**TOÁN ỨNG DỤNG THỐNG KÊ**



**TỐI ƯU LỒI VÀ PHƯƠNG PHÁP BÌNH PHƯƠNG TỐI TIỂU**

**BÁO CÁO ĐỒ ÁN 4**



Sinh viên thực hiện : 21120580 – Trần Thị Kim Trinh

Sinh viên thực hiện: 21120580 – Trần Thị Kim Trinh

Giảng viên: TS.Dương Việt Hằng

Trợ giảng: Trần Hà Sơn

Mục lục

[Một số thư viện có sử dụng trong bài làm 2](#_Toc135499290)

[Bài 1: Kiểm tra tính lồi lõm của hàm số 2](#_Toc135499291)

[Ý tưởng 2](#_Toc135499292)

[Áp dụng thực tế 2](#_Toc135499293)

[Bài 2: Tìm công thức giảm lượng thuốc đối với bệnh nhân theo thời gian 3](#_Toc135499294)

[Ý tưởng 3](#_Toc135499295)

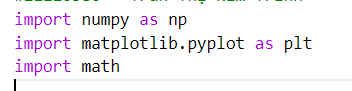
[Áp dụng thực tế 3](#_Toc135499296)

[Bài 3: Tìm mối liên hệ giữa độ giảm trọng lượng của hợp chất và khoảng thời gian mà hợp chất tiếp xúc với không khí 4](#_Toc135499297)

[Ý tưởng 4](#_Toc135499298)

[Áp dụng 4](#_Toc135499299)

# Một số thư viện có sử dụng trong bài làm

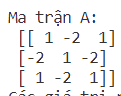


# Bài 1: Kiểm tra tính lồi lõm của hàm số

## Ý tưởng

* Tạo ma trận A như đề đã cung cấp
* Tìm các giá trị riêng của ma trận A và xét
  + Nếu tất cả giá trị riêng lớn hơn 0 (hay lớn hơn hoặc bằng 0) thì ma trận đã cho xác định dương (hay nửa xác định dương) 🡪 hàm số lồi
  + Nếu tất cả giá trị riêng nhỏ hơn 0 (hay nhỏ hơn hoặc bằng 0) thì ma trận xác định âm (nửa xác định âm) 🡪 hàm số lõm
  + Nếu không thuộc các trường hợp trên thì hàm só không lồi không lõm
* Tìm điểm dừng (điểm cực tiểu đối với hàm lồi, điểm cực đại toàn cực với hàm lõm) :
  + Điểm dừng sẽ là nghiệm của hệ phương trình ∇ f = 2Ax + q = 0 (nếu có)

## Áp dụng thực tế

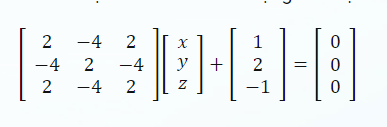
* Tạo ma trận A,q bằng np.array()với dữ liệu như đề đã cung cấp, ta được kết quả sau:
* 
* Tìm các giá trị riêng của ma trận A bằng hàm np.linalg.eig(A)[0]

Ta được các giá trị riêng của A:

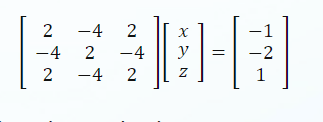


Do các giá trị riêng vừa có âm vừa có dương và có cả số 0, nên hàm số đề cho không lồi không lõm 🡪 không có điểm dừng

* Tìm điểm dừng: do đã dự đoán hàm không có điểm dừng, ta sẽ thử kiểm lại bằng cách giải phương trình ∇ f = 2Ax + q, ∇ f = 0 🡪 giải hệ phương trình 2Ax + q = 0



**↓**

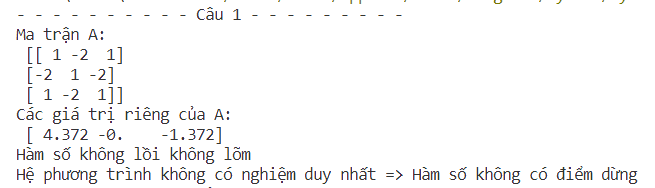


Để giải phương trình trên, gọi hàm np.linalg.solve(2A,-q), kết hợp với try .. catch để bắt lỗi khi phương trình không có nghiệm duy nhất, ta được kết quả



Vậy hàm số đã cho không lồi không lõm, không có điểm cực đại hay cực tiểu

Kết quả có được sau khi chạy code của bài 1:



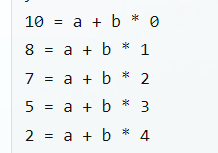
# Bài 2: Tìm công thức giảm lượng thuốc đối với bệnh nhân theo thời gian

## Ý tưởng

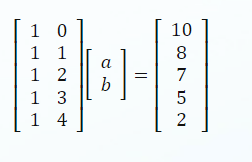
* Phương trình tuyến tính cần tìm sẽ có dạng **y = a + bx**
* Thế x và y đã cho vào phương trình trên, ta được hệ phương trình đã biết y và x, từ đó áp dụng phương pháp bình phương tối tiểu để tìm a b
* Thế a b đã tính toán được lại vào phương trình tuyến tính, ta đã tìm được công thức giảm lượng thuốc đối với bệnh nhân theo thời gian

## Áp dụng thực tế

* Từ x và y đề bài đã cho, ta lập được hệ phương trình sau:



chuyển về dạng ma trận:



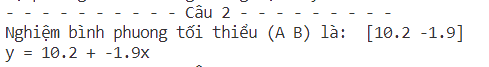
Đặt tên các ma trận trên từ trái sang phải là A,x,B; tạo lập các ma trận A B bằng lệnh np.array()

* Áp dụng phương pháp bình phương tối tiểu để tìm x, ta được kết quả sau:



* Vậy phương trình của ta sẽ có dạng: **y = 10.2 – 1.9 \* x**

Kết quả có được khi chạy code của bài 2:



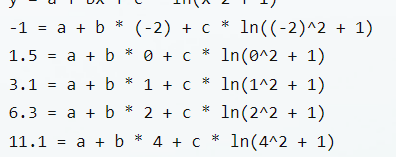
# Bài 3: Tìm mối liên hệ giữa độ giảm trọng lượng của hợp chất và khoảng thời gian mà hợp chất tiếp xúc với không khí

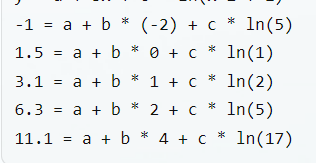
## Ý tưởng

* Phương trình cần tìm có dạng: **y = a + bx + c ln(x^2 + 1)**
* Thế x và y đã cho vào phương trình trên, ta được hệ phương trình đã biết y và x, từ đó áp dụng phương pháp bình phương tối tiểu để tìm a b c
* Vẽ đồ thị nhờ các hàm trong thư viện matplotlib.pyplot, và thế x = 6.5 vào phương trình tìm được để dự đoán y

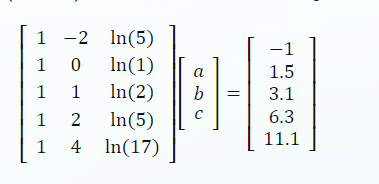
## Áp dụng

* Từ x y và dạng phương trình đã cho, ta được hệ phương trình sau:





Chuyển về dạng ma trận:



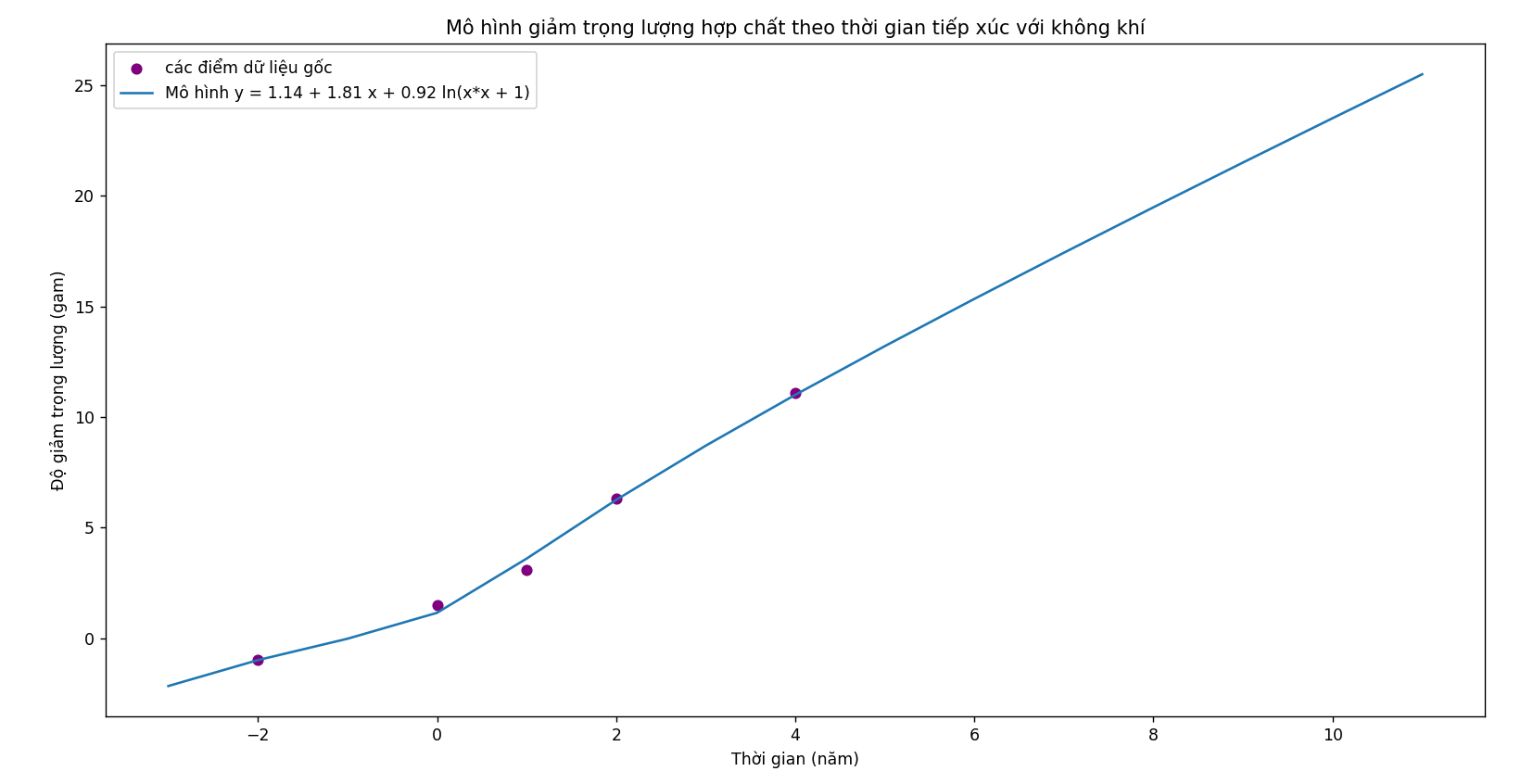
Áp dụng phương pháp bình phương tối tiểu để tìm , ta được kết quả sau:



Vậy a ≈ 1.14, b ≈ 1.81, c ≈ 0.92 🡪 phương trình cần tìm có dạng

**y = 1.14 + 1.81 \* x + 0.92 \* ln(x^2 + 1)**

b. Vẽ minh họa và dự đoán:



* Thế x = 6.5, ta tính toán được y ≈ 16.391

Kết quả chạy chương tình ở phần dự đoán:



c. Trả lời câu hỏi: **“Có nên dùng mô hình y=a+bx+cln(x) hoặc y = a + bx + c/x để xấp xỉ dữ liệu trên không? Vì sao?”**

🡺 Không nên, vì các hàm số trên không liên tục trên tập số thực ( R ), cụ thể là:

* Hàm số y=a+bx+cln(x) chỉ xác định khi x > 0, nhưng trong tập dữ liệu của ta tồn tại   
  x = -2 và x = 0 🡪 sẽ không biểu diễn được khi x tại các điểm này
* Hàm số y = a + bx + c/x cũng không nên dùng vì hàm số này chỉ xác định khi x khác 0, nhưng trong tập dữ liệu cuả chúng ta lại có điểm x = 0 🡪 sẽ không biểu diễn được khi x=0