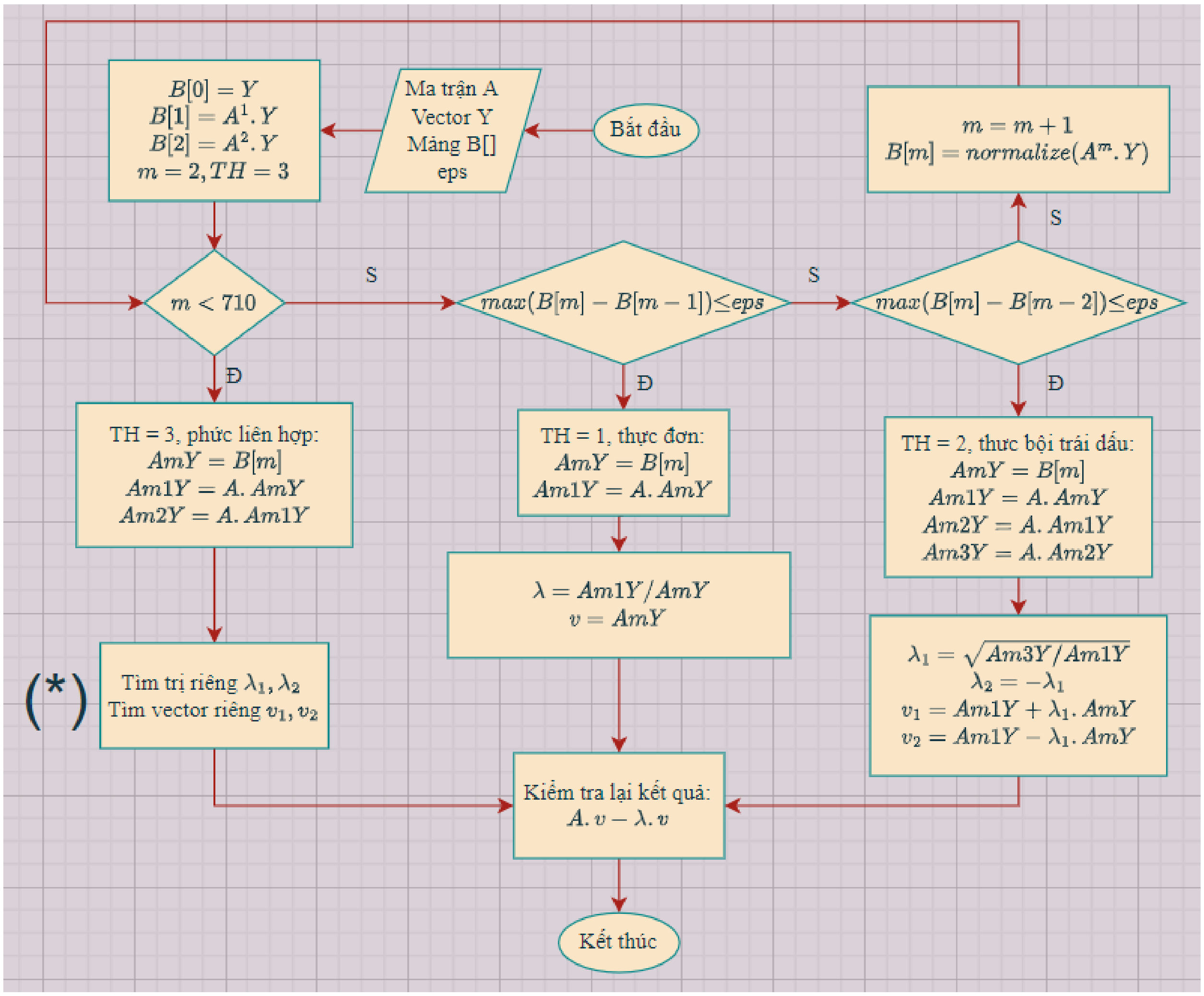
GVHD: Hà Thị Ngọc Yến

Họ và tên sinh viên: Trần Tuấn Kiệt

MSSV: 20206290

Phương pháp luỹ thừa tìm giá trị riêng trội

1. Thuật toán



❖ Bổ sung và giải thích các gói sử dụng:

➢ Với các tác tính toán ở bước (*) trong sơ đồ khối:

- Giải thích hình thành công thức: 3 vector AmY, Am1Y, Am2Y hình thành một tổ hợp tuyến tính:

$$Am2Y - (\lambda_1 + \lambda_2).Am1Y + \lambda_1.\lambda_2.AmY = 0$$

Đặt lamda1 + lamda2 = -p; lamda1.lamda2 = q

⇨ $Am2Y + p.Am1Y + q.AmY = 0$ (1)

⇔ $Z^2 + p.Z + q = 0$

Nghiệm của phương trình trên chính là 2 trị riêng phức cần tìm

- Để tìm p, q ta có:

$$\begin{vmatrix} Z^2 & Z & 1 \\ (Am2Y)_1 & (Am1Y)_1 & (AmY)_1 \\ (Am2Y)_2 & (Am1Y)_2 & (AmY)_2 \end{vmatrix} = 0$$

(chỉ số 1 biểu bị thành phần thứ 1 của vector)

⬄

$$\begin{vmatrix} Z^2 & Z & 1 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = 0$$

⇨

$$p = \frac{a_1.c_2 - c_1.a_2}{c_1.b_2 - b_1.c2}$$

$$q = \frac{b_1.a_2 - a_1.b_2}{c_1.b_2 - b_1.c2}$$

- Ta có: $\Delta = p^2 - 4.q$

⇨

$$\lambda_1 = \frac{-b - i.\sqrt{|\Delta|}}{2a}$$

$$\lambda_2 = \frac{-b + i.\sqrt{|\Delta|}}{2a}$$

$$v_1 = normalize(Am2Y - \lambda_2.Am1Y)$$

$$v_2 = normalize(Am2Y - \lambda_1.Am1Y)$$

- Hàm normalize:
  Input: Vector gồm n thành phần chưa được chuẩn hoá
  Output: Vector gồm n thành phần đã được chuẩn hoá
  Mô tả hàm:
  - Tìm vị trí thành phần thứ i của vector có giá trị tuyệt đối lớn nhất.
  - Chia tất cả các thành phần của vector cho thành phần thứ i.

2. Ví dụ test

VD1: Trường hợp nghiệm đơn, thực, hội tụ nhanh:

- Kết quả test: sai số 1e^-6

```
Ma trận đầu vào:
[[2.1 1.9 1.8 1.7]
 [1.9 2.  1.7 1.8]
 [1.8 1.7 2.1 1.9]
 [1.7 1.8 1.9 2. ]]
vector Y:
[[2]
 [1]
 [1]
 [2]]
Ma trận rơi vào TH1 sau 6 lần thực hiện luỹ thừa:
Giá trị riêng tìm được là:  [7.45034719]
Vector riêng tìm được là:
[[1.        ]
 [0.98620756]
 [1.        ]
 [0.98620755]]
Kiểm tra lại kết quả:
A.X - lamda.X =
 [[0.00000000e+00]
 [3.27313554e-09]
 [3.32236603e-08]
 [2.92130222e-08]]
```

VD2: Trường hợp đơn, thực, hội tụ chậm:

- Kết quả test: sai số 1e^-6

```
Ma trận đầu vào:
[[ 5.  -2.  -0.5  1.5]
 [-2.   5.   1.5 -0.5]
 [-0.5  1.5  5.  -2. ]
 [ 1.5 -0.5 -2.   5. ]]
vector Y:
[[1]
 [2]
 [1]
 [1]]
Ma trận rơi vào TH1 sau 25 lần thực hiện luỹ thừa:
Giá trị riêng tìm được là:  [8.99999838]
Vector riêng tìm được là:
[[-0.99999998]
 [ 1.        ]
 [ 0.99999917]
 [-0.99999915]]
Kiểm tra lại kết quả:
A.X - lamda.X =
 [[ 0.00000000e+00]
 [-7.84163703e-08]
 [ 3.24264575e-06]
 [-3.32106212e-06]]
```

VD3: Trường hợp nghiệm thực, trái dấu, hội tụ chậm

- Kết quả test: sai số 1e^-6

```
Ma trận đầu vào:
[[ 4.   0.   0.   0. ]
 [ 0.  -4.   0.   0. ]
 [ 1.   2.   1.5  4. ]
 [ 5.   3.   1.   1. ]]
vector Y:
[[1]
 [2]
 [2]
 [2]]
Ma trận rơi vào TH2 sau 56 lần thực hiện luỹ thừa:
lamda1 =  4.0
v1 =
[[0.15217472]
 [0.        ]
 [1.        ]
 [0.58695729]]
Kiểm tra lại kết quả:
A.X - lamda.X =
 [[0.00000000e+00]
 [0.00000000e+00]
 [3.88215008e-06]
 [1.71354653e-06]]
```

```
lamda2 =  -4.0
v2 =
[[-0.        ]
 [ 1.        ]
 [ 0.08510463]
 [-0.61702205]]
A.X - lamda.X =
 [[ 0.00000000e+00]
 [ 0.00000000e+00]
 [-1.27555686e-05]
 [-5.63019456e-06]]
```

VD4: Trường hợp nghiệm phức liên hợp:

```
Ma trận đầu vào:
[[-2.  1.  1.  1.]
 [-7. -5. -2. -1.]
 [ 0. -1. -3. -2.]
 [-1.  0. -1.  0.]]
vector Y:
[[1]
 [1]
 [2]
 [1]]
Ma trận rơi vào TH3:
Phươnng trình định thức = 0 tìm đc: Z^2 + 8.0Z + 19.999999999999996 = 0
lamda1 =  (-4-1.9999999999999991j)
v1 =
[[-0.2+0.4j]
 [ 1. -0.j ]
 [ 0.2-0.4j]
 [-0. -0.j ]]
A.X - lamda.X =
 [[ 2.22044605e-16-2.22044605e-16j]
 [ 8.88178420e-16-8.88178420e-16j]
 [-2.22044605e-16+2.22044605e-16j]
 [ 0.00000000e+00+0.00000000e+00j]]
```

```
lamda2 =  (-4+1.9999999999999991j)
v2 =
[[-0.2-0.4j]
 [ 1. -0.j ]
 [ 0.2+0.4j]
 [-0. -0.j ]]
A.X - lamda.X =
 [[ 2.22044605e-16+2.22044605e-16j]
 [ 8.88178420e-16+8.88178420e-16j]
 [-2.22044605e-16-2.22044605e-16j]
 [ 0.00000000e+00+0.00000000e+00j]]
```

3. Nhận xét về phương pháp:

- Tốc độ hội tụ phụ thuộc vào độ trội của giá trị riêng trội so với các giá trị riêng còn lại
- Có thể tìm được giá trị riêng trội phức liên hợp
- Lập trình dễ hơn so với Danileski
- Có thể gây ra tràn số nếu không có bước chuẩn hoá vector

- Thỉnh thoảng có thể gây ra nhầm kết quả khi chọn vector Y ban đầu không là tổ hợp tuyến tính của các vector riêng.