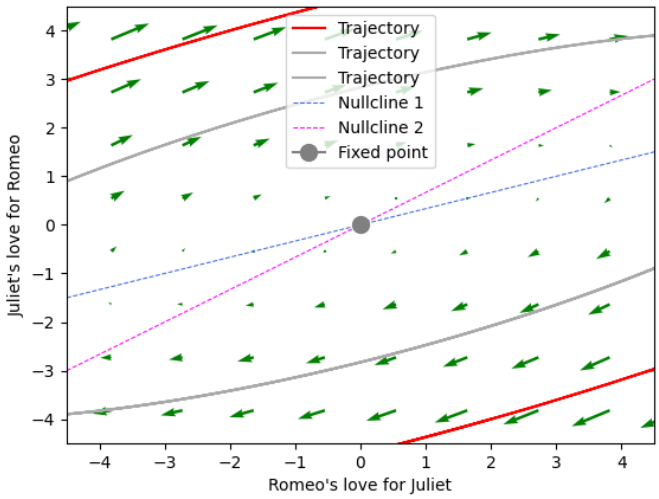
Đồ thị biểu diễn ở hình 19.



Hình 19: *The love between a cautious lover and a narcissistic nerd*

3.3.3 Phase portrait

Ứng với hệ phương trình trong 2 ví dụ trên với các điều kiện ban đầu bất kì $\begin{cases} R' = -3R + 9J \\ J' = -2R + 3J \end{cases}$ ta có phase portrait trong mặt phẳng RJ được biểu diễn trong hình 20.



Hình 20: *The phase portrait of the love between a cautious lover and a narcissistic nerd*

3.4 Hermit

Điều kiện của a, b trong phong cách này là $a < 0$ và $b < 0$. Ta xét 2 ví dụ cùng ma trận hệ số nhưng điều kiện ban đầu khác nhau. Đoạn code trong *Python* dùng để vẽ các đồ thị và phase portrait tương tự mục Eager Beaver, chỉ thay đổi tham số a, b, c, d, R_0, J_0.

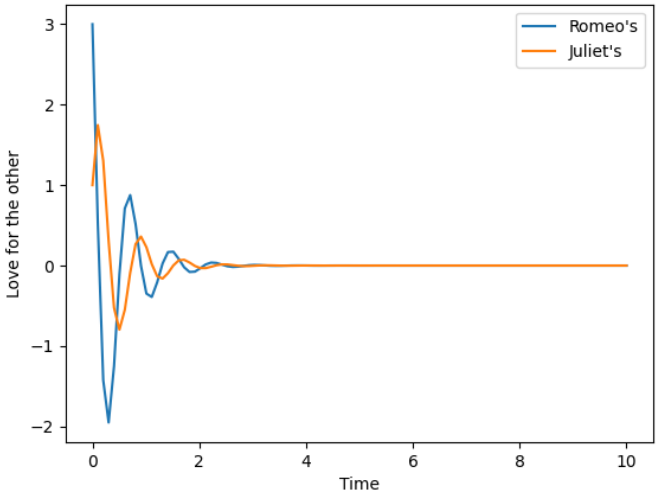
3.4.1 Ví dụ 1

$$\begin{cases} R' = -3R - 13J \\ J' = 5R - J \\ R(0) = 3 \\ J(0) = 1 \end{cases}$$

Theo bài 1, ta có $\Delta = -256 < 0$, hệ có 2 trị riêng phức liên hợp $\lambda_{1,2} = -2 \pm 8i$. Áp dụng công thức nghiệm chính xác ở trường hợp 2:

$$\begin{cases} R(t) = e^{-2t}(3\cos 8t - 2\sin 8t) \\ J(t) = e^{-2t}(\cos 8t + 2\sin 8t) \end{cases}$$

Biểu diễn R (hermit) và J (cautious lover) trên cùng một đồ thị theo biến thời gian t (hình 21):



Hình 21: *The love between a hermit and a cautious lover*

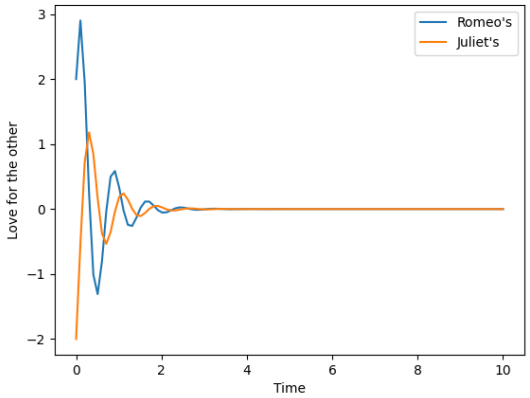
3.4.2 Ví dụ 2

$$\begin{cases} R' = -3R - 13J \\ J' = 5R - J \\ R(0) = 2 \\ J(0) = -2 \end{cases}$$

Tương tự ví dụ 1 nhưng $R_0 = 2, J_0 = -2$ Áp dụng công thức nghiệm chính xác ở trường hợp 2:

$$\begin{cases} R(t) = e^{-2t}(2\cos 8t + 3\sin 8t) \\ J(t) = e^{-2t}(-2\cos 8t + \sin 8t) \end{cases}$$

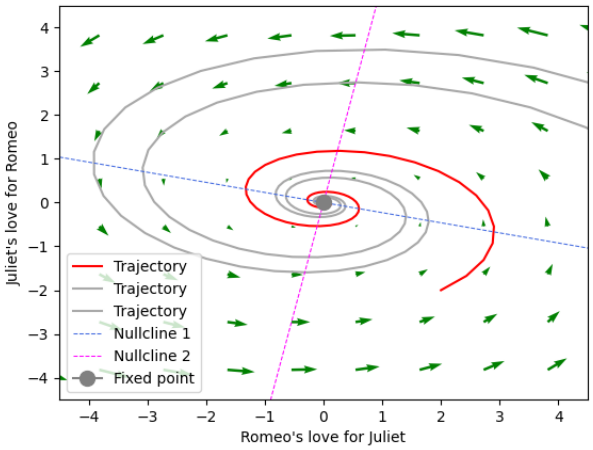
Đồ thị biểu diễn ở hình 22.



Hình 22: *The love between a hermit and a cautious lover*

3.4.3 Phase portrait

Ứng với hệ phương trình trong 2 ví dụ trên với các điều kiện ban đầu bất kì $\begin{cases} R' = -3R - 13J \\ J' = 5R - J \end{cases}$ ta có phase portrait trong mặt phẳng RJ được biểu diễn trong hình 23.



Hình 23: *The phase portrait of the love between a hermit and a cautious lover*

4 Bài tập 3

4.1 Hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng (Nonhomogeneous Linear System)

Khi tình yêu giữa Romeo và Juliet phụ thuộc vào điều kiện bên ngoài (như gia đình, xã hội,..) ta sẽ có hệ phương trình tổng quát như sau:

$$\begin{cases} R' = aR + bJ + f(t) \\ J' = cR + dJ + g(t) \\ R(0) = R_0, J(0) = J_0 \end{cases} \tag{15}$$

Trong đó:
$R : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$ (*hàm đại diện tình yêu của Romeo dành cho Juliet).*
$J : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$(*hàm đại diện tình yêu của Juliet dành cho Romeo).*
a, b, c, $d \in \mathbb{R}$: mô tả sự tương tác tình yêu của một người đến người còn lại.
R_0, J_0: lần lượt là tình yêu của Romeo dành cho Juliet và của Juliet dành cho Romeo tại thời điểm ban đầu.
$f(t)$, $g(t)$ là hai hàm theo biến thời gian t, lần lượt là điều kiện bên ngoài ảnh hưởng tới tình yêu của Romeo và Juliet.

Hệ (15) có dạng hệ phương trình vi phân tuyến tính không thuần nhất. Ta biểu diễn lại hệ (15) dưới dạng tổng quát:

$$\begin{cases} x' = Ax + B(t) \\ x(0) = x_0 \end{cases} \tag{16}$$

Để tìm nghiệm tổng quát cho hệ (15), ta sẽ tìm cách giải và nghiệm tổng quát của hệ (16).

4.1.1 Hàm mũ ma trận (Matrix Exponential)

4.1.1.a Định nghĩa

Cho A là một ma trận $n \times n$ với hệ số không đổi. Hàm mũ ma trận của A (kí hiệu là e^{At}) được biểu diễn như sau:

$$e^{At} = \sum_{k=0}^{\infty} A^k \frac{t^k}{k!} = I_n + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + ...$$

4.1.1.b Áp dụng hàm mũ ma trận trong tìm nghiệm hệ phương trình vi phân trình vi phân tuyến tính thuần nhất

Cho hệ phương trình vi phân tuyến tính thuần nhất:

$$\begin{cases} x' = Ax \\ x(0) = x_0 \end{cases} \tag{17}$$

Giả sử e^{At} là hàm mũ ma trận của A, ta được nghiệm tổng quát của hệ (17):

$$x = e^{At}x_0 \tag{18}$$

Thật vậy, ta có thể chứng minh công thức trên như sau:
Dựa và định nghĩa ta có:

$$e^{At} = I_n + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + ...$$
$$\Rightarrow \frac{d}{dt}(e^{At}) = A + A^2t + \frac{1}{2}A^3t^2 + ...$$
$$\Rightarrow \frac{d}{dt}(e^{At}) = A(I_n + At + \frac{1}{2}A^2t^2 + ...)$$
$$\Rightarrow \frac{d}{dt}(e^{At}) = Ae^{At}$$

Điều này cho thấy e^{At} là nghiệm của hệ phương trình vi phân tuyến tính $x' = Ax$. Nghiệm x lúc này được biểu diễn dưới dạng $x = e^{At}.c$ (với c là hằng số). Kết hợp với điều kiện ban đầu $x(0) = x_0$, ta được nghiệm tổng quát của hệ (17):

$$x = e^{At}x_0$$

4.1.1.c Phương pháp tìm hàm mũ ma trận

Như ta chứng minh ở trên, nếu có hàm mũ của ma trận A thì ta sẽ tìm được nghiệm của hệ phương trình vi phân tuyến tính thuần nhất. Nhưng làm cách nào để tìm được hàm mũ của một ma trận? Sau đây sẽ là một số phương pháp được sử dụng để tìm hàm mũ ma trận.

Phương pháp sử dụng định nghĩa

Với phương pháp này, ta chỉ áp dụng với các ma trận khi lũy thừa có điểm dừng hoặc có thể đưa về dạng tổng quát.
Ta xét các ví dụ sau:

Ví dụ 1

Tìm hàm mũ của ma trận $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Cách giải
Ta tính lũy thừa liên tiếp của A:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$
$$A^2 = A^3 = ... = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$
$$\Rightarrow e^{At} = I_n + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

Ví dụ 2

Tìm hàm mũ của ma trận $A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$

Cách giải
Ta tính lũy thừa liên tiếp của A:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$
$$A^2 = \begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$$
$$\vdots$$
$$A^n = \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix}$$
$$\Rightarrow e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix}$$

Nhận xét:
Nếu tính lũy thừa của A như trong hai ví dụ trên, ta sẽ rất khó khi tính những hàm mũ không có quy luật rõ ràng. Vì vậy ta sẽ tìm hiểu thêm những phương pháp tiếp theo.

Phương pháp dùng đại số tuyến tính

Phương pháp này được áp dụng khi ma trận cần tìm hàm mũ là một ma trận chéo hóa được.
Cho A là một ma trận $n \times n$ với hệ số không đổi. Giả sử ma trận A chéo hóa được với các vectơ riêng độc lập $v_1, v_2, ..., v_n$ và các trị riêng tương ứng $\lambda_1, \lambda_2, ..., \lambda_n$.
Đặt:

$$S = \begin{bmatrix} v_1 & v_2 & .. & v_n \end{bmatrix}$$
$$D = \begin{bmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & .. & \lambda_n \end{bmatrix}$$
$$\Rightarrow S^{-1}.A.S = D$$

Khi đó:

$$e^{Dt} = \begin{bmatrix} e^{\lambda_1 t} & 0 & ... & 0 \\ 0 & e^{\lambda_2 t} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & .. & e^{\lambda_n t} \end{bmatrix}$$

Ta lại có:

$$e^{Dt} = \sum_{n=0}^{\infty} \frac{t^n (S^{-1}AS)^n}{n!} = \sum_{n=0}^{\infty} \frac{t^n S^{-1} A^n S}{n!} = S^{-1} \sum_{n=0}^{\infty} \frac{t^n A^n}{n!} S = S^{-1} e^{At} S$$

Suy ra:

$$e^{At} = S e^{Dt} S^{-1}$$

Hay

$$e^{At} = S \begin{bmatrix} e^{\lambda_1 t} & 0 & ... & 0 \\ 0 & e^{\lambda_2 t} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & .. & e^{\lambda_n t} \end{bmatrix} S^{-1} \tag{19}$$

Phương pháp Williamson

Phương pháp này được áp dụng khi ma trận cần tìm hàm mũ không chéo hóa được.

Cho A là một ma trận $n \times n$ với hệ số không đổi. Giả sử $\lambda_1, \lambda_2, ..., \lambda_n$ là danh sách các trị riêng, với nhiều giá trị lặp lại theo bội số của chúng.
Đặt:

$$\begin{cases} a_1 = e^{\lambda_1 t} \\ a_k = \int_0^t \mathrm{e}^{\lambda_k (t-u)} a_{k-1}(u) \mathrm{d}u \end{cases}$$

$$\begin{cases} B_1 = I \\ B_2 = (A - \lambda_{k-1} I) B_{k-1} \end{cases}$$

Với $k = 2, 3, ..., n$
Khi đó, ta có hàm mũ ma trận:

$$e^{At} = a_1 B_1 + a_2 B_2 + ... + a_n B_n \tag{20}$$

4.1.2 Các tính chất về nghiệm hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng

Định lí 1: Nếu $X^*(t)$ là nghiệm của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng, $X_1(t), X_2(t), ..., X_n(t)$ là hệ nghiệm cơ bản của hệ phương trình vi phân tuyến tính thuần nhất với hệ số hằng tương ứng thì nghiệm tổng quát của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng có dạng:

$$X = c_1 X_1(t) + c_2 X_2(t) + ... + C_n X_n(t) + X^*(t)$$

trong đó $c_1, c_2, ..., c_n$ là các hằng số bất kì.

Định lí 2: Nếu $X_1(t), X_2(t)$ là hai nghiệm tương ứng của các hệ phương trình

$$x' = Ax + F_1(x); \qquad x' = Ax + F_2(x)$$

thì $X(t) = X_1(t) + X_2(t)$ là nghiệm của hệ phương trình

$$x' = Ax + F_1(x) + F_2(x)$$

4.1.3 Nghiệm tổng quát của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng

Trở lại với hệ (16):

$$\begin{cases} x' = Ax + B(t) \\ x(0) = x_0 \end{cases}$$

Gọi x_c là nghiệm của hệ phương trình thuần nhất:

$$\begin{cases} x' = Ax \\ x(0) = x_0 \end{cases}$$

Gọi x_p là nghiệm riêng của hệ (16).
Ta được nghiệm tổng quát của hệ (16) có dạng như sau:

$$x = x_c + x_p$$

Ta tiếp tục tìm dạng tổng quát của x_c và x_p
Dựa và công thức (18) nghiệm x_c có dạng:

$$x_c = e^{At}x_0$$

Suy ra, nghiệm x_p có dạng:

$$x_p = e^{At}U(t)$$

Ta tiếp tục tìm dạng tổng quát của U(t).
Thay x_p vào hệ (16) ta được:

$$\begin{aligned} x_p' &= Ax_p + B(t) \\ \Rightarrow Ae^{At}U(t) + e^{At}U'(t) &= Ae^{At}U(t) + B(t) \\ \Rightarrow e^{At}U'(t) &= B(t) \\ U(t) &= \int_0^t \mathrm{e}^{-As}B(s)\mathrm{d}s \end{aligned}$$

Suy ra, nghiệm của hệ (16) có dạng tổng quát:

$$x = e^{At}x_0 + \int_0^t \mathrm{e}^{A(t-s)}B(s)\mathrm{d}s \tag{21}$$

Từ công thức nghiệm (21), ta suy ra hệ (15) có nghiệm như sau:

$$\begin{bmatrix} R \\ J \end{bmatrix} = e^{At}x_0 + \int_0^t \mathrm{e}^{A(t-s)} \begin{bmatrix} f(x) \\ g(x) \end{bmatrix} \mathrm{d}s \tag{22}$$

4.1.4 Ví dụ tìm nghiệm của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số không đổi

Ví dụ 1: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = -3R + 4J + sin(t) \\ J' = -2R + 3J + t \\ R(0) = 0, J(0) = 1 \end{cases}$$

Giải

Biểu diễn hệ phương trình trên lại dưới dạng:

$$\begin{cases} x' = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} sin(t) \\ t \end{bmatrix} \\ x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases}$$

Với $x = \begin{bmatrix} R \\ J \end{bmatrix}$

Xét ma trận $A = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix}$

Phương trình đặc trưng của A có 2 trị riêng thực phân biệt $\lambda_1 = 1; \lambda_2 = -1$, vectơ riêng tương ứng là $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; v_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. Suy ra hàm mũ của A:

$$e^{At} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} e^t & 0 \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} -e^t + 2e^{-t} & 2e^t - 2e^{-t} \\ -e^t + e^{-t} & 2e^t - e^{-t} \end{bmatrix}$$

Gọi x_c là nghiệm của hệ thuần nhất suy ra từ hệ ban đầu, x_p là nghiệm riêng của hệ ban đầu. Suy ra:

$$x_c = e^{At} x_0 = \begin{bmatrix} -e^t + 2e^{-t} & 2e^t - 2e^{-t} \\ -t & 2e^t - e^{-t} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2e^t - 2e^{-t} \\ 2e^t - e^{-t} \end{bmatrix}$$

$$x_p = \int_0^t \mathrm{e}^{A(t-s)} . \begin{bmatrix} sin(s) \\ s \end{bmatrix} \mathrm{d}s = \int_0^t \begin{bmatrix} -e^{t-s} + 2e^{-(t-s)} & 2e^{t-s} - 2e^{-(t-s)} \\ -e^{t-s} + e^{-(t-s)} & 2e^{t-s} - e^{-(t-s)} \end{bmatrix} \begin{bmatrix} sin(s) \\ s \end{bmatrix} \mathrm{d}s$$

$$= \begin{bmatrix} \int_0^t ((sin(s)(-e^{t-s} + 2e^{-t+s})) + s(2e^{t-s} - 2e^{-t+s}))\mathrm{d}s \\ \int_0^t ((sin(s)(-e^{t-s} + e^{-t+s})) + s(2e^{t-s} - e^{-t+s}))\mathrm{d}s \end{bmatrix}$$

$$= \begin{bmatrix} -4t + \frac{3}{2}e^t - e^{-t} - \frac{1}{2}cos(t) + \frac{3}{2}sin(t) \\ -1 - 3t + \frac{3}{2}e^t - \frac{1}{2}e^{-t} + sin(t) \end{bmatrix}$$

Suy ra nghiệm của hệ phương trình ban đầu:

$$x = x_c + x_p = \begin{bmatrix} \frac{7}{2}e^t - 3e^{-t} + \frac{3}{2}sin(t) - \frac{1}{2}cos(t) - 4t \\ \frac{7}{2}e^t - \frac{3}{2}e^{-t} + sin(t) - 3t - 1 \end{bmatrix}$$

Vậy hệ phương trình đã cho có nghiệm duy nhất:

$$R = \frac{7}{2}e^t - 3e^{-t} + \frac{3}{2}sin(t) - \frac{1}{2}cos(t) - 4t$$
$$J = \frac{7}{2}e^t - \frac{3}{2}e^{-t} + sin(t) - 3t - 1$$

Ví dụ 2: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = 3R - J + t \\ J' = R + J + t^2 \\ R(0) = 3, J(0) = 4 \end{cases}$$

Giải

Biểu diễn hệ phương trình trên lại dưới dạng:

$$\begin{cases} x' = \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix} x + \begin{bmatrix} t \\ t^2 \end{bmatrix} \\ x(0) = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \end{cases}$$

Với $x = \begin{bmatrix} R \\ J \end{bmatrix}$

Xét ma trận $A = \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix}$. Phương trình đặc trưng $det(A - \lambda I) = 0$ có nghiệm bội $\lambda_1 = \lambda_2 = 2$.

Để tìm hàm mũ của ma trận A. Ta sẽ tính dãy a_k và B_k (với k=1,2):

Dãy a_k:

$$\begin{cases} a_1 = e^{2t} \\ a_2 = \int_0^t \mathrm{e}^{2(t-u)} * \mathrm{e}^{2u} \mathrm{d}u = e^{2t} \int_0^t \mathrm{d}u = t.e^{2t} \end{cases}$$

Dãy B_k:

$$\begin{cases} B_1 = I \\ B_2 = (A - 2I)I = A - 2I = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \end{cases}$$

Suy ra hàm mũ của A:

$$e^{At} = a_1.B_1 + a_2.B_2 = e^{2t} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + t.e^{2t}. \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} e^{2t} + t.e^{2t} & -te^{2t} \\ te^{2t} & e^{2t} - te^{2t} \end{bmatrix}$$

Suy ra nghiệm của hệ phương trình thuần nhất:

$$x_c = e^{At}.x_0 = \begin{bmatrix} -t.e^{2t} + 3e^{2t} \\ -te^{2t} + 4e^{2t} \end{bmatrix}$$

Nghiệm riêng của hệ phương trình đã cho:

$$x_p = \int_0^t \mathrm{e}^{A(t-s)}.\begin{bmatrix} s \\ s^2 \end{bmatrix} \mathrm{d}s = \int_0^t \begin{bmatrix} e^{2(t-s)} + (t-s)e^{2(t-s)} & -(t-s)e^{2(t-s)} \\ (t-s)e^{2(t-s)} & e^{2(t-s)} - (t-s)e^{2(t-s)} \end{bmatrix} \begin{bmatrix} s \\ s^2 \end{bmatrix} \mathrm{d}s$$

$$= \begin{bmatrix} \frac{1}{8}(-2t^2 - 6t + 3e^{2t} - 3) \\ \\ \frac{3}{8}(-2t^2 - 2t + e^{2t} - 1) \end{bmatrix}$$

Vậy nghiệm của phương trình đã cho là:

$$x = x_c + x_p \begin{bmatrix} -t.e^{2t} + 3e^{2t} + \frac{1}{8}(-2t^2 - 6t + 3e^{2t} - 3) \\ \\ -te^{2t} + 4e^{2t} + \frac{3}{8}(-2t^2 - 2t + e^{2t} - 1) \end{bmatrix}$$

Hay:

$$R = -t.e^{2t} + 3e^{2t} + \frac{1}{8}(-2t^2 - 6t + 3e^{2t} - 3)$$

$$J = -te^{2t} + 4e^{2t} + \frac{3}{8}(-2t^2 - 2t + e^{2t} - 1)$$

Ví dụ 3: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = J + t - 1 \\ J' = -R + t^2 \\ R(0) = 1, J(0) = 2 \end{cases}$$

Giải

Biểu diễn hệ phương trình trên lại dưới dạng:

$$\begin{cases} x' = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} t-1 \\ t^2 \end{bmatrix} \\ x(0) = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \end{cases}$$

Với $x = \begin{bmatrix} R \\ J \end{bmatrix}$

Xét ma trận $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Phương trình đặc trưng $det(A - \lambda I) = 0$ có nghiệm phức với phần thực bằng 0 $\lambda_1 = i$; $\lambda_2 = -i$.
Để tìm hàm mũ của ma trận A. Ta sẽ tính dãy a_k và B_k (với k=1,2):

Dãy a_k:

$$\begin{cases} a_1 = e^{it} \\ a_2 = \int\limits_0^t \mathrm{e}^{-i(t-u)}.\mathrm{e}^{iu}\mathrm{d}u = e^{-it}\int\limits_0^t \mathrm{e}^{2iu}\mathrm{d}u = \frac{i}{2}(e^{-it} - e^{it}) \end{cases}$$

Áp dụng công thức euler: $e^{ix} = cos(x) + i.sin(x)$. Ta viết lại dãy a_k:

$$\begin{cases} a_1 = cos(t) + i.sin(t) \\ a_2 = \frac{i}{2}(cos(t) - i.sin(t) - cos(t) - i.sin(t)) = sin(t) \end{cases}$$

Dãy B_k:

$$\begin{cases} B_1 = I \\ B_2 = (A - iI)I = A - iI = \begin{bmatrix} -i & 1 \\ -1 & -i \end{bmatrix} \end{cases}$$

Suy ra hàm mũ của A:

$$e^{At} = a_1.B_1 + a_2.B_2 = \begin{bmatrix} cos(t) & sin(t) \\ -sin(t) & cos(t) \end{bmatrix}$$

Suy ra nghiệm của hệ phương trình thuần nhất:

$$x_c = e^{At}.x_0 = \begin{bmatrix} 2sin(t) - cos(t) \\ sin(t) + 2cos(t) \end{bmatrix}$$

Nghiệm riêng của hệ phương trình đã cho:

$$x_p = \int\limits_0^t \mathrm{e}^{A(t-s)}.\begin{bmatrix} s-1 \\ s^2 \end{bmatrix}\mathrm{d}s = \int\limits_0^t \begin{bmatrix} cos(t-s) & sin(t-s) \\ -sin(t-s) & cos(t-s) \end{bmatrix}\begin{bmatrix} s-1 \\ s^2 \end{bmatrix}\mathrm{d}s$$

$$= \begin{bmatrix} t^2 - sin(t) + cos(t) - 1 \\ t - sin(t) - cos(t) + 1 \end{bmatrix}$$

Suy ra nghiệm duy nhất của hệ phương trình đã cho:

$$x = x_p + x_c = \begin{bmatrix} sin(t) + t^2 - 1 \\ cos(t) + t + 1 \end{bmatrix}$$

Hay:

$$R = sin(t) + t^2 - 1$$

$$J = cos(t) + t + 1$$

Ví dụ 4: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = 2R - 5J + e^{2t} \\ J' = 2R - 4J + e^t \\ R(0) = 1, J(0) = 0 \end{cases}$$

Giải

Biểu diễn hệ phương trình trên lại dưới dạng:

$$\begin{cases} x' = \begin{bmatrix} 2 & -5 \\ 2 & -4 \end{bmatrix} x + \begin{bmatrix} e^{2t} \\ e^{t} \end{bmatrix} \\ x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{cases}$$

Với $x = \begin{bmatrix} R \\ J \end{bmatrix}$

Xét ma trận $A = \begin{bmatrix} 2 & -5 \\ 2 & -4 \end{bmatrix}$. Phương trình đặc trưng $det(A - \lambda I) = 0$ có nghiệm phức với phần thực khác 0 $\lambda_1 = -1 + i$; $\lambda_2 = -1 - i$.
Để tìm hàm mũ của ma trận A. Ta sẽ tính dãy a_k và B_k (với k=1,2):
Dãy a_k:

$$\begin{cases} a_1 = e^{(-1-i)t} \\ a_2 = \int\limits_0^t \mathrm{e}^{(-1+i)(t-u)}.\mathrm{e}^{(-1-i)u}\mathrm{d}u = e^{(-1+i)t}\int\limits_0^t \mathrm{e}^{-2iu}\mathrm{d}u = \frac{i}{2}(e^{(-1-i)t} - e^{(-1+i)t}) \end{cases}$$

Áp dụng công thức euler: $e^{ix} = cos(x) + i.sin(x)$. Ta viết lại dãy a_k:

$$\begin{cases} a_1 = e^{-t}(cos(t) - i.sin(t)) \\ a_2 = e^{-t}sin(t) \end{cases}$$

Dãy B_k:

$$\begin{cases} B_1 = I \\ B_2 = (A - iI)I = A - (-1 - i)I = \begin{bmatrix} 3+i & -5 \\ 2 & -3+i \end{bmatrix} \end{cases}$$

Suy ra hàm mũ của A:

$$e^{At} = a_1.B_1 + a_2.B_2 = e^{-t}\begin{bmatrix} cos(t) + 3sin(t) & -5sin(t) \\ 2sin(t) & cos(t) - 3sin(t) \end{bmatrix}$$

Suy ra nghiệm của hệ phương trình thuần nhất:

$$x_c = e^{At}.x_0 = e^{-t}\begin{bmatrix} cos(t) + 3sin(t) \\ 2sin(t) \end{bmatrix}$$

Nghiệm riêng của hệ phương trình đã cho:

$$x_p = \int\limits_0^t \mathrm{e}^{A(t-s)}.\begin{bmatrix} e^{2t} \\ e^{t} \end{bmatrix}\mathrm{d}s = \int\limits_0^t e^{-t}\begin{bmatrix} cos(t-s) + 3sin(t-s) & -5sin(t-s) \\ 2sin(t-s) & cos(t-s) - 3sin(t-s) \end{bmatrix}\begin{bmatrix} e^{2t} \\ e^{t} \end{bmatrix}\mathrm{d}s$$

$$= \begin{bmatrix} \frac{1}{5}(e^t(3e^t - 5) + 6e^{-t}sin(t) + 2e^{-t}cos(t)) \\ \\ \frac{1}{5}(e^t(e^t - 1) + 4e^{-t}sin(t)) \end{bmatrix}$$

Suy ra nghiệm duy nhất của hệ phương trình đã cho:

$$x = x_p + x_c = \begin{bmatrix} e^{-t}cos(t) + 3e^{-t}sin(t) + \frac{1}{5}(e^t(3e^t - 5) + 6e^{-t}sin(t) + 2e^{-t}cos(t)) \\ 2e^{-t}sin(t) + \frac{1}{5}(e^t(e^t - 1) + 4e^{-t}sin(t)) \end{bmatrix}$$

Hay:

$$R = e^{-t}cos(t) + 3e^{-t}sin(t) + \frac{1}{5}(e^t(3e^t - 5) + 6e^{-t}sin(t) + 2e^{-t}cos(t))$$

$$J = 2e^{-t}sin(t) + \frac{1}{5}(e^t(e^t - 1) + 4e^{-t}sin(t))$$

Ví dụ 5: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = -3R + 4J + e^{2t} + t \\ J' = -2R + 3J - e^{2t} \\ R(0) = -1, J(0) = -2 \end{cases}$$

Giải

Biểu diễn hệ phương trình trên lại dưới dạng:

$$\begin{cases} x' = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} e^{2t} \\ -e^{2t} \end{bmatrix} + \begin{bmatrix} t \\ 0 \end{bmatrix} \\ x(0) = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \end{cases}$$

Với $x = \begin{bmatrix} R \\ J \end{bmatrix}$

Xét ma trận $A = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix}$

Làm tương tự ví dụ 1. Suy ra hàm mũ của A:

$$e^{At} = \begin{bmatrix} -e^t + 2e^{-t} & 2e^t - 2e^{-t} \\ -e^t + e^{-t} & 2e^t - e^{-t} \end{bmatrix}$$

Gọi x_c là nghiệm của hệ thuần nhất suy ra từ hệ ban đầu.

x_{p1} là nghiệm riêng của hệ

$$\begin{cases} x' = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} e^{2t} \\ -e^{2t} \end{bmatrix} \\ x(0) = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \end{cases}$$

x_{p2} là nghiệm riêng của hệ

$$\begin{cases} x' = \begin{bmatrix} -3 & 4 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} t \\ 0 \end{bmatrix} \\ x(0) = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \end{cases}$$

Suy ra:

$$x_c = e^{At}x_0 = \begin{bmatrix} -e^t + 2e^{-t} & 2e^t - 2e^{-t} \\ -e^t + e^{-t} & 2e^t - e^{-t} \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} -3e^t + 2e^{-t} \\ -3e^t + e^{-t} \end{bmatrix}$$

$$x_{p1} = \int_0^t \mathrm{e}^{A(t-s)} . \begin{bmatrix} e^{2s} \\ -e^{2s} \end{bmatrix} \mathrm{d}s = \int_0^t \begin{bmatrix} -e^{t-s} + 2e^{-(t-s)} & 2e^{t-s} - 2e^{-(t-s)} \\ -e^{t-s} + e^{-(t-s)} & 2e^{t-s} - e^{-(t-s)} \end{bmatrix} \begin{bmatrix} e^{2s} \\ -e^{2s} \end{bmatrix} \mathrm{d}s$$

$$= \begin{bmatrix} -\frac{4}{3}e^{-t} + 3e^t - \frac{5}{3}e^{2t} \\ -\frac{2}{3}e^{-t} + 3e^t - \frac{7}{3}e^{2t} \end{bmatrix}$$

$$x_{p2} = \int_0^t \mathrm{e}^{A(t-s)} . \begin{bmatrix} s \\ 0 \end{bmatrix} \mathrm{d}s = \int_0^t \begin{bmatrix} -e^{t-s} + 2e^{-(t-s)} & 2e^{t-s} - 2e^{-(t-s)} \\ -e^{t-s} + e^{-(t-s)} & 2e^{t-s} - e^{-(t-s)} \end{bmatrix} \begin{bmatrix} s \\ 0 \end{bmatrix} \mathrm{d}s$$

$$= \begin{bmatrix} 3t + 2e^{-t} - e^t - 1 \\ 2t + e^{-t} - e^t \end{bmatrix}$$

Theo tính chất 2 về nghiệm hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng. Suy ra nghiệm của hệ phương trình ban đầu:

$$x = x_c + x_{p1} + x_{p2} = \begin{bmatrix} -\frac{5}{3}e^{2t} - e^t + \frac{8}{3}e^{-t} + 3t - 1 \\ -\frac{7}{3}e^{2t} - e^t + \frac{4}{3}e^{-t} + 2t \end{bmatrix}$$

Vậy hệ phương trình đã cho có nghiệm duy nhất:

$$R = -\frac{5}{3}e^{2t} - e^t + \frac{8}{3}e^{-t} + 3t - 1$$
$$J = -\frac{7}{3}e^{2t} - e^t + \frac{4}{3}e^{-t} + 2t$$

4.1.5 Điều kiện tồn tại nghiệm của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng

Hàm số sơ cấp: là hàm gồm hằng số, tổ hợp của một số hữu hạn các phép toán số học ($+ - \times \div$) và các hàm theo biến x gồm:

1. Lũy thừa của x: x, x^2, x^3,...
2. Căn của x: $\sqrt{x}$, $\sqrt[3]{x}$,...
3. Hàm mũ: e^x
4. Logarit: $log(x)$
5. Hàm lượng giác: $sin(x), cos(x), ...$

6. Hàm lượng giác ngược: $arcsin(x), arccos(x), ...$
7. Hàm Hyperbolic: $sinh(x), cosh(x), ...$
8. Tất cả các hàm số được tạo thành bằng cách thay x (trong một hàm số sơ cấp) bởi bất kỳ một hàm số sơ cấp nào khác: $log(sin(x)), ...$
9. Tất cả các hàm số được tạo thành bằng cách cộng, trừ, nhân hay chia các hàm số sơ cấp trước đó: $2x - sin(x), ...$

Tính chất của hàm số sơ cấp:

1. Mọi hàm số sơ cấp đều liên tục trên tập xác định của nó.
2. Mọi hàm số sơ cấp đều tồn tại nguyên hàm trên tập xác định của nó.

Điều kiện tồn tại nghiệm của hệ phương trình vi phân tuyến tính không thuần nhất với hệ số hằng:

Trở lại với hệ (15):

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

Hệ phương trình vi phân này tồn tại nghiệm khi và chỉ khi:

$f(t)$, $g(t)$ **liên tục** và **tồn tại nguyên hàm** trên khoảng xác định.

Hệ phương trình vi phân này tồn tại nghiệm là hàm sơ cấp khi và chỉ khi:

1. $f(t)$, $g(t)$ là **hàm số sơ cấp**.
2. $f(t)$, $g(t)$ có **nguyên hàm** là **hàm số sơ cấp**.

Ví dụ về các hệ phương trình vi phân tuyến tính không thuần nhất không thể tìm nghiệm chính xác:

Ví dụ 1: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = -R + J + \frac{sin(t+1)}{t+1} \\ J' = -R + J + t \\ R(0) = R0, J(0) = J0 \end{cases}$$

Giải

Ở phương trình thứ nhất, ta thấy $\frac{sin(t+1)}{t+1}$ có nguyên hàm không là hàm sơ cấp. Lấy nguyên hàm 2 vế theo t ta được:

$$R = \int -R\mathbf{d}t + \int J\mathbf{d}t + \int \frac{sin(t+1)}{t+1}\mathbf{d}t$$

Nhận xét: Ta không thể tìm được $\int \frac{sin(t+1)}{t+1}\mathbf{d}t$ theo hàm số sơ cấp. Suy ra R là hàm không sơ cấp.
Vậy suy ra hệ đã cho tồn tại nghiệm không phải là hàm sơ cấp..

Ví dụ 2: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = 2R + J + t \\ J' = R + J + e^{\frac{1}{t}} \\ R(0) = R0, J(0) = J0 \end{cases}$$

Giải
Ở phương trình thứ hai, ta thấy $e^{\frac{1}{t}}$ có nguyên hàm không là hàm sơ cấp. Làm như ví dụ 1, suy ra hệ đã cho tồn tại nghiệm không phải là hàm sơ cấp.

Ví dụ 3: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = R + J + e^{-t^2} \\ J' = R + J + \frac{e^t}{t} \\ R(0) = R0, J(0) = J0 \end{cases}$$

Giải
Ở hệ trên, ta thấy e^{-t^2} và $\frac{e^t}{t}$ đều có nguyên hàm không là hàm sơ cấp. Làm như ví dụ 1, suy ra hệ đã cho tồn tại nghiệm không phải là hàm sơ cấp.

Ví dụ 4: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = R + J + it \\ J' = R + J + t^2 \\ R(0) = R0, J(0) = J0 \end{cases}$$

Giải
Ở hệ trên, ta thấy it không là hàm sơ cấp và tồn tại nguyên hàm.
Vậy hệ đã cho tồn tại nghiệm không phải là hàm sơ cấp.

Ví dụ 5: Tìm nghiệm của hệ phương trình sau:

$$\begin{cases} R' = R + J + it^2 \\ J' = R + J + erf(t) \\ R(0) = R0, J(0) = J0 \end{cases}$$

Giải
Ở hệ trên, ta thấy it^2 và $erf(t)$ không là hàm sơ cấp và tồn tại nguyên hàm.
Vậy hệ đã cho tồn tại nghiệm không phải là hàm sơ cấp.

4.2 Hệ phương trình vi phân thường cấp 1 tổng quát

Xét một hệ phương trình vi phân phức tạp hơn của tình yêu giữa Romeo và Juliet:

$$\begin{cases} R' = f(t, R, J) \\ J' = g(t, R, J) \\ R(0) = R_0, J(0) = J_0 \end{cases} \tag{23}$$

Trong đó f, g là hai hàm phụ thuộc vào t, R, J.
Hệ (23) có dạng hệ phương trình vi phân thường cấp 1 tổng quát.

4.2.1 Điều kiện Lipschitz

Định nghĩa:

Hàm F(x,y) được gọi là thỏa mãn điều kiện **Lipschitz** trên miền D trên mặt phẳng x,y khi tồn tại một số dương K thỏa:

$$|F(x, y_1) - F(x, y_2)| < K|y_1 - y_2|$$

Với (x, y_1) và (x, y_2) là 2 điểm bất kì thuộc miền D. K được gọi là **hằng số Lipschitz**.

Ví dụ về điều kiện Lipschitz:

Ví dụ 1: Hàm $F(x, y) = 4x^2 + y^2$ là thỏa điều kiện Lipschitz trên miền $D : |x| \leq 1, |y| \leq 1$

Chứng minh

Xét $|F(x, y_1) - F(x, y_2)| = |y_1^2 - y_2^2| = |(y_1 + y_2)(y_1 - y_2)| \leq 2(y_1 - y_2)$
Vậy hàm $F(x, y) = 4x^2 + y^2$ là thỏa điều kiện Lipschitz trên miền $D : |x| \leq 1, |y| \leq 1$.

Ví dụ 2: Hàm $F(x, y) = 4x^2 + y^2$ là thỏa điều kiện Lipschitz trên miền $D : |x| \leq 1, y \in R$

Chứng minh

Xét $|F(x, y_1) - F(x, y_2)| = |y_1^2 - y_2^2| = |(y_1 + y_2)(y_1 - y_2)|$
Nhận xét: Nếu y_1, y_2 tiến với ∞ thì $|F(x, y_1) - F(x, y_2)|$. Vậy hàm $F(x, y) = 4x^2 + y^2$ không là hàm Lipschitz trên miền $D : |x| \leq 1, y \in R$.

4.2.2 Điều kiện tồn tại và duy nhất nghiệm

Hệ phương trình vi phân (23) tồn tại và duy nhất nghiệm khi và chỉ khi thỏa mãn các điều kiện sau:

1. Các hàm f, g liên tục trong miền G (với G là miền xác định của hệ phương trình vi phân).
2. Các hàm f, g thỏa mãn điều kiện **Lipschitz** theo R, J trong miền G với cùng hằng số **Lipschitz** $L > 0$.

Khi đó tồn tại nghiệm duy nhất

$$J(t), R(t)$$

thỏa mãn điều kiện ban đầu

$$J(t_0) = J0, R(t_0) = R0$$

4.2.3 Ví dụ về hệ phương trình vi phân thường cấp 1 tổng quát không thể tìm nghiệm chính xác

Ví dụ 1: Tìm nghiệm của hệ phương trình:

$$\begin{cases} R' = \sqrt{1-R^2}J \\ J' = \sqrt{1-J^2}R \\ R(0) = 0.2, J(0) = 0.3 \end{cases}$$

Giải

Miền G trong hệ trên:

$$G = \{t \geq 0; -1 \leq R \leq 1; -1 \leq J \leq 1\}$$

Xét $|F1(R, J_1) - F1(R, J_2)| = |\sqrt{1-R^2}(J_1 - J_2)| \leq 1|J_1 - J_2|$

Do đó $\sqrt{1-R^2}J$ thỏa điều kiện Lipschitz theo R, J trên miền G. Chứng minh tương tự, $\sqrt{1-J^2}R$ cũng thỏa điều kiện Lipschitz theo R, J trên miền G. Suy ra hệ đã cho tồn tại nghiệm.

Ví dụ 2: Tìm nghiệm của hệ phương trình:

$$\begin{cases} R' = R(1-J) \\ J' = R(1-J) \\ R(0) = 2, J(0) = 3 \end{cases}$$

Giải

Nhận xét: Miền G trong hệ trên:

$$G = \{t \geq 0; a \leq R \leq b; c \leq J \leq d\}$$

Với a,b,c,d là hằng số.
Xét $|F1(R, J_1) - F1(R, J_2)| = |-RJ_1 + RJ_2| = |R(J_1 - J_2)|$

$$|F2(R, J_1) - F2(R, J_2)| = |(R-1)J_1 - (R-1)J_2| = |(R-1).(J_1 - J_2)|$$

Lại có R bị chặn trên miền xác định. Do đó $R(1-J)$ và $R(1-J)$ thỏa điều kiện Lipschitz theo R, J trên miền G. Suy ra hệ đã cho tồn tại nghiệm.

Ví dụ 3: Tìm nghiệm của hệ phương trình:

$$\begin{cases} R' = \frac{1}{\sqrt{(R-2)J}} \\ J' = \frac{1}{\sqrt{(J-2)R}} \\ R(0) = 3, J(0) = 4 \end{cases}$$

Giải

Nhận xét: $R' > 0$ và $J' > 0$. Suy ra miền G trong hệ trên:

$$G = \{t \geq 0; R \geq 3; J \geq 4\}$$

Xét $|F1(R, J_1) - F1(R, J_2)| = |\frac{1}{\sqrt{(R-2)J_1}} - \frac{1}{\sqrt{(R-2)J_2}}| = |\frac{1}{\sqrt{(R-2)J_1 J_2}\sqrt{J_1+J_2}}(J_1 - J_2)| < 1|J_1 - J_2|$

Do đó $\frac{1}{\sqrt{(R-1)J}}$ thỏa điều kiện Lipschitz theo R, J trên miền G. Chứng minh tương tự, $\frac{1}{\sqrt{(J-1)R}}$ cũng thỏa điều kiện Lipschitz theo R, J trên miền G. Suy ra hệ đã cho tồn tại nghiệm.

Ví dụ 4: Tìm nghiệm của hệ phương trình:

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

Giải

Nhận xét: $R' > 0$ và $J' > 0$. Suy ra miền G trong hệ trên:

$$G = \{t \geq 0; R \geq 1; J \geq 2\}$$

Xét $|F1(R, J_1) - F(R, J_2)| = |\frac{1}{RJ_1J_2}(J_1 - J_2)| < 1|J_1 - J_2|$

Do đó $\sqrt{R-1} + \frac{1}{RJ}$ thỏa điều kiện Lipschitz theo R, J trên miền G. Chứng minh tương tự, $\sqrt{J-1} + \frac{1}{RJ}$ cũng thỏa điều kiện Lipschitz theo R, J trên miền G. Suy ra hệ đã cho tồn tại nghiệm.

Ví dụ 5: Tìm nghiệm của hệ phương trình:

$$\begin{cases} R' = R^2J^2 \\ J' = \sqrt{16 - R^2} + \sqrt{81 - J^2} \\ R(0) = 3, J(0) = 5 \end{cases}$$

Giải

Nhận xét: $R' \geq 0$ và $J' \geq 0$. Suy ra miền G trong hệ trên:

$$G = \{t \geq 0; 3 \leq R \leq 4; 5 \leq J \leq 9\}$$

Xét $|F1(R, J_1) - F1(R, J_2)| = |R^2(J_1 + J_2)(J_1 - J_2)| \leq 228|J_1 - J_2|$

$$|F2(R, J_1) - F2(R, J_2)| = |\sqrt{81 - J_1^2} - \sqrt{81 - J_2^2}| = |\frac{J_1 + J_2}{\sqrt{81-J_1^2} + \sqrt{81-J_2^2}}(J_1 - J_2)| < 2|J_1 - J_2|$$

Do đó R^2J^2 và $\sqrt{16 - R^2} + \sqrt{81 - J^2}$ thỏa điều kiện Lipschitz theo R, J trên miền G. Suy ra hệ đã cho tồn tại nghiệm.

5 Bài tập 4

5.1 Giải xấp xỉ hệ phương trình vi phân bằng phương pháp Euler tường minh

5.1.1 Phương pháp Euler tường minh

Phương pháp Euler tường minh (Explicit Euler method) là phương pháp đơn giản nhất để giải xấp xỉ hệ phương trình vi phân tổng quát (23).

Với giá trị R_0, J_0 là giá trị của hàm $R(t), J(t)$ tại thời điểm ban đầu t_0, giá trị xấp xỉ của R_{k+1}, J_{k+1} tại thời điểm $t_{k+1} = t_0 + h.(k+1)$, với h là bước nhảy thời gian, được tính theo công thức:

$$\begin{cases} R_{k+1} = R_k + f(t_k, R_k, J_k) * h \\ J_{k+1} = J_k + g(t_k, R_k, J_k) * h \end{cases} \tag{24}$$

5.1.2 Nguồn gốc

Có nhiều cách để suy ra công thức của phương pháp Euler tường minh.

Cách thứ nhất là sử dụng khai triển Taylor của hàm R và hàm J tại thời điểm t_k:

$$\begin{cases} R(t_{k+1}) = R(t_k + h) = R(t_k) + h.R'(t_k) + \frac{1}{2}h^2R"(t_k) + O(h^3) \\ J(t_{k+1}) = J(t_k + h) = J(t_k) + h.J'(t_k) + \frac{1}{2}h^2J"(t_k) + O(h^3) \end{cases} \tag{25}$$

$$\Rightarrow \begin{cases} R(t_{k+1}) = R(t_k) + h.f(t_k, R_k, J_k) + \frac{1}{2}h^2f'(t_k, R_k, J_k) + O(h^3) \\ J(t_{k+1}) = J(t_k) + h.g(t_k, R_k, J_k) + \frac{1}{2}h^2g'(t_k, R_k, J_k) + O(h^3) \end{cases} \tag{26}$$

Nếu bỏ qua các số hạng bậc hai và bậc cao hơn, ta sẽ có được hệ phương trình (24), chính là phương pháp Euler tường minh.

Một cách khác chính là biến đổi thông qua công thức sai phân, với $t_{k+1} = t_k + h$, ta cũng thu được phương pháp Euler:

$$\begin{cases} R(t_{k+1}) \approx R(t_k) + (t_{k+1} - t_k)R'(t_k) \\ J(t_{k+1}) \approx J(t_k) + (t_{k+1} - t_k)J'(t_k) \end{cases} \tag{27}$$

$$\Rightarrow \begin{cases} R(t_{k+1}) \approx R(t_k) + hf(t_k, R_k, J_k) \\ J(t_{k+1}) \approx J(t_k) + hg(t_k, R_k, J_k) \end{cases} \tag{28}$$

Ngoài ra, ta cũng có thể tích phân phương trình vi phân từ t_k tới $t_k + h$ và áp dụng các định lí cơ bản của tích phân và vi phân để có được:

$$R(t_k + h) - R(t_k) = \int_{t_k}^{t_k+h} f(t, R(t), J(t)\ \mathrm{d}t$$

Tính xấp xỉ tích phân trên bằng tổng Riemann trái, ta được:

$$\int_{t_k}^{t_k+h} f(t, R(t), J(t)\ \mathrm{d}t \approx hf(t_k, R(t_k), J(t_k)$$

Kết hợp cả 2 phương trình, ta suy ra phương pháp Euler. Chứng minh tương tự với hàm $J(t)$.

5.1.3 Sai số cắt ngắn cục bộ (Local Truncation Error - LTE)

Theo phương pháp Euler tường minh đã được đề cập ở phần trên, giá trị xấp xỉ R_1, J_1 tại thời điểm $t_1 = t_0 + h$, được tính bằng:

$$\begin{cases} R_1 = R_0 + f(t_0, R_0, J_0) * h \\ J_1 = J_0 + g(t_0, R_0, J_0) * h \end{cases} \tag{29}$$

"Sai số cắt ngắn cục bộ" của phương pháp Euler là sai số trong 1 bước duy nhất. Nó là khoảng cách giữa kết quả xấp xỉ sau 1 bước R_1, J_1 và kết quả chính xác $R(t_1), J(t_1)$ tại thời điểm $t_1 = t_0 + h$. Đối với hệ phương trình vi phân, "sai số cắt ngắn cục bộ" được tính bởi công thức:

$$E(t_1) = \sqrt{[R(t_1) - R_1]^2 + [J(t_1) - J_1]^2}$$

Với $t_1 = t_0 + h$, áp dụng khai triển Taylor đối với $R(t_1)$ và $J(t_1)$, ta có:

$$\begin{cases} R(t_1) = R(t_0 + h) = R(t_0) + h.R'(t_0) + \frac{1}{2}h^2R"(t_0) + O(h^3) \\ J(t_1) = J(t_0 + h) = J(t_0) + h.J'(t_0) + \frac{1}{2}h^2J"(t_0) + O(h^3) \end{cases} \tag{30}$$

$$\Rightarrow \begin{cases} R(t_1) = R(t_0) + h.f(t_0, R_0, J_0) + \frac{1}{2}h^2f'(t_0, R_0, J_0) + O(h^3) \\ J(t_1) = J(t_0) + h.g(t_0, R_0, J_0) + \frac{1}{2}h^2g'(t_0, R_0, J_0) + O(h^3) \end{cases} \tag{31}$$

$$\Rightarrow \begin{cases} R(t_1) = R_1 + \frac{1}{2}h^2f'(t_0, R_0, J_0) + O(h^3) \\ J(t_1) = J_1 + \frac{1}{2}h^2g'(t_0, R_0, J_0) + O(h^3) \end{cases} \tag{32}$$

$$\Rightarrow \begin{cases} R(t_1) - R_1 = \frac{1}{2}h^2f'(t_0, R_0, J_0) + O(h^3) \\ J(t_1) - J_1 = \frac{1}{2}h^2g'(t_0, R_0, J_0) + O(h^3) \end{cases} \tag{33}$$

Đối với h nhỏ, ta có $lim(O(h^3)) = 0$. Do đó ta có thể tính xấp xỉ như sau:

$$\begin{cases} R(t_1) - R_1 \approx \frac{1}{2}h^2f'(t_0, R_0, J_0) \\ J(t_1) - J_1 \approx \frac{1}{2}h^2g'(t_0, R_0, J_0) \end{cases} \tag{34}$$

$$\begin{aligned} \Rightarrow E(t_1) &\approx \sqrt{[\tfrac{1}{2}h^2f'(t_0, R_0, J_0)]^2 + [\tfrac{1}{2}h^2g'(t_0, R_0, J_0)]^2} \\ &= \tfrac{1}{2}h^2\sqrt{f'^2(t_0, R_0, J_0) + g'^2(t_0, R_0, J_0)} \end{aligned}$$

Như vậy, trong trường hợp h nhỏ, ta thấy $E(t_1)$ xấp xỉ tỉ lệ thuận với h^2.

5.1.4 Độ ổn định số của phương pháp Euler tường minh

Đối với trường hợp bước nhảy $h < 1$, phương pháp Euler tường minh có tính ổn định về mặt phương pháp số, tức là khi $t \to \infty$, kết quả chính xác và kết quả xấp xỉ theo phương pháp Euler tường minh luôn hội tụ. Tuy nhiên, trong trường hợp $h \geq 1$, phương pháp Euler tường minh không ổn định về mặt phương pháp số, có nghĩa là lời giải số tăng rất nhanh, trong khi lời giải chính xác không tăng khi $t \to \infty$. Đây là một trong những lí do khiến phương pháp này không được sử dụng nhiều trong giải tích số.

5.2 Giải xấp xỉ hệ phương trình vi phân bằng phương pháp Euler ẩn

5.2.1 Phương pháp Euler ẩn

Phương pháp Euler ẩn (Implicit Euler method), hay còn gọi là phương pháp Euler lùi (Backward Euler method) là một biến thể của phương pháp Euler tường minh. Nó cũng là một trong những phương pháp đơn giản nhất để giải xấp xỉ một hệ phương trình vi phân tổng quát, tuy nhiên phương pháp này có thể khắc phục được nhược điểm về sự ổn định số của phương pháp Euler tường minh ban đầu.

Với giá trị R_0, J_0 là giá trị của hàm $R(t), J(t)$ tại thời điểm ban đầu t_0, giá trị xấp xỉ của R_{k+1}, J_{k+1} tại thời điểm $t_{k+1} = t_0 + h.(k+1)$, với h là bước nhảy thời gian, được tính theo công thức:

$$\begin{cases} R_{k+1} = R_k + f(t_{k+1}, R_{k+1}, J_{k+1}) * h \\ J_{k+1} = J_k + g(t_{k+1}, R_{k+1}, J_{k+1}) * h \end{cases} \tag{35}$$

Ta thấy rằng công thức của phương pháp Euler ẩn này khá giống với phương pháp Euler tường minh, tuy nhiên điểm khác nhau giữa chúng là ở cả 2 vế của 2 phương trình trong công thức của phương pháp Euler ẩn đều xuất hiện giá trị R_{k+1} và J_{k+1}. Do đó phương pháp này được gọi là phương pháp ẩn, và khi áp dụng phương pháp Euler ẩn chúng ta phải giải 1 hệ phương trình. Điều này làm cho việc thực hiện tốn kém thời gian hơn so với phương pháp Euler tường minh.

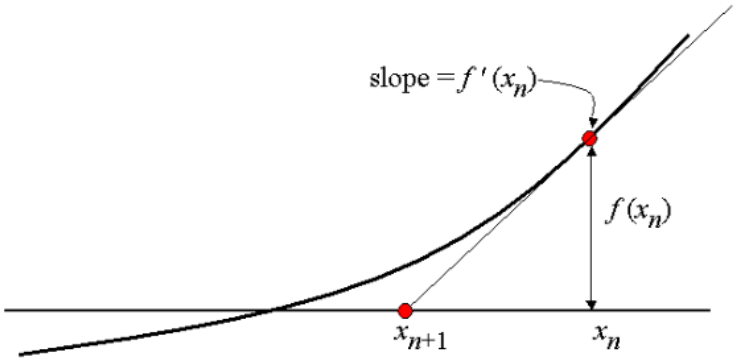
Để giải hệ phương trình và tìm ra nghiệm xấp xỉ như trên, người ta thường dùng phương pháp Newton - Raphson. Trong giải tích số, phương pháp Newton - Raphson là một giải thuật tìm nghiệm, tạo ra các giá trị gần đúng liên tiếp cho các nghiệm của phương trình $f(x) = 0$, với $f(x) \in R, \forall x \in R$. Ở dạng cơ bản nhất, với một hàm số f được định nghĩa theo biến thực x, có đạo hàm $f'(x)$, và ta đã biết nghiệm đang tìm nằm gần điểm $x = x_0$, phương pháp Newton - Raphson sẽ cho ta một nghiệm xấp xỉ khác gần nghiệm chính xác hơn so với x_0, đó là:

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

Quá trình này sẽ được lặp đi lặp lại nhiều lần, cho đến khi đạt được độ chính xác mong muốn. Tóm lại, với bất kì giá trị x_n, ta có:

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

Biểu diễn hình học của phương pháp Newton - Raphson như sau:



Hình 24: *Biểu diễn hình học phương pháp Newton-Raphson*

Ta vẽ một đường tiếp tuyến với đồ thị $f(x)$ tại điểm $x = x_n$. Đường thẳng này có hệ số góc $f'(x_n)$ và đi qua điểm $(x_n, f(x_n)$ nên có phương trình $y = f'(x_n)(x - x_n) + f(x_n)$. Bây giờ ta tìm

nghiệm xấp xỉ mới x_{n+1} của phương trình này. Giải phương trình ta được $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.

Tuy nhiên phương pháp Newton - Raphson cũng có những giới hạn nhất định. Phương pháp này có thể không hoạt động nếu có các điểm uốn, điểm cực đại hoặc cực tiểu địa phương xung quanh điểm làm gốc x_0.

Phương pháp Newton - Raphson cũng có thể được mở rộng cho hệ phương trình. Cụ thể, đối với hệ phương trình vi phân tổng quát của bài tập này, công thức hồi quy của phương pháp Newton - Raphson sẽ có dạng:

$$\begin{bmatrix} R_{new} \\ J_{new} \end{bmatrix} = \begin{bmatrix} R_{old} \\ J_{old} \end{bmatrix} - \alpha . \begin{bmatrix} \frac{\partial f}{\partial R} & \frac{\partial f}{\partial J} \\ \frac{\partial g}{\partial R} & \frac{\partial g}{\partial J} \end{bmatrix}^{-1} \begin{bmatrix} f \\ g \end{bmatrix}$$

Trong công thức trên:

- $\begin{bmatrix} R_{new} \\ J_{new} \end{bmatrix}$ là các giá trị xấp xỉ mới.
- $\begin{bmatrix} R_{new} \\ J_{new} \end{bmatrix}$ là các giá trị cũ.
- $\begin{bmatrix} \frac{\partial f}{\partial R} & \frac{\partial f}{\partial J} \\ \frac{\partial g}{\partial R} & \frac{\partial g}{\partial J} \end{bmatrix}$ là ma trận Jacobian: Ma trận Jacobian là ma trận chứa các đạo hàm riêng bậc nhất của các hàm f, g theo các biến R, J.
- $\begin{bmatrix} f \\ g \end{bmatrix}$ là các hàm f, g trong hệ phương trình vi phân.
- α là một hệ số chọn. Trong 5 ví dụ bên dưới, ta chọn $\alpha = 1$.

Giá trị đầu vào cho công thức hồi quy phải ở dạng phương trình Euler ẩn, với các biến, các hàm được định nghĩa như phía dưới:

$$R_{k+1} = R_k + f(t_{k+1}, R_{k+1}, J_{k+1}) * h$$
$$J_{k+1} = J_k + g(t_{k+1}, R_{k+1}, J_{k+1}) * h$$

Ta đưa hai phương trình trên về dạng sau để áp dụng giải thuật Newton - Raphson:

$$R_k + f(t_{k+1}, R_{k+1}, J_{k+1}) * h - R_{k+1} = 0$$
$$J_k + g(t_{k+1}, R_{k+1}, J_{k+1}) * h - Jk + 1 = 0$$

5.2.2 Nguồn gốc

Lấy tích phân phương trình vi phân từ t_k tới $t_k + h$ và áp dụng các định lí cơ bản của tích phân và vi phân để có được:

$$R(t_k + h) - R(t_k) = \int_{t_k}^{t_k+h} f(t, R(t), J(t)\ \mathrm{d}t$$

Tính xấp xỉ tích phân trên bằng tổng Riemann phải, ta được:

$$\int_{t_k}^{t_k+h} f(t, R(t), J(t)\ \mathrm{d}t \approx h f(t_{k+1}, R(t_{k+1}), J(t_{k+1})$$

Kết hợp cả 2 phương trình, ta suy ra phương pháp Euler ẩn. Chứng minh tương tự với hàm $J(t)$.

5.2.3 Sai số cắt ngắn cục bộ

Giống với phương pháp Euler tường minh, sai số cắt ngắn cục bộ của phương pháp Euler ẩn cũng được tính theo công thức:

$$E(t_1) = \sqrt{[R(t_1) - R_1]^2 + [J(t_1) - J_1]^2}$$

Với $t_0 = t_1 - h$, áp dụng khai triển Taylor đối với $R(t_0)$ và $J(t_0)$, ta có:

$$\begin{cases} R(t_0) = R(t_1 - h) = R(t_1) - h.R'(t_1) + \frac{1}{2}h^2R"(t_1) + O(h^2) \\ J(t_0) = J(t_1 - h) = J(t_1) - h.J'(t_1) + \frac{1}{2}h^2J"(t_1) + O(h^2) \end{cases} \tag{36}$$

$$\Rightarrow \begin{cases} R_0 = R(t_1) - h.f(t_1, R_1, J_1) + \frac{1}{2}h^2f'(t_1, R_1, J_1) + O(h^2) \\ J_0 = J(t_1) - h.g(t_1, R_1, J_1) + \frac{1}{2}h^2g'(t_1, R_1, J_1) + O(h^2) \end{cases} \tag{37}$$

$$\Rightarrow \begin{cases} R(t_1) = R_1 - \frac{1}{2}h^2f'(t_1, R_1, J_1) + O(h^2) \\ J(t_1) = J_1 - \frac{1}{2}h^2g'(t_1, R_1, J_1) + O(h^2) \end{cases} \tag{38}$$

$$\Rightarrow \begin{cases} R(t_1) - R_1 = -\frac{1}{2}h^2f'(t_1, R_1, J_1) + O(h^2) \\ J(t_1) - J_1 = -\frac{1}{2}h^2g'(t_1, R_1, J_1) + O(h^2) \end{cases} \tag{39}$$

Bỏ qua phần $O(h^2)$ ta được:

$$\begin{cases} R(t_1) - R_1 = -\frac{1}{2}h^2f'(t_1, R_1, J_1) \\ J(t_1) - J_1 = -\frac{1}{2}h^2g'(t_1, R_1, J_1) \end{cases} \tag{40}$$

$$\Rightarrow E(t_1) \approx \sqrt{[-\frac{1}{2}h^2f'(t_1, R_1, J_1)]^2 + [-\frac{1}{2}h^2g'(t_1, R_1, J_1)]^2}$$
$$= \frac{1}{2}h^2\sqrt{f'^2(t_1, R_1, J_1) + g'^2(t_1, R_1, J_1)}$$

Như vậy, trong phương pháp Euler ẩn, $E(t_1)$ cũng xấp xỉ tỉ lệ thuận với h^2.

5.2.4 Nhược điểm của phương pháp Euler ẩn

Thứ nhất, do trong công thức của phương pháp Euler ẩn có ẩn số R_{k+1} và J_{k+1} ở cả 2 vế của hệ phương trình, nên để tìm được chúng, ta phải giải hệ phương trình đó. Quá trình này có thể tốn nhiều thời gian và phức tạp hơn so với phương pháp Euler tường minh.

Thứ hai, sai số cắt ngắn cục bộ tỉ lệ thuận với h^2, điều này đồng nghĩa với việc khi $h > 1$ sai số sẽ trở nên rất lớn, khi đó khoảng cách giữa nghiệm chính xác và nghiệm gần đúng sẽ là rất lớn, ảnh hưởng tới tính đúng đắn của bài toán

5.2.5 Ví dụ

Ví dụ 1: Tìm nghiệm xấp xỉ của hệ phương trình bằng phương pháp Euler ẩn:

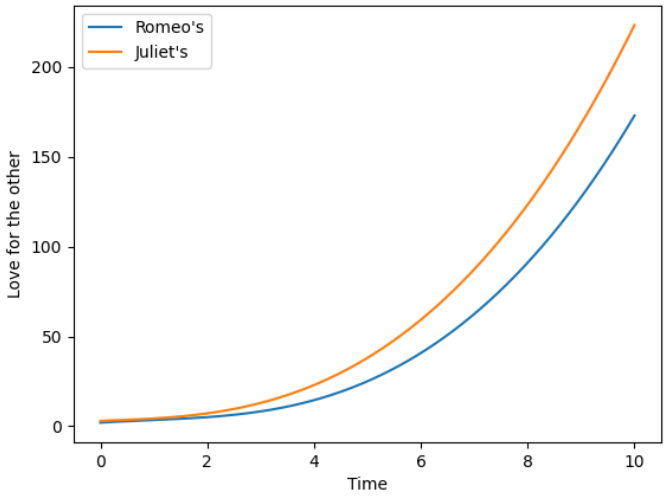
$$\begin{cases} R' = -R + J + \frac{sin(t+1)}{t+1} \\ J' = -R + J + t \\ R(0) = 2, J(0) = 3 \end{cases}$$

Giải

Dây là ví dụ về hệ phương trình vi phân tuyến tính không thuần nhất không thể tìm nghiệm chính xác với điều kiện ban đầu $R_0 = 2, J_0 = 3$.
Công thức hồi quy như sau:

$$\begin{cases} R_n = R_{n-1} + h(-R_n + J_n + \frac{sin(t_n+1)}{t_n+1}) \\ J_n = J_{n-1} + h(-R_n + J_n + t_n) \\ t_n = t_{n-1} + h \end{cases}$$

1. **Vẽ đồ thị nghiệm.**
   Biểu diễn R và J trên cùng một đồ thị theo biến thời gian t (hình 25):



Hình 25: *Solution figure Ví dụ 1*

Giải thích các đoạn code trong Python vẽ đồ thị ví dụ trên (các đoạn code này được ta định nghĩa trong file **implicit_euler_trajectory.py** và sẽ được dùng để vẽ **phase portrait**) như sau:

- Nạp các modules cần thiết, bao gồm: **matplotlib.pyplot**(module đồ họa vẽ đồ thị), **numpy**(module toán học xử lý ma trận và mảng), **numpy.linalg.inv**(module để nghịch đảo ma trận), **math**(hình 26).

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3  from numpy.linalg import inv
4  import math
```

Hình 26: *Nạp modules*

- Tạo 3 công thức của phương pháp Euler ẩn dưới dạng **Newton-Raphson** tương ứng với R (Romeo), J (Juliet), t (Thời điểm) (Ở đây, ta tạo thêm một công thức Euler ẩn so với kiến thức chuẩn bị cho biến thời gian t với **time step = h**)(hình 27):

old_value + h * f(new_value) - new_value

```python
def Romeo(R, new_R, new_J, new_T, dR, h):
    return R + h * dR(new_R, new_J, new_T) - new_R


def Juliet(J, new_R, new_J, new_T, dJ, h):
    return J + h * dJ(new_R, new_J, new_T) - new_J


def Time(T, new_T, h):
    return T + h - new_T
```

Hình 27: *Công thức Euler ẩn của **Romeo, Juliet, Time***

- Tạo ma trận **Jacobian** để tính toán đạo hàm riêng theo từng biến của 3 công thức Euler ẩn của hệ phương trình vi phân đa biến (Ma trận **Jacobian** bao gồm 9 hạng tử) (hình 28).

```python
def Jacobi(R, J, T, new_R, new_J, new_T, dR, dJ, h):
    jacob = np.ones((3, 3))
    d = 1e-9

    jacob[0, 0] = (Romeo(R, (new_R + d), new_J, new_T, dR, h) - Romeo(R, new_R, new_J, new_T, dR, h)) / d
    jacob[0, 1] = (Romeo(R, new_R, (new_J + d), new_T, dR, h) - Romeo(R, new_R, new_J, new_T, dR, h)) / d
    jacob[0, 2] = (Romeo(R, new_R, new_J, (new_T + d), dR, h) - Romeo(R, new_R, new_J, new_T, dR, h)) / d

    jacob[1, 0] = (Juliet(J, (new_R + d), new_J, new_T, dJ, h) - Juliet(J, new_R, new_J, new_T, dJ, h)) / d
    jacob[1, 1] = (Juliet(J, new_R, (new_J + d), new_T, dJ, h) - Juliet(J, new_R, new_J, new_T, dJ, h)) / d
    jacob[1, 2] = (Juliet(J, new_R, new_J, (new_T + d), dJ, h) - Juliet(J, new_R, new_J, new_T, dJ, h)) / d

    jacob[2, 0] = 0
    jacob[2, 1] = 0
    jacob[2, 2] = -1

    return jacob
```

Hình 28: *Ma trận Jacobian*

- Viết thuật toán **Newton-Raphson**, vòng lặp của thuật toán **Newton-Raphson** dừng lại khi sai số cắt ngắn cục bộ **LTE** có thể chấp nhận được (cụ thể $error < 10^{-9}$) tại một thời điểm t (hình 29).

```python
def Newton_Raphson(R, J, T, random_R, random_J, random_T, dR, dJ, h):
    X_init = np.ones((3, 1))
    X_init[0] = random_R
    X_init[1] = random_J
    X_init[2] = random_T

    X = np.ones((3, 1))

    error = 9e9
    tol = 1e-9

    while error > tol:
        jacob = Jacobi(R, J, T, X_init[0], X_init[1], X_init[2], dR, dJ, h)

        X[0] = Romeo(R, X_init[0], X_init[1], X_init[2], dR, h)
        X[1] = Juliet(J, X_init[0], X_init[1], X_init[2], dJ, h)
        X[2] = Time(T, X_init[2], h)

        X_new = X_init - np.matmul(inv(jacob), X)
        error = np.max(np.abs(X_new - X_init))
        X_init = X_new

    return [X_new[0], X_new[1], X_new[2]]
```

Hình 29: *Giải thuật **Newton-Raphson***

- Viết hàm **implicit_euler** theo phương pháp Euler ẩn sử dụng giải thuật **Newton-Raphson** để tính toán nghiệm xấp xỉ cho hệ phương trình vi phân (hình 30).

```
62  def Implicit_Euler(dR, dJ, R0, J0, tspan, dt):
63      t = np.arange(0, tspan, dt)
64      R = np.zeros(len(t))
65      J = np.zeros(len(t))
66      T = np.zeros(len(t))
67
68      R[0] = R0
69      J[0] = J0
70      T[0] = 0
71
72      random_R = 10
73      random_J = 10
74      random_T = dt
75
76      for i in range(1, len(t)):
77          R[i], J[i], T[i] = Newton_Raphson(R[i - 1], J[i - 1], T[i - 1], random_R, random_J, random_T, dR, dJ, dt
78
79          random_R = R[i]
80          random_J = J[i]
81          random_T = T[i]
82
83      return [t, R, J]
```

Hình 30: *Phương pháp Euler ẩn*

- Cuối cùng ta nhập các công thức của R', J', các giá trị ban đầu R_0, J_0 đã cho trong hệ phương trình vi phân. Chọn bước nhảy thời gian **time step** $h = 0.001$ và sau đó vẽ hình (hình 31).

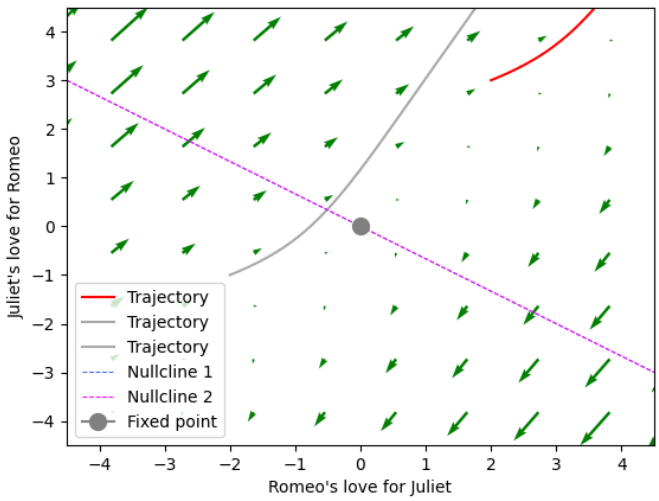
```
86  def dR(R, J, T):
87      return -R + J + math.sin(T + 1) / (T + 1)
88
89
90  def dJ(R, J, T):
91      return -R + J + T
92
93
94  R0 = 2
95  J0 = 3
96
97  t, R, J = Implicit_Euler(dR, dJ, R0, J0, 10, 0.001)
98  plt.plot(t, R)
99  plt.plot(t, J)
100 plt.xlabel("Time")
101 plt.ylabel("Love for the other")
102 plt.legend(["Romeo's", "Juliet's"])
103 plt.show()
```

Hình 31: *Nhập input và xuất ra output*

2. **Vẽ Phase portrait**
   Phase portrait của hệ phương trình vi phân

$$\begin{cases} R' = -R + J + \frac{sin(t+1)}{t+1} \\ J' = -R + J + t \end{cases}$$

với điều kiện ban đầu bất kì trong mặt phẳng RJ được biểu diễn trong hình 32.
Đoạn code Python được sử dụng để vẽ phase portrait tương tự như **Bài tập 2**:



Hình 32: *Phase portrait figure*

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from implicit_euler_trajectory import dR, dJ, R0, J0
Sts = [[R0, J0], [R0 + 4, J0 + 4], [R0 - 4, J0 - 4]]
# ODE System


def ivpSys(s, t, dR, dJ):
    R, J = s
    T = t
    dRdt = dR(R, J, T)
    dJdt = dJ(R, J, T)
    return [dRdt, dJdt]


# Vector R', J' at t = 0 with 144 values of pair (R0, J0)
y1 = np.linspace(-6, 6, 12)
y2 = np.linspace(-6, 6, 12)
Y1, Y2 = np.meshgrid(y1, y2)
t = 0
u, v = np.zeros(Y1.shape), np.zeros(Y2.shape)
NI, NJ = Y1.shape
for i in range(NI):
    for j in range(NJ):
        x = Y1[i, j]
        y = Y2[i, j]
        yvector = ivpSys([x, y], t, dR, dJ)
        u[i, j] = yvector[0]
        v[i, j] = yvector[1]
```

```
Q = plt.quiver(Y1, Y2, u, v, color='g')
# Trajectory of Sts with 200 values of t between (0,5)
tspan = np.linspace(0, 5, 200)
for elementSt, St in enumerate(Sts):
    sol = odeint(ivpSys, St, tspan, args=(dR, dJ))
    if elementSt == 0:
        color = "r"
    else:
        color = "darkgray"
    plt.plot(sol[:, 0], sol[:, 1], color, label='Trajectory')
# Figure phase portrait
x = np.linspace(-4.5, 4.5, 100)


def fx(x, a, b):
    return -a * x / b


plt.plot(x, fx(x, 2, 3), linestyle='dashed', linewidth=0.75,
         color='royalblue', label='Nullcline 1')
plt.plot(x, fx(x, 2, 3), linestyle='dashed', linewidth=0.75,
         color='magenta', label='Nullcline 2')
plt.plot(0, 0, "red", marker="o", markersize=10.0,
         color='grey', label='Fixed point')
plt.xlabel("Romeo's love for Juliet")
plt.ylabel("Juliet's love for Romeo")
plt.legend()
plt.xlim([-4.5, 4.5])
plt.ylim([-4.5, 4.5])
plt.show()
```

Ví dụ 2: Tìm nghiệm xấp xỉ của hệ phương trình bằng phương pháp Euler ẩn:

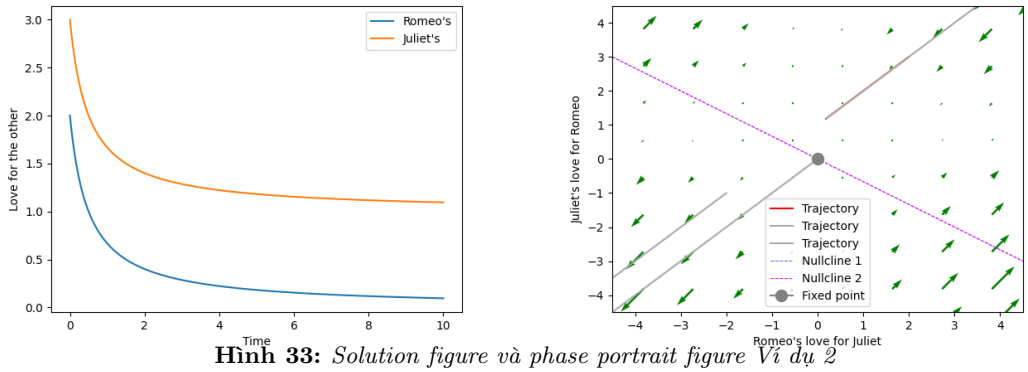
$$\begin{cases} R' = R(1-J) \\ J' = R(1-J) \\ R(0) = 2, J(0) = 3 \end{cases}$$

Giải

Đây là ví dụ về hệ phương trình vi phân thường cấp 1 tổng quát không thể tìm nghiệm chính xác với điều kiện ban đầu $R0 = 2, J0 = 3$.
Công thức hồi quy như sau:

$$\begin{cases} R_n = R_{n-1} + h(R_n - R_n * J_n) \\ J_n = J_{n-1} + h(R_n - R_n * J_n) \\ T_n = T_{n-1} + h \end{cases}$$

Chọn $h = 0.001$. Đoạn code trong Python dùng để vẽ đồ thị và phase portrait tương tự như **ví dụ 1**, chỉ thay đổi $dR, dJ, R0, J0$. Hình vẽ được biểu diễn ở hình 33.



Hình 33: *Solution figure và phase portrait figure Ví dụ 2*

Ví dụ 3: Tìm nghiệm xấp xỉ của hệ phương trình bằng phương pháp Euler ẩn:

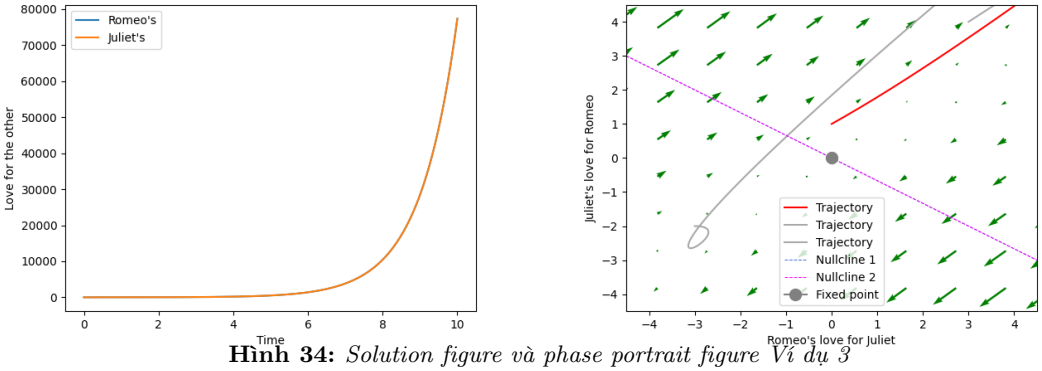
$$\begin{cases} R' = -3R + 4J + sin(t) \\ J' = -2R + 3J + t \\ R(0) = 0, J(0) = 1 \end{cases}$$

Giải

Đây là ví dụ về hệ phương trình vi phân tuyến tính không thuần nhất với hệ số không đổi và giá trị ban đầu $R0 = 0, J0 = 1$.
Công thức hồi quy như sau:

$$\begin{cases} R_n = R_{n-1} + h(-3R_n + 4J_n + \sin(T_n)) \\ J_n = J_{n-1} + h(-2R_n + 3J_n + T_n) \\ T_n = T_{n-1} + h \end{cases}$$

Chọn $h = 0.001$. Đoạn code trong Python dùng để vẽ đồ thị và phase portrait tương tự như **ví dụ 1**, chỉ thay đổi $dR, dJ, R0, J0$. Hình vẽ được biểu diễn ở hình 34.



Hình 34: *Solution figure và phase portrait figure Ví dụ 3*

Ví dụ 4: Tìm nghiệm xấp xỉ của hệ phương trình bằng phương pháp Euler ẩn:

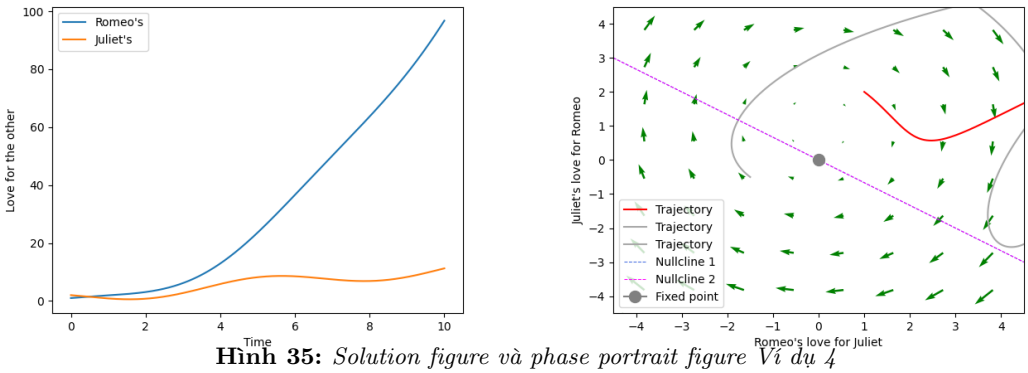
$$\begin{cases} R' = J + t - 1 \\ J' = -R + t^2 \\ R(0) = 1, J(0) = 2 \end{cases}$$

Giải

Đây là ví dụ về hệ phương trình vi phân tuyến tính không thuần nhất với hệ số không đổi và giá trị ban đầu $R0 = 1, J0 = 2$.
Công thức hồi quy như sau:

$$\begin{cases} R_n = R_{n-1} + h(J_n + T_n - 1) \\ J_n = J_{n-1} + h(-R_n + T_n^2) \\ T_n = T_{n-1} + h \end{cases}$$

Chọn $h = 0.001$. Đoạn code trong Python dùng để vẽ đồ thị và phase portrait tương tự như **ví dụ 1**, chỉ thay đổi $dR, dJ, R0, J0$. Hình vẽ được biểu diễn ở hình 35.



Hình 35: *Solution figure và phase portrait figure Ví dụ 4*

Ví dụ 5: Tìm nghiệm xấp xỉ của hệ phương trình bằng phương pháp Euler ẩn:

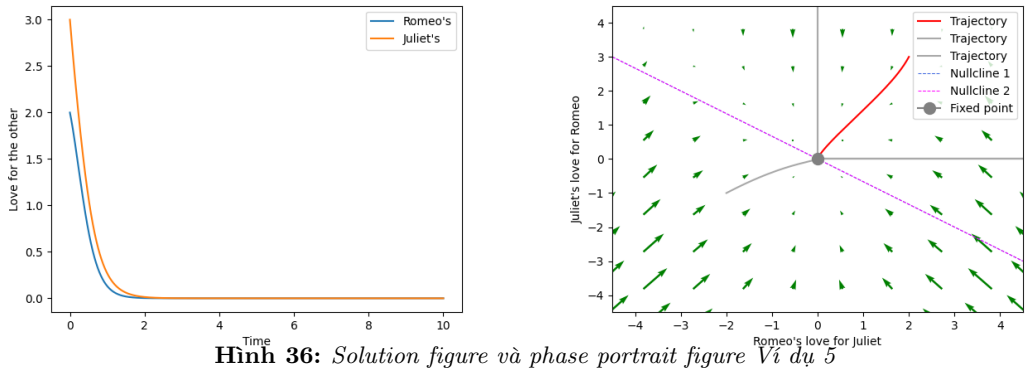
$$\begin{cases} R' = -4R + RJ \\ J' = R^2 - 3J \\ R(0) = 2, J(0) = 3 \end{cases}$$

Giải

Đây là ví dụ về hệ phương trình vi phân thường cấp 1 tổng quát không thể tìm nghiệm chính xác với điều kiện ban đầu $R0 = 2, J0 = 3$.
Công thức hồi quy như sau:

$$\begin{cases} R_n = R_{n-1} + h(-4R_n + R_n J_n) \\ J_n = J_{n-1} + h(R_n^2 - 3J_n) \\ T_n = T_{n-1} + h \end{cases}$$

Chọn $h = 0.001$. Đoạn code trong Python dùng để vẽ đồ thị và phase portrait tương tự như **ví dụ 1**, chỉ thay đổi $dR, dJ, R0, J0$. Hình vẽ được biểu diễn ở hình 36.



Hình 36: *Solution figure và phase portrait figure Ví dụ 5*

Link toàn bộ bài làm của nhóm:
https://github.com/ngyngcphu/Dynamics-of-Love

References

[1] Vladimir I Arnold. *Ordinary Differential Equations.* Springer Science & Business Media, 1992.

[2] Morris W Hirsch and Stephen Smale. *Differential Equations, Dynamical Systems, and Linear Algebra.* Academic Press, 1974.

[3] David G Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications,* volume 1. Wiley New York, 1979.

[4] Steven H Strogatz. Love affairs and differential equations. *Mathematics Magazine,* 61(1):35–35, 1988.

[5] Phương trình vi phân - Phương pháp giá trị riêng giải hệ phương trình vi phân tuyến thuần nhất
https://www.youtube.com/watch?v=LVbsb6mAFZ4

[6] Phase portraits.
https://www.math.colostate.edu/~gerhard/M345/CHP/ch9_3-4.pdf

[7] Giải thuật Newton-Rahpson
https://brilliant.org/wiki/newton-raphson-method/

[8] Phương pháp Implicit-Euler
https://skill-lync.com/student-projects/Solving-a-system-of-ODEs-using-Implicit-Euler-met