# Phương pháp Cholesky giải phương trình Ax=B

## THUẬT TOÁN

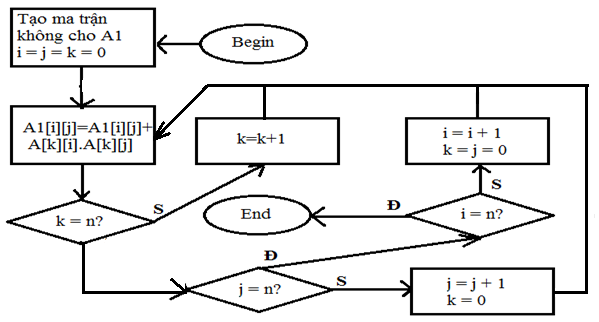
**Sơ đồ khối mô tả thuật toán**

Do phải thực hiện nhiều phép toán phức tạp nên ta chia sơ đồ thuật toán thành nhiều phần:

1. Cholesky1: Tính A1 = At.A
2. Cholesky2: Tính B1 = At.B
3. Cholesky : Tìm ma trận tam giác trên S theo khai triển Cholesky của A1
4. Cholesky3: Giải phương trình S.Y=B để tìm Y
5. Cholesky4: Giải phương trình St.X=Y để tìm X
6. Thuật toán cuối cùng.

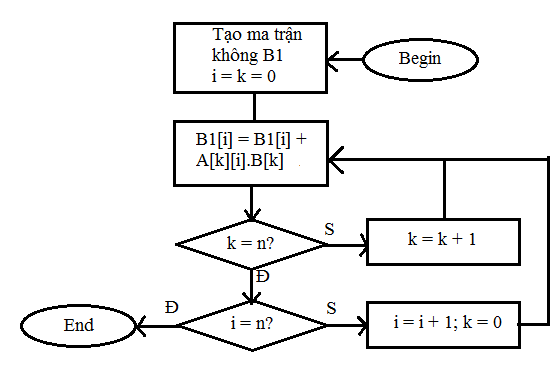
Ta sẽ đi vào từng phần nhỏ:

1. **Cholesky1:**

****

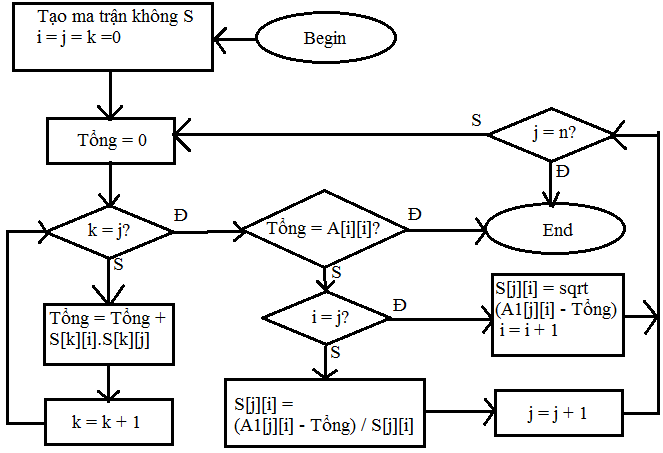
Ở bước này, ta sử dụng tối đa n3 phép nhân và n3 phép cộng.

1. **Cholesky2:**

****

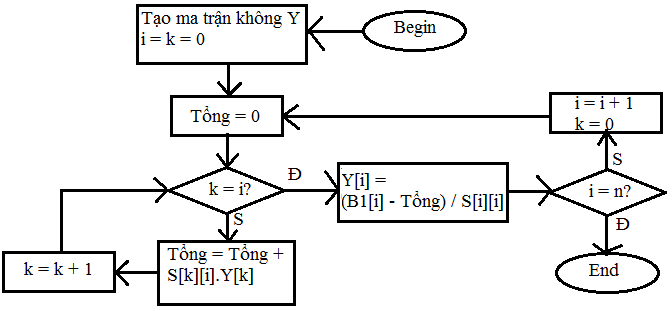
Ở bước này, ta sử dụng tối đa n2 phép nhân và n2 phép cộng.

1. **Cholesky:**

****

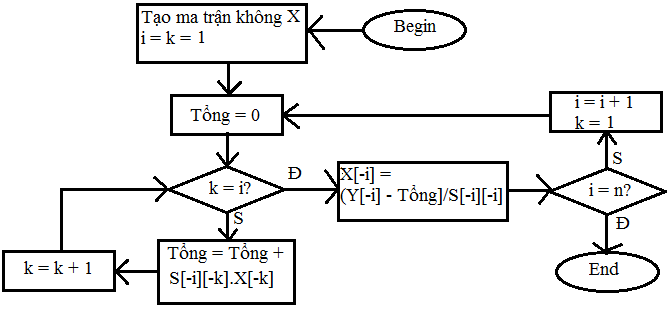
Ở bước này, ta sử dụng tối đa (n3+6n2+11n)/6 phép nhân; (n3+6n2+17n)/6 phép cộng.

1. **Cholesky3:**

****

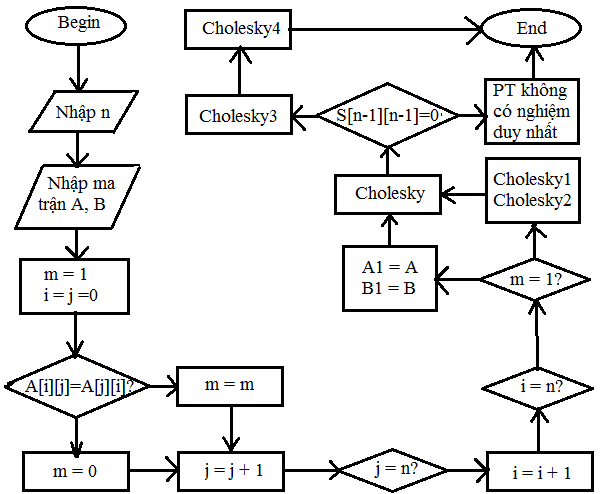
Ở bước này, ta sử dụng tối đa (n2+n)/2 phép nhân và (n2+n)/2 phép cộng.

1. **Cholesky4:**

****

Ở bước này, ta sử dụng tối đa (n2+n)/2 phép nhân và (n2+n)/2 phép cộng.

1. **Thuật toán cuối cùng:**

****

Ở bước này, chúng ta không sử dụng thêm các phép tính, mà chỉ sử dụng các hàm được tạo ra sẵn. Vì vậy, cả thuật toán sử dụng tất cả (7n3+18n2+17n)/6 phép nhân và (7n3+18n2+23n)/6 phép cộng.

Ta có bảng sau đây:

| n | Số phép nhân | Số phép cộng |
| --- | --- | --- |
| 3 | 67 | 70 |
| 5 | 235 | 240 |
| 10 | 1495 | 1505 |
| 100 | 1196950 | 1197050 |
| 1000 | 1169669500 | 1169670500 |

## Chương trình

**from** cmath **import** sqrt  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"  
def** cholesky1():  
 A1=[[0]\*n **for** i **in** range(n)]  
 **for** i **in** range(n):  
 **for** j **in** range(n):  
 A1[i][j]=sum(A[k][i]\*A[k][j] **for** k **in** range(n))  
 **return** A1  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"  
def** cholesky2():  
 B1=[0]\*n  
 **for** i **in** range(n):  
 B1[i]=sum(A[k][i]\*B[k] **for** k **in** range(n))  
 **return** B1  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"  
def** cholesky():  
 S=[[0]\*n **for** i **in** range(n)]  
 **for** i **in** range(n):  
 **for** j **in** range(i+1):  
 tổng = sum(S[k][i] \* S[k][j] **for** k **in** range(j))  
 **if** abs(tổng- A1[i][i])<0.0000000001:  
 **break  
 if** i==j:  
 S[j][i] = sqrt(A1[i][i] - tổng)  
 **else**:  
 S[j][i] = (A1[j][i] - tổng)/S[j][j]  
 **if** S[i][i]==0:  
 **break  
 return** S  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"  
def** cholesky3():  
 Y=[0]\*n  
 **for** i **in** range(n):  
 Y[i]=(B1[i]-sum(S[k][i]\*Y[k] **for** k **in** range(i)))/S[i][i]  
 **return** Y  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"  
def** cholesky4():  
 X=[0]\*n  
 **for** i **in** range(1, n+1):  
 X[-i]=(Y[-i]-sum(S[-i][-k]\*X[-k] **for** k **in** range(1, i)))/S[-i][-i]  
 **return** X  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"**file = open(**'sample\_array.txt'**)  
matrix = [line.split() **for** line **in** file]  
n=len(matrix)  
A=[[0]\*n **for** i **in** range(n)]  
**for** i **in** range(n):  
 **for** j **in** range(n):  
 A[i][j]=float(matrix[i][j])  
print(**'A= '**+str(A))  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"**B=[0]\*n  
**for** i **in** range(n):  
 B[i]=float(matrix[i][n])  
print(**'B= '**+str(B))  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"**print(**''**)  
print(**'Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B'**)  
**"''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''"**print(**''**)  
print(**'Tìm ma trận A1 đối xứng:'**)  
m=1  
**for** i **in** range(n):  
 **for** j **in** range(n):  
 **if** A[i][j]==A[j][i]:  
 m\*=1  
 **else**:  
 m\*=0  
**if** m==1:  
 print(**'A là ma trận vuông đối xứng có thể khai triển theo Cholesky'**)  
 A1=A  
 B1=B  
**else**:  
 print(**'A không đối xứng'**)  
 print(**'Để giải được theo Cholesky ta cần nhân cả 2 vế với A^t:'**)  
 A1=cholesky1()  
 B1=cholesky2()  
print(**''**)  
print(**'Khi đó phương trình trở thành A1\*X=B1, với: '**)  
print(**'A1= '**+str(A1))  
print(**'và B1= '**+str(B1))  
S=cholesky()  
**if** S[n-1][n-1]==0:  
 print(**'Ma trận A1 có định thức bằng 0 nên không thể khai triển Cholesky'**)  
 print(**'Suy ra phương trình AX=B không có nghiệm duy nhất'**)  
**else**:  
 print(**''**)  
 print(**'Phân tích A1 theo Cholesky: A1=S^t\*S'**)  
 print(**'S= '**+str(S))  
 print(**''**)  
 print(**'Khi đó phương trình trở thành: S^t\*S\*X=B1'**)  
 print(**'Đặt Y=S\*X có: S^t\*Y=B1'**)  
 print(**'Giải phương trình này ta được: '**)  
 Y=cholesky3()  
 print(**'Y= '**+str(Y))  
 print(**''**)  
 print(**'Giải phương trình SX=Y ta được nghiệm của phương trình:'**)  
 X=cholesky4()  
 print(**'X= '**+str(X))

## Ví dụ test

**Ví dụ 1:** ma trận đối xứng suy biến

1 2 3 4 5 1  
2 4 6 8 10 2  
3 6 26 18 24 3  
4 8 18 70 52 4  
5 10 24 52 82 5

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/pythonProject/cholesky.py

A= [[1.0, 2.0, 3.0, 4.0, 5.0], [2.0, 4.0, 6.0, 8.0, 10.0], [3.0, 6.0, 26.0, 18.0, 24.0], [4.0, 8.0, 18.0, 70.0, 52.0], [5.0, 10.0, 24.0, 52.0, 82.0]]

B= [1.0, 2.0, 3.0, 4.0, 5.0]

Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B

Tìm ma trận A1 đối xứng:

A là ma trận vuông đối xứng có thể khai triển theo Cholesky

Khi đó phương trình trở thành A1\*X=B1, với:

A1= [[1.0, 2.0, 3.0, 4.0, 5.0], [2.0, 4.0, 6.0, 8.0, 10.0], [3.0, 6.0, 26.0, 18.0, 24.0], [4.0, 8.0, 18.0, 70.0, 52.0], [5.0, 10.0, 24.0, 52.0, 82.0]]

và B1= [1.0, 2.0, 3.0, 4.0, 5.0]

Ma trận A1 có định thức bằng 0 nên không thể khai triển Cholesky

Suy ra phương trình AX=B không có nghiệm duy nhất

Process finished with exit code 0

**Ví dụ 2:** Ma trận đối xứng khả nghịch

1 3 -2 0 -2 1  
3 4 -5 1 -3 2  
-2 -5 3 -2 2 3  
0 1 -2 5 3 4  
-2 -3 2 3 4 5

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/pythonProject/cholesky.py

A= [[1.0, 3.0, -2.0, 0.0, -2.0], [3.0, 4.0, -5.0, 1.0, -3.0], [-2.0, -5.0, 3.0, -2.0, 2.0], [0.0, 1.0, -2.0, 5.0, 3.0], [-2.0, -3.0, 2.0, 3.0, 4.0]]

B= [1.0, 2.0, 3.0, 4.0, 5.0]

Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B

Tìm ma trận A1 đối xứng:

A là ma trận vuông đối xứng có thể khai triển theo Cholesky

Khi đó phương trình trở thành A1\*X=B1, với:

A1= [[1.0, 3.0, -2.0, 0.0, -2.0], [3.0, 4.0, -5.0, 1.0, -3.0], [-2.0, -5.0, 3.0, -2.0, 2.0], [0.0, 1.0, -2.0, 5.0, 3.0], [-2.0, -3.0, 2.0, 3.0, 4.0]]

và B1= [1.0, 2.0, 3.0, 4.0, 5.0]

Phân tích A1 theo Cholesky: A1=S^t\*S

S= [[(1+0j), (3+0j), (-2+0j), 0j, (-2+0j)], [0, 2.23606797749979j, -0.4472135954999579j, -0.4472135954999579j, -1.3416407864998738j], [0, 0, 0.8944271909999157j, 2.012461179749811j, 1.565247584249853j], [0, 0, 0, (3.04138126514911+0j), (2.2193863286223237+0j)], [0, 0, 0, 0, 0.8219949365267873j]]

Khi đó phương trình trở thành: S^t\*S\*X=B1

Đặt Y=S\*X có: S^t\*Y=B1

Giải phương trình này ta được:

Y= [(1+0j), 0.4472135954999579j, -5.366563145999496j, (-2.301585822275003+0j), -3.781176708023214j]

Giải phương trình SX=Y ta được nghiệm của phương trình:

X= [(-7.399999999999999+0j), (-2.7999999999999963-0j), (-3.800000000000003-0j), (2.5999999999999934+0j), (-4.599999999999991-0j)]

Process finished with exit code 0

**Ví dụ 3:** Ma trận không đối xứng khả nghịch

1 8 4 1 2 4 1  
9 8 2 1 4 1 2  
8 5 3 2 5 2 3  
5 2 8 2 1 3 4  
7 5 3 2 4 1 5  
9 5 3 1 9 1 6

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/pythonProject/cholesky.py

A= [[1.0, 8.0, 4.0, 1.0, 2.0, 4.0], [9.0, 8.0, 2.0, 1.0, 4.0, 1.0], [8.0, 5.0, 3.0, 2.0, 5.0, 2.0], [5.0, 2.0, 8.0, 2.0, 1.0, 3.0], [7.0, 5.0, 3.0, 2.0, 4.0, 1.0], [9.0, 5.0, 3.0, 1.0, 9.0, 1.0]]

B= [1.0, 2.0, 3.0, 4.0, 5.0, 6.0]

Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B

Tìm ma trận A1 đối xứng:

A không đối xứng

Để giải được theo Cholesky ta cần nhân cả 2 vế với A^t:

Khi đó phương trình trở thành A1\*X=B1, với:

A1= [[301.0, 210.0, 134.0, 59.0, 192.0, 60.0], [210.0, 207.0, 109.0, 45.0, 140.0, 66.0], [134.0, 109.0, 111.0, 37.0, 78.0, 54.0], [59.0, 45.0, 37.0, 15.0, 35.0, 18.0], [192.0, 140.0, 78.0, 35.0, 143.0, 38.0], [60.0, 66.0, 54.0, 18.0, 38.0, 32.0]]

và B1= [152.0, 102.0, 82.0, 33.0, 103.0, 35.0]

Phân tích A1 theo Cholesky: A1=S^t\*S

S= [[(17.349351572897472+0j), (12.104198771788935+0j), (7.723631597236749+0j), (3.4007034644549865+0j), (11.066696019921311+0j), (3.45834250622541+0j)], [0, (7.777427086962836+0j), (1.9944420865068082+0j), (0.49337772754666154+0j), (0.7774436918917089+0j), (3.103794431475366+0j)], [0, 0, (6.882420774243353+0j), (1.416682630220435+0j), (-1.3114062292229935+0j), (3.0655926155822875+0j)], [0, 0, 0, (1.0884873403197919+0j), (-1.0659549522860388+0j), (0.33522466554305647+0j)], [0, 0, 0, 0, (4.131316285428605+0j), (0.40958398654160505+0j)], [0, 0, 0, 0, 0, (0.85342517753148+0j)]]

Khi đó phương trình trở thành: S^t\*S\*X=B1

Đặt Y=S\*X có: S^t\*Y=B1

Giải phương trình này ta được:

Y= [(8.761134349104372+0j), (-0.5202892399583007+0j), (2.2332131636692147+0j), (0.27462413176542566+0j), (2.3404336283160054+0j), (-1.8524402843388406+0j)]

Giải phương trình SX=Y ta được nghiệm của phương trình:

X= [(-0.6111111111111218+0j), (0.3338945005611762+0j), (1.0931537598204382+0j), (1.6863075196408708+0j), (0.7817059483726186+0j), (-2.1705948372615365+0j)]

Process finished with exit code 0

**Ví dụ 4:** Ma trận không đối xứng suy biến vuông

50 107 36 1  
81 173 57 2  
31 66 21 3

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/pythonProject/cholesky.py

A= [[50.0, 107.0, 36.0], [81.0, 173.0, 57.0], [31.0, 66.0, 21.0]]

B= [1.0, 2.0, 3.0]

Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B

Tìm ma trận A1 đối xứng:

A không đối xứng

Để giải được theo Cholesky ta cần nhân cả 2 vế với A^t:

Khi đó phương trình trở thành A1\*X=B1, với:

A1= [[10022.0, 21409.0, 7068.0], [21409.0, 45734.0, 15099.0], [7068.0, 15099.0, 4986.0]]

và B1= [305.0, 651.0, 213.0]

Ma trận A1 có định thức bằng 0 nên không thể khai triển Cholesky

Suy ra phương trình AX=B không có nghiệm duy nhất

Process finished with exit code 0

**Ví dụ 5:** Ma trận đối xứng suy biến không vuông

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

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/pythonProject/cholesky.py

A= [[1.0, 2.0, 3.0, 4.0, 5.0], [0.0, 0.0, 1.0, 2.0, 1.0], [0.0, 0.0, 4.0, 1.0, 2.0], [0.0, 0.0, 0.0, 7.0, 4.0], [0.0, 0.0, 0.0, 0.0, 6.0]]

B= [1.0, 2.0, 3.0, 4.0, 5.0]

Giải hệ phương trình tuyến tính bằng phương pháp Cholesky: AX=B

Tìm ma trận A1 đối xứng:

A không đối xứng

Để giải được theo Cholesky ta cần nhân cả 2 vế với A^t:

Khi đó phương trình trở thành A1\*X=B1, với:

A1= [[1.0, 2.0, 3.0, 4.0, 5.0], [2.0, 4.0, 6.0, 8.0, 10.0], [3.0, 6.0, 26.0, 18.0, 24.0], [4.0, 8.0, 18.0, 70.0, 52.0], [5.0, 10.0, 24.0, 52.0, 82.0]]

và B1= [1.0, 2.0, 17.0, 39.0, 59.0]

Ma trận A1 có định thức bằng 0 nên không thể khai triển Cholesky

Suy ra phương trình AX=B không có nghiệm duy nhất

Process finished with exit code 0

# Phương pháp Jacobi giải phương trình Ax=B

## Thuật toán

1. **Gói A: Kiểm tra tính chéo trội**

Input: ma trận A

Output: kt(A)

B1: m = [] = sum(abs(aij) for j =1 to n

B2:

if (2abs(aii) > m[i] for all i ∈ (1,n)):

return kt(A)= 0

else:

m = sum(abs(aij)) for i = 1 to n

for all i ∈ (1, n):

if 2abs(ajj) > m[j]):

return kt(A) = 1

else: return kt(A) = -1

1. **Gói B: Tính chuẩn**

Input: kt(A), B, x

Output: chuan(B), chuan(x)

if kt(A)=0:

chuan(B) := chuan\_vc(B)

chuan(v) := chuan\_vc(v)

if kt(A)=1:

chuan(B) := chuan\_1(B)

chuan(v) := chuan\_1(v)

1. **Gói C: Xác định ma trận B và vecto d**

Input: A, b

Output: B, d

**Bước 1**: Xác định B

for i=1 to n

for j=1 to n

if i==j: bij:=0

else: bij:=-(aij)/(aii)

**Bước 2**: Xác định d

for i=1 to n: d[i] := b[i]/aii

1. **Gói D: Gói lặp**

Input: B, d, e

Output: x\*

**Bước 1**: x0:=d;

**Bước 2**: xây dựng hàm phi(x) : = Bx+d

**Bước 3**: xk:=phi(x0); x(k-1) :=x0

**Bước 4**: thực hiện vòng lặp While ||xk-x(k-1)||>e do:

x(k-1)=xk ;

x(k) = phi(x(k-1))

**Bước 5**: kết luận xk là nghiệm của hệ phương trình

1. **Main:**

Input: A, b, e

Output: x\*

Kiểm tra A

if kt(A) == -1: kết luận: ma trận A không chéo trội

else

if kt(A) == 0:

sử dụng gói C, tính B, d

eps:=e(1-chuan(B))/chuan(B)

if kt(A) == 1:

tính chuanB1, chuanT, chuanD

sử dụng gói C tính B, d

eps := e\*(1-chuanB1)/( chuanB1\*chuanT\*chuanD)

sử dụng gói lặp tính lapdon(B, d, eps).

## Chương trình

*# Nhập dữ liệu từ file input  
# file input là ma trận cấp n x (n+1)  
# trong đó n cột đầu biểu diễn ma trận A, cột cuối cùng biểu diễn vecto b*f1 = open(**"sample\_array.txt"**, **"r"**)  
data = f1.readlines()  
n = sum(1 **for** \_ **in** data)  
f1.close()  
A = []  
b = []  
**for** s **in** data:  
 row = s.strip().split(**' '**)  
 row1 = [float(i) **for** i **in** row]  
 row2 = []  
 **for** i **in** range(len(row1)-1):  
 row2.append(row1[i])  
 A.append(row2)  
 b.append(row1[len(row1)-1])  
print(**"A = "**, A)  
print(**"b = "**, b)  
*# sai số e*eps = e = 1e-6  
  
  
*# Gói A  
# kiểm tra điều kiện chéo trội của A***def** kt(A):  
 m = []  
 m = [sum(abs(j) **for** j **in** r) **for** r **in** A]  
 **if** all(2 \* abs(A[i][i]) > m[i] **for** i **in** range(n)):  
 **return** 0  
 **else**:  
 m = [sum([abs(x[i]) **for** x **in** A]) **for** i **in** range(len(A[0]))]  
 **if** all(2 \* abs(A[j][j]) > m[j] **for** j **in** range(n)):  
 **return** 1  
 **else**:  
 **return** -1  
  
  
*# chuẩn vc của ma trận***def** chuanvc(a):  
 a1 = []  
 **for** r **in** a:  
 m = (sum(abs(i) **for** i **in** r))  
 a1.append(m)  
 c = max(i **for** i **in** a1)  
 **return** c  
  
  
*# chuẩn 1 của ma trận***def** chuan1(a):  
 a1 = [sum([abs(x[i]) **for** x **in** a]) **for** i **in** range(len(a[0]))]  
 m = max(abs(i) **for** i **in** a1)  
 **return** m  
  
  
*# xác định loại chuẩn ma trận***def** chuan(a):  
 **if** kt(A) == 0:  
 **return** chuanvc(a)  
 **else**:  
 **if** kt(A) == 1:  
 **return** chuan1(a)  
 **else**:  
 print(**"error!"**)  
  
  
*# xác định loại chuẩn vecto***def** chuan\_vecto(v):  
 **if** kt(A) == 0:  
 **return** sum(abs(i) **for** i **in** v)  
 **if** kt(A) == 1:  
 **return** max(abs(i) **for** i **in** v)  
 **if** kt(A) == -1:  
 print(**"error!"**)  
  
  
*# Hiệu hai vecto***def** sub(x, y):  
 m = []  
 **for** i **in** range(len(x)-2):  
 m.append((x[i] - y[i]))  
 **return** m  
  
  
*# Gói C:  
# xác định ma trận B và vecto tự do*B = []  
d = []  
  
  
**def** xacdinh\_b():  
 **for** i **in** range(n):  
 B.append([])  
 **for** j **in** range(n):  
 **if** i == j:  
 B[i].append(0)  
 **else**:  
 B[i].append(-(A[i][j]) / (A[i][i]))  
  
  
**def** xacdinh\_d():  
 **for** i **in** range(n):  
 d.append(b[i] / A[i][i])  
  
  
*# lặp đơn***def** lap\_don(B, d, e):  
 x0 = d  
  
 *# xây dựng hàm phi(x)* **def** phi(B, x, d):  
 m = []  
 y = []  
 **for** r **in** B:  
 p = sum(x[i] \* r[i] **for** i **in** range(n))  
 m.append(p)  
 **for** i **in** range(n):  
 y.append(m[i] + d[i])  
 **return** y  
  
 xk = phi(B, x0, d)  
 dem = 0  
 **while** chuan\_vecto(sub(x0, xk)) > e:  
 x0 = xk  
 xk = phi(B, x0, d)  
 dem = dem + 1  
 print(**"Số bước lặp: "**, dem)  
 print(**"Nghiem của hệ phương trình là: "**, xk)  
 **return** xk  
  
  
**def** kiemtra(sol):  
 m = []  
 t = []  
 **for** r **in** A:  
 p = sum(sol[i] \* r[i] **for** i **in** range(n))  
 m.append(p)  
 **for** i **in** range(n):  
 t.append(m[i] - b[i])  
 **return** t  
  
  
*# main  
# Bước 1: Kiểm tra A***if** kt(A) == -1:  
 print(**"Ma trận A không phải ma trận chéo trội."**)  
**else**:  
 **if** kt(A) == 0:  
 print(**"A là ma trận chéo trội hàng"**)  
 xacdinh\_d()  
 xacdinh\_b()  
 eps = e\*(1-chuan(B))/(chuan(B))  
 print(**"chuan(B) = "**, chuan(B))  
 **if** kt(A) == 1:  
 print(**"A là ma trận chéo trội cột"**)  
 xacdinh\_d()  
 xacdinh\_b()  
 chuanT = max(abs(1 / A[i][i]) **for** i **in** range(n))  
 chuanD = max(abs(A[i][i]) **for** i **in** range(n))  
 chuan\_B1 = max([sum([abs(x[j] / A[j][j]) **for** x **in** A]) **for** j **in** range(n)]) - 1  
 eps = e\*(1-chuan\_B1)/(chuan\_B1 \* chuanT \* chuanD)  
 print(**"chuan(B1) = "**, chuan\_B1)  
 *# Bước 2: lặp đơn* nghiem = lap\_don(B, d, eps)  
  
 *# Kiểm tra tính đúng đắn của thuật toán* print(**"kiểm tra tính đúng đắn của thuật toán: "**)  
 print(**"Ax\* - b = "**, kiemtra(nghiem))

## Ví dụ test:

**Ví dụ 1:** không chéo trội hàng, không chéo trội cột

1 2 3 8 1  
2 4 0 6 2  
1 2 7 1 3  
3 5 0 0 4

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/Admin/PycharmProjects/pythonProject/Chủ đề 11.2\_PP Jacobi.py"

A = [[1.0, 2.0, 3.0, 8.0], [2.0, 4.0, 0.0, 6.0], [1.0, 2.0, 7.0, 1.0], [3.0, 5.0, 0.0, 0.0]]

b = [1.0, 2.0, 3.0, 4.0]

Ma trận A không phải ma trận chéo trội.

Process finished with exit code 0

**Ví dụ 2:** chéo trội hàng

15 1 4 -8 1  
2 16 9 3 2  
1 -9 12 1 3  
1 0 5 23 4

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/Admin/PycharmProjects/pythonProject/Chủ đề 11.2\_PP Jacobi.py"

A = [[15.0, 1.0, 4.0, -8.0], [2.0, 16.0, 9.0, 3.0], [1.0, -9.0, 12.0, 1.0], [1.0, 0.0, 5.0, 23.0]]

b = [1.0, 2.0, 3.0, 4.0]

A là ma trận chéo trội hàng

chuan(B) = 0.9166666666666667

Số bước lặp: 31

Nghiem của hệ phương trình là: [0.07816501314527526, -0.027580469482353598, 0.2124398992169983, 0.12433191519129433]

kiểm tra tính đúng đắn của thuật toán:

Ax\* - b = [-9.969655863084625e-07, -2.6469002394335917e-06, -5.571826822148296e-08, -1.441369963828265e-06]

Process finished with exit code 0

**Ví dụ 3:** chéo trội cột

3 3.001 0 1  
2 7.1 0.0001 2  
0 4 0.001 3

Kết quả:

C:\Users\Admin\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/Admin/PycharmProjects/pythonProject/Chủ đề 11.2\_PP Jacobi.py"

A = [[3.0, 3.001, 0.0], [2.0, 7.1, 0.0001], [0.0, 4.0, 0.001]]

b = [1.0, 2.0, 3.0]

A là ma trận chéo trội cột

chuan(B1) = 0.986056338028169

Số bước lặp: 48

Nghiem của hệ phương trình là: [0.11337069087783047, 0.21988934600615506, 2120.4426159726645]

kiểm tra tính đúng đắn của thuật toán:

Ax\* - b = [-2.0372592501871623e-12, -3.3717473257866004e-12, -2.715161429023283e-12]

Process finished with exit code 0