



BÁO CÁO ĐỒ ÁN 1



Giảng viên: TS.Dương Việt Hằng

Trợ giảng: Trần Hà Sơn

Sinh viên thực hiện: 21120580 – Trần Thị Kim Trinh

Mục lục

Hàm Gauss_elimination(A):	2
Mô tả hàm	2
Ý tưởng thực hiện:	2
Hàm back_substitution(A):	3
Mô tả hàm:	3
Ý tưởng thực hiện	3
Hàm rank():	3
Hàm back_substitution(A):	3
Đặc tả file input:	4

Hàm Gauss_elimination(A):

Mô tả hàm

- Input: ma trận A (ma trận mở rộng của hệ phương trình tuyến tính)
- Output: dạng bậc thang tương ứng của ma trận A
- Minh họa:

Ma trận A:	Gauss_elimination(A):
[1, 2, 3, 1]	[1, 2, 3, 1]
[4, 5, 6, 1]	[0, 3, 6, 3]
[7, 7, 9, 0]	[0, 0, 6, 0]

Ý tưởng thực hiện:

- Xét từng dòng trong ma trận, đặt biến “chuẩn” là vị trí cột của phần tử làm chuẩn, dòng đang xét hiện tại là dòng i
 - o Do điều kiện dừng là xét hết các dòng, chuẩn là vị trí của cột nên sẽ có trường hợp chưa xét hết dòng thì chuẩn đã ra ngoài phạm vi cột \rightarrow thêm điều kiện dừng

```
[1, 2, 3, 1]
[0, 4, 6, 3]
[0, 0, -16, -6]
[0, 0, 0, 88]
[0, 0, 0, 0]
```

Ví dụ: chỉ cần thực hiện chuẩn hóa đến vị trí chuẩn đã được khoanh tròn là có thể dừng, không cần xét dòng dưới

- o Nếu giá trị tại cột chuẩn của dòng i bằng 0, tìm trong cột đó dòng $i+1, i+2, \dots$ có hệ số khác 0 thì đổi vị trí 2 dòng đó với nhau
- o Nếu sau khi đổi, hệ số tại cột chuẩn của dòng hiện tại vẫn bằng 0 (do tất cả phần tử trong cột chuẩn nằm dưới hàng hiện tại đều bằng 0) thì chuyển sang cột kế tiếp (chuẩn = chuẩn + 1), và vẫn xét ở dòng hiện tại

```
> function variables
> 0: [4, -2, -4, 2, 1]
> 1: [0, 0, -24, -8, -6]
> 2: [0, 0, -144, -160, -36]
> 3: [0, 0, 48, -64, 12]
```

Như ví dụ trên, chuẩn đang là 1 (do sau khi $A[0][0]$ được nhận làm phần tử chuẩn thì chuẩn tăng lên 1, dòng đang xét cũng tương tự), toàn bộ các giá trị ở cột 1 phía dưới dòng 1 đều bằng 0 nên bắt buộc phải chuyển qua tìm giá trị chuẩn ở cột 2, dòng không thay đổi

- o Sau khi đã tìm được hệ số thích hợp làm chuẩn, xét các phần tử còn lại trong cột chuẩn, ở dưới hàng hiện tại, nếu hệ số đó khác 0 thì thực hiện chuẩn hóa hàng đó
- o Làm xong các bước trên thì tăng chuẩn lên 1 đơn vị, xét tiếp hàng tiếp theo (tăng hàng đang xét lên 1 đơn vị)

Hàm back_substitution(A):

Mô tả hàm:

- Input: ma trận A là ma trận có dạng bậc thang từ ma trận mở rộng của hệ phương trình
- Output:
 - o nghiệm của hệ phương trình nếu có nghiệm
 - o một cơ sở của không gian nghiệm nếu trường hợp vô số nghiệm
 - o thông báo hệ phương trình vô nghiệm nếu hệ vô nghiệm
- Hàm có sử dụng thêm hàm rank() để tính hạng của ma trận A

Ý tưởng thực hiện

Hàm rank():

- Input là ma trận A, output là một số thể hiện hạng của A
- Trước khi tính toán, cần phải đưa A về dạng bậc thang
- Ban đầu, gán hạng bằng số dòng của A, sau đó dò lần lượt qua tất cả các dòng, nếu có N dòng toàn bộ phần tử đều bằng 0 thì hạng của ma trận A là số dòng - N

Hàm back_substitution(A):

- Gọi làm hàm Gauss_elimination(A) để phòng trường hợp ma trận A đưa vào chưa ở dạng bậc thang rút gọn
- Tính hạng của ma trận hệ số (ma trận temp - có được bằng cách chép lại toàn bộ ma trận A, nhưng bỏ cột cuối cùng), và hạng của ma trận mở rộng (A) của hệ phương trình
 - o Nếu 2 hạng tính được ở trên bằng nhau và bằng số ẩn của hệ: hệ có nghiệm duy nhất:
 - Tạo một mảng con gồm 1 dòng và có số cột bằng số ẩn của hệ để lưu nghiệm
 - Bắt đầu tính từ hàng dưới cùng mà có hệ số của nghiệm khác 0 (hàng thứ $\text{rank}(A) - 1$ (do python đánh số thứ tự từ 0))
 - Nghiệm (x_i) được tính theo công thức: Ban đầu x_i sẽ là bằng hệ số tự do b_i cùng dòng, sau đó trừ đi tổng của (các hệ số j cùng dòng nhân x_j , (x_j đã được tính ở trước do thay thế lùi), j thuộc $[i+1, \text{số thứ tự biến cuối}]$) rồi chia cho hệ số của x_i
 - o Nếu 2 hạng trên bằng nhau nhưng khác số ẩn: hệ có vô số nghiệm:
 - Bước 1: Số nghiệm tự do = số ẩn - hạng của ma trận (A)
 - Bước 2: Tạo một ma trận tạm thời để hỗ trợ việc tìm cơ sở, có số dòng là số dòng của ma trận A + số nghiệm tự do, sao chép các giá trị từ A qua ma trận mới được tạo, các phần tử của các dòng dư ra gán bằng 0

Ví dụ:

```
0: [2, 5, 8, 6, 1, 48, 2]
1: [0, 9, 30, 24, -5, 238, 6]
2: [0, 0, 4, 8, 0, 88, -6]
```

ma trận A

```

0: array([ 2.,  5.,  8.,  6.,  1., 48.,  2.])
1: array([ 0.,  9., 30., 24., -5., 238.,  6.])
2: array([ 0.,  0.,  4.,  8.,  0., 88., -6.])
3: array([0., 0., 0., 0., 0., 0., 0.])
4: array([0., 0., 0., 0., 0., 0., 0.])
5: array([0., 0., 0., 0., 0., 0., 0.])

```

Do A có 3 nghiệm tự do nên thêm vào 3 dòng

- Bước 3: Xác định các nghiệm tự do, ứng với mỗi nghiệm tự do thì hệ số của nghiệm ấy bằng 1 (hệ số các nghiệm khác bằng 0), hệ số tự do là một số ngẫu nhiên

```

> 0: array([ 2.,  5.,  8.,  6.,  1., 48.,  2.])
> 1: array([ 0.,  9., 30., 24., -5., 238.,  6.])
> 2: array([ 0.,  0.,  4.,  8.,  0., 88., -6.])
> 3: array([0., 0., 0., ①, 0., 0., ②]) Phương trình mới của x4
> 4: array([0., 0., 0., 0., ①, 0., ③]) Phương trình mới của x5
> 5: array([0., 0., 0., 0., 0., ①, ④]) Phương trình mới của x6

```

- Bước 4: Lấy hệ phương trình có được ở bước trên đi giải lại, ta được 1 nghiệm cơ bản
- Lặp lại các bước 3 và bước 4 n lần (số nghiệm tự do) lần, ta được n nghiệm cơ bản => đây là cơ sở cần tìm

Giải lần 1

Phương trình cơ sở nghiệm, cơ sở 1 cơ sở là:

```

[[ -82.94444444  66.77777778 -29.5          3.          4.
   1.          ]
 -139.27777778 117.11111111 -53.5          4.          3.
   2.          ]
 -79.83333333  63.33333333 -27.5          2.          5.
   1.          ]

```

Đổi hệ số ngẫu nhiên, giải lần 2

Đổi hệ số ngẫu nhiên, giải lần 3

- Nếu 2 hạng trên khác nhau: hệ vô nghiệm, xuất ra thông báo hệ pt vô nghiệm

Đặc tả file input:

- Tên file: "INPUT.txt"
- Dòng đầu tiên chứa 2 số m và n, với m là số dòng, n là số cột
- Các m dòng tiếp theo, mỗi dòng là n phần tử trong cùng một hàng, giữa các số trong cùng một hàng các nhau bởi khoảng trắng

≡ INPUT.txt

```

1  3 4
2  1 2 3 1
3  4 5 6 1
4  7 7 9 0

```