PHÂN RÃ QR GIẢI HỆ PHƯƠNG TRÌNH TUYẾN TÍNH VÀ MÔ HÌNH INPUT-OUTPUT CỦA LEONTIEF TRONG PHÂN TÍCH KINH TẾ

QR DIFFERENCE SOLUTIONS SYSTEM OF LINEAR EQUATIONS AND MODELS LEONTIEF'S INPUT-OUTPUT IN ECONOMIC ANALYSIS

NGUYỄN VĂN LỘC(*)

TÓM TẮT: *Bài viết trình bày cách sử dụng QR decomposition giải các dạng hệ phương trình tuyến tính tổng quát và mô hình Input-Output của Leontief trong phân tích kinh tế.*
Từ khóa: *phân rã ma trận QR; mô hình Input-Output.*

ABSTRACT: *This paper presents how to use QR decomposition to solve general linear systems of equations and Leontief's Input-Output model in economic analysis.*
Key words: *QR matrix decomposition; Input-Output model.*

1. ĐẶT VẤN ĐỀ

Khi xử lý thông tin về các mô hình kinh tế của các quốc gia biểu diễn dưới dạng hệ phương trình tuyến tính với hàng trăm ẩn có ma trận hệ số cấp $m\times n, m\neq n$, sử dụng phương pháp Gauss để giải không khả thi. Để giải quyết khó khăn này, một trong các phương pháp là sử dụng phân rã QR, tách ma trận hệ số thành các phần đơn giản hơn kết hợp với phương pháp lập trình.

2. NỘI DUNG

2.1. QR decomposition bằng phương pháp lập trình

Phân rã QR dành cho ma trận $m\times n$ (không giới hạn ở ma trận vuông), nó phân tách ma trận thành các thành phần Q và R.

Công thức: A = Q.R hoặc A = QR

Với A là ma trận muốn phân rã còn: thừa số $Q\in M_{m\times m}$ (Q-factor): gồm các cột trực chuẩn (Orthonormal Columns) có kích thước $m\times m$: $Q^T.Q = I_n$; thừa số $R\in M_{m\times n}$ (R-factor): là ma trận tam giác trên kích thước $m\times n$, $R_{ij}\neq 0$, khả nghịch.

Ví dụ: Phân rã QR ma trận

$$A=\begin{bmatrix} 2 & 4 & -1 & 5 & -2 \\ -4 & -5 & 3 & -8 & 1 \\ 2 & -5 & -4 & 1 & 8 \\ -6 & 0 & 7 & -3 & 1 \end{bmatrix}.$$ [5, tr.128]

Giải: Sử dụng gói numpy và hàm qr()

```
Entrée [1 ] In[1]:   from numpy import array
                     from scipy.linalg import qr
Entrée [2 ]          A = array([[2, 4, -1, 5, -2], [-4, -5, 3, -
In[2]:               8, 1], [2, -5, -4, 1, 8], [-6, 0, 7, -3, 1]])
Entrée [3 ] :In[3]:  #fatorize
                     Q, R = qr( A )
Entrée [ 4 ] :In[4]: print(Q)
   Out[ 4]:          [[-0.25819889  0.43675975  0.86161696
                     -0.01380096]
                      [ 0.51639778 -0.48814325  0.39284139
                     -0.58371667]
                      [-0.25819889 -0.71936899  0.29645219
                     0.5726759 ]
                      [ 0.77459667  0.23122575  0.12412879
                     0.57543609]]
Entrée [5]:          print(R)
In[5]:
```

(*) PGS.TS. Trường Đại học Văn Lang, *loc.nv@vlu.edu.vn, Mã số: TCKH31-05-2022*

```
Out[5 ]:    [[-7.74596669e+00 -2.32379001e+00
            8.26236447e+00 -8.00416558e+00
            -2.58198890e-01]
            [ 0.00000000e+00  7.78460018e+00
            2.59486673e+00  4.67589846e+00
            -6.88538894e+00]
            [ 0.00000000e+00  0.00000000e+00
            -2.59896547e-15  1.08941954e+00
            1.16535373e+00]
            [ 0.00000000e+00  0.00000000e+00
            0.00000000e+00  3.44709617e+00
            4.60072855e+00]]
```

2.2. QR decomposition giải hệ phương trình tuyến tính

Phương pháp:

$$AX = B(1) \Leftrightarrow QRX = B$$

$$\Leftrightarrow \begin{cases} QY = B(2) \Leftrightarrow Y = Q^{-1}.B = Q^{T}.B \\ RX = Y(3) \Leftrightarrow X = R^{+}Y \end{cases}$$

Với: $Q_{m\times m}; B_{m\times 1}; R_{m\times n}; R^{+}_{n\times m};$
$Q^{-1} = Q^{T}; Q^{T}.Q = Q^{-1}.Q = I$

Thay vì giải hệ phương trình (1), ta lần lượt:

Bước 1: Giải hệ phương trình (2), tìm $Y = Q^{-1}.B = Q^{T}.B$

Bước 2: Giải hệ phương trình (3), tìm X, với $R^{+}_{n\times m}$ là ma trận giả đảo.

2.2.1. QR decomposition giải hệ phương trình tuyến tính Cramer

Ví dụ 1: Giải hệ phương trình:

$$\begin{cases} 2x + 3y + 11z + 5t = 2 \\ x + y + 5z + 2t = 1 \\ 2x + y + 3z + 2t = -3 \\ x + y + 3z + 4t = -3 \end{cases} \text{[4, tr.82]}$$

Giải: Sử dụng gói numpy và hàm qr()

```
In[1 ]:     from numpy import array
            from scipy.linalg import qr
In[2 ]:     A = array([[2, 3, 11, 5], [1, 1, 5, 2],
            [2, 1, 3, 2], [1, 1, 3, 4]])
In[3 ]:     Q, R = qr(A)
In[4 ]:     print(Q)
In[5 ]:     print(R)
In[ 6]:     C = Q.T
In[7 ]:     import numpy as np
            b = np.array([[2],[1],[-3], [-3]])
In[ 8]:     Y = C.dot(b)
In[9 ]:     from numpy.linalg import pinv
            K = pinv(R)
In[10 ]:    X = K.dot(Y)
            X
Out[10]:    array([[-2.00000000e+00],
            [-7.10542736e-15],
            [ 1.00000000e+00],
            [-1.00000000e+00]])
```

Nghiệm của hệ phương trình là: {x =-2.00000000e+00, y=[-7.10542736e-15, z=1.00000000e+00,t=-.00000000e+00}

Ví dụ 2: Giải hệ phương trình:

$$\begin{cases} 2x + y + 4z + 8t = -1 \\ 4x + 3y - 6z + 2t = 3 \\ 3x - 2y + 2z - 2t = 8 \\ 2x - y + 2z = 4 \end{cases} \text{[4, tr.82]}$$

Đáp số: Nghiệm của hệ phương trình là:{x = 2, y=-3, z= -3/2, t =1/2}

2.2.2. QR decomposition giải hệ phương trình tuyến tính có số phương trình lớn hơn số ẩn

Ví dụ 1: Giải hệ phương trình:

$$\begin{cases} x + 2y + 3z + t = 3 \\ x + 4y = 5z + 2t = 2 \\ 2x + 9y + 8z + 3t = 7 \\ 3x + 7y + 7z + 2t = 12 \\ 5x + 7y + 9z + 2t = 20 \end{cases} \text{[4, tr.101]}$$

Giải: Sử dụng gói numpy và hàm qr()

```
Entrée [1]:     from numpy import array
In[1]:          from scipy.linalg import qr
Entrée [2 ]:    A = array([[1, 2, 3, 1], [1, 4, 5, 2],
In[2]:          [2, 9, 8, 3], [3, 7, 7, 2], [5, 7, 9, 2]])
Entrée [3 ]:    #fatorize
In[3]:          Q, R = qr( A )
Entrée [4 ]:    print(Q)
In[4]:
Entrée [5]:     print(R)
In[5]:
Entrée [6 ]:    K =Q.dot(R)
In[6]:
Entrée [7 ]:    C = Q.T
In[7]:
Entrée [ 8]:    import numpy as np
In[8]:          b = np.array([[3], [2], [7], [12], [20]])
```

```
Entrée [ 9]:    Y = C.dot(b)
In[9]:
Entrée [10 ]:   from numpy.linalg import pinv
In[10]:         E = pinv(R)
Entrée [11 ]:   X = E.dot(Y)
In[11]:         print(X)
Out[11]:        [[ 4.11111111]
                 [ 0.66666667]
                 [-0.11111111]
                 [-2.11111111]]
```

Nghiệm của hệ phương trình là:

{x = 4.11111111, y = 0.66666667, z=-0.11111111, t= -2.11111111}

Ví dụ 2: Giải hệ phương trình:

$$\begin{cases} 8x+6y+5z+2t=21 \\ 3x+3y+2z+t=10 \\ 4x+2y+3z+t=8 \\ 3x+5y+z+t=15 \\ 7x+4y+5z+2t=18 \end{cases}$$ [4, tr.100]

Đáp số: Nghiệm của hệ phương trình là:

{x=3.00000000e+00, y=-1.33226763e-15, z=-5.00000000e+00, t=1.10000000e+01}

2.2.3. QR decomposition giải hệ phương trình tuyến tính có số phương trình ít hơn số ẩn

Ví dụ 1: Giải hệ phương trình:

$$\begin{cases} x+2y+3z-2t+m=4 \\ 3x+6y+5z-4t+3m=5 \\ x+2y+7z-4t+m=11 \\ 2x+4y+2z-3t+3m=6 \end{cases}.$$ [4, tr.100]

Giải: Sử dụng gói numpy và hàm qr()

```
Entrée [1 ] :   from numpy import array
In[1]:          from scipy.linalg import qr
Entrée [2 ] :   A = array([[1, 2, 3, -2, 1], [3, 6, 5, -
In[2]:          4, 3], [1, 2, 7, -4, 1], [2, 4, 2, -3, 3]])
Entrée [ 3] :   #fatorize
In[3]:          Q, R = qr( A )
Entrée [4 ] :   print(Q)
In[4]:
Entrée [ 5] :   print(R)
In[5]:
Entrée [6 ] :   K =Q.dot(R)
In[6]:
Entrée [ 7] :   C = Q.T
In[7]:
Entrée [ 8] :   import numpy as np
In[8]:          b = np.array([[4], [5], [11], [6]])
Entrée [9 ] :   Y = C.dot(b)
In[9]:
Entrée [10 ] :  from numpy.linalg import pinv
In [10]:        E = pinv(R)
Entrée[ 11] :   X = E.dot(Y)
In[11]:         print(X)
Out[11]:        [[-0.9673913 ]
                 [-1.93478261]
                 [ 0.33695652]
                 [-2.82608696]
                 [ 2.17391304]]
```

Nghiệm của hệ phương trình là: {x =-0.9673913, y=-1.93478261, z=0.33695652, t=-2.82608696, m = 2.17391304}

Ví dụ 2: Giải hệ phương trình:

$$\begin{cases} 2x+7y+3z+t=6 \\ 3x+5y+2z+2t=4 \\ 9x+4y+z+7t=2 \end{cases}.$$ [4, tr.99]

Đáp số: Nghiệm của hệ phương trình là: {x=([[-0.05714286, y = 0.74285714, z= 0.34285714, t=[-0.11428571}

2.2.4. QR decomposition giải hệ phương trình tuyến tính có số phương trình bằng số ẩn nhưng định thức ma trận hệ số bằng 0

Ví dụ 1: Giải hệ phương trình:

$$\begin{cases} 2x+5y+z+3t=2 \\ 4x+6y+3z+5t=4 \\ 4x+14y+z+7t=4 \\ 2x-3y+3z+2t=7 \end{cases}.$$ [4, tr.102]

Giải: Sử dụng gói numpy và hàm qr()

```
In[1 ]:    from numpy import array
           from scipy.linalg import qr
In[ 2]:    A = array([[ 2, 5, 1, 3], [4, 6, 3, 5],
           [4,14, 1, 7], [2, -3, 3, 2]])
In[3 ]:    #fatorize
           Q, R = qr( A )
In[4 ]:    print(Q)
In[5 ]:    print(R)
In[ 6]:    K =Q.dot(R)
In[ 7]:    C = Q.T
In[8 ]:    import numpy as np
```

```
           b = np.array([[2],[4],[4],[7]])
In[9 ]:    array([[2],[4],[4],[7]])
In[10 ]:   Y = C.dot(b)
In[11 ]:   from numpy.linalg import pinv
           E = pinv(R)
In[12 ]:   X = E.dot(Y)
           print(X)
Out[12]:   [[-1.69127517]
            [-1.62416107]
            [-1.4966443 ]
            [ 5.        ]]
```

Nghiệm của hệ phương trình là: {x= -1.69127517, y=-1.62416107, z=-1.4966443,t=5}

Ví dụ 2: Giải hệ phương trình:

$$\begin{cases} 3x+2y+5z+4t=3 \\ 2x+3y+6z+8t=5 \\ x-6y-9z-20t=-11 \\ 4x+y+4z+t=2 \end{cases}$$. [4, tr.102]

Đáp số: Nghiệm của hệ phương trình là:{{x=0.8877551, y=-0.63265306, z=[-0.47959184, t= 1.}

2.3. QR decomposition giải mô hình kinh tế Input-Output của Leontief

2.3.1. Giới thiệu mô hình Input-Output

Xét một nền kinh tế có n ngành sản xuất, ngành 1,2,3,..,n. Biểu diễn lượng cầu của tất cả các loại hàng hóa ở dạng giá trị, tức là đo bằng tiền. Tổng cầu về sản phẩm hàng hóa của ngành i (i = 1, 2, …, n) được xác định bởi: $x_i = x_{i1} + x_{i2} + ... + x_{ik}, ...(i = 1, 2, ..., n) ...(*)$. Ở đây: x_{ik} là giá trị sản phẩm của ngành i mà ngành k cần sử dụng cho quá trình sản xuất của mình (giá trị cầu trung gian); b_i là giá trị sản phẩm của ngành i dành cho nhu cầu tiêu dùng và xuất khẩu (giá trị cầu cuối cùng). Ký hiệu a_{ik} là tỷ phần chi phí đầu vào của ngành k đối với sản phẩm của ngành i, tính bởi công thức: $a_{ik} = \frac{x_{ik}}{x_k}, ...(i, k = 1, 2, ..n)$ Chú ý: $0 \le a_{ik} < 1$, ở đây, giả thiết a_{ik} là cố định đối với mỗi ngành sản xuất i , (k = 1,2,3,..,n). Người ta còn gọi a_{ik} là hệ số chi phí đầu vào và ma trận $A = (a_{ik})_n$ được gọi là ma trận hệ số chi phí đầu vào (hay ma trận hệ số kỹ thuật).

Đặt: $X = \begin{bmatrix} x_1 & x_2 & ... & x_n \end{bmatrix}^T ; b = \begin{bmatrix} b_1 & b_2 & ... & b_n \end{bmatrix}^T$.

Ta gọi X là ma trận tổng cầu và b là ma trận cầu cuối cùng. Khi đó, từ đẳng thức (*), thay $x_{ik} = a_{ik}.x_k$ ta có:

$$\begin{cases} x_i = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n + b_i \\ \quad i = 1, 2, 3, ..., n \end{cases}$$

Hay biểu diễn dưới dạng ma trận:

$$\begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{bmatrix}$$

$$\Leftrightarrow X = AX + b...(**)$$

2.3.2. Phương pháp giải mô hình

Từ (**), ta có: (I - A)X = b, ở đây I là ma trận đơn vị cấp n, nếu (I-A) không suy biến thì: $X = (I - A)^{-1}.b...(***)$. Công thức (***) được gọi là công thức tính ma trận tổng cầu. Ma trận (I - A) được gọi là ma trận Leontief. Như vậy, nếu chúng ta biết ma trận hệ số kỹ thuật A và ma trận cầu cuối cùng thì sẽ xác định được giá trị tổng cầu của các ngành sản xuất [1, tr.81].

2.3.3. QR decomposition giải các bài toán ứng dụng mô hình Input-Output

Phương pháp:

Bước 1: Thiết lập mô hình Input-Output, chuyển dịch bài toán từ “ngôn ngữ kinh tế” sang “ngôn ngữ Toán học”.

Bước 2: Giải mô hình Input-Output bằng phương pháp lập trình.

Bước 3: Chuyển dịch kết quả bài toán từ kết quả “ngôn ngữ lập trình” sang “ngôn ngữ kinh tế”.

Ví dụ 1: Giả sử trong một nền kinh tế có 5 ngành sản xuất: ngành 1, ngành 2, ngành 3, ngành 4, ngành 5. Biết ma trận hệ số kỹ thuật là:

$$A = \begin{bmatrix} 0,3 & 0,1 & 0,2 & 0,1 & 0,2 \\ 0,2 & 0,3 & 0,2 & 0,1 & 0,2 \\ 0,1 & 0,4 & 0,3 & 0,1 & 0,2 \\ 0,2 & 0,2 & 0,1 & 0,1 & 0,2 \\ 0,1 & 0,1 & 0,1 & 0,1 & 0,1 \end{bmatrix}$$

Và mức cầu cuối cùng đối với sản phẩm của từng ngành thứ tự là 30, 40, 50, 70 và 90 (đơn vị tính: triệu dollar). Hãy xác định mức tổng cầu của mỗi ngành sản xuất.

Giải: Bước 1: Thiết lập mô hình Input-Output. Với A là ma trận kỹ thuật, I là ma trận đơn vị, X là ma trận tổng cầu, b là ma trận cầu cuối, ta có:

$$A = \begin{bmatrix} 0,3 & 0,1 & 0,2 & 0,1 & 0,2 \\ 0,2 & 0,3 & 0,2 & 0,1 & 0,2 \\ 0,1 & 0,4 & 0,3 & 0,1 & 0,2 \\ 0,2 & 0,2 & 0,1 & 0,1 & 0,2 \\ 0,1 & 0,1 & 0,1 & 0,1 & 0,1 \end{bmatrix};$$

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; b = \begin{bmatrix} 30 \\ 40 \\ 50 \\ 70 \\ 90 \end{bmatrix}; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

$$\Rightarrow X = \left[I - A\right]^{-1}.b$$

Bước 2: Phương pháp lập trình (sử dụng gói numpy và hàm qr ())

```
In[1 ]:    from numpy import array
           from scipy.linalg import qr
In[2 ]:    A = array([[0.3, 0.1, 0.2, 0.1, 0.2],
           [0.2, 0.3, 0.2, 0.1, 0.2], [0.1, 0.4, 0.3,
           0.1, 0.2], [0.2, 0.2, 0.1, 0.1, 0.2], [0.1,
           0.1, 0.1, 0.1, 0.1]])
In[3 ]:    I = array([[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0,
           0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 0, 1]])
In[4 ]:    C = I - A
In[ 5]:    from scipy.linalg import qr
           Q,R = qr(C)
In[6 ]:    print(Q)
In[7 ]:    print(R)
In[8 ]:    from numpy.linalg import pinv
           K = pinv(Q)
In[9 ]:    import numpy as np
           b = np.array([[30],[40], [50], [70], [90]])
In[ 10]:   Y = K.dot(b)
In[11 ]:   E = pinv(R)
In[12 ]:   X = E.dot(Y)
           X
Out[12]:   array([[411.02941176],
                  [474.90808824],
                  [545.88439542],
                  [402.82883987],
                  [303.8500817 ]])
```

Bước 3: Kết luận: Mức tổng cầu của mỗi ngành sản xuất:

$$\begin{Bmatrix} \{x_1 = 411.02941176, x_2 = 474.90808824, \\ x_3 = 545.88439542, x_4 = 402.82883987, \\ x_5 = 303.8500817 \end{Bmatrix}$$

Ví dụ 2: Giả sử trong một nền kinh tế có 6 ngành sản xuất: ngành 1, ngành 2, ngành 3, ngành 4, ngành 5, ngành 6. Biết ma trận hệ số kỹ thuật là: $A = \begin{bmatrix} 0,3 & 0,1 & 0,2 & 0,1 & 0,2 & 0,1 \\ 0,2 & 0,3 & 0,2 & 0,1 & 0,2 & 0,1 \\ 0,1 & 0,4 & 0,3 & 0,1 & 0,2 & 0,1 \\ 0,2 & 0,2 & 0,1 & 0,1 & 0,2 & 0,1 \\ 0,1 & 0,1 & 0,1 & 0,1 & 0,1 & 0,2 \\ 0,1 & 0,1 & 0,1 & 0,1 & 0,1 & 0,2 \end{bmatrix}$

Và mức cầu cuối cùng đối với sản phẩm của từng ngành thứ tự là 30, 40, 50, 60, 70 và 90 (đơn vị tính: triệu dollar). Hãy xác định mức tổng cầu của mỗi ngành sản xuất.

Đáp số: Mức tổng cầu của mỗi ngành sản xuất là:

$$\begin{Bmatrix} x_1 = 1208.26498423, x_2 = 1371.79810726, \\ x_3 = 1553.50157729, x_4 = 1099.2681388, \\ x_5 = 853.26182965, x_6 = 873.26182965 \end{Bmatrix}$$

3. KẾT LUẬN

Sử dụng QR decomposition kết hợp với phương pháp lập trình, giúp cho việc tìm ra nhanh chóng kết quả lời giải các hệ phương trình trong các tình huống khác nhau và ứng dụng trong giải mô hình tuyến tính trong phân tích kinh tế. Điều đó cho thấy, trong trường đại học cần kết hợp giảng dạy Toán, lập trình và tăng cường tính ứng dụng của sự kết hợp đó trong giải quyết các vấn đề kinh tế. Quá trình đào tạo theo hướng này sẽ giúp cho sự đáp ứng yêu cầu ngày càng cao của thị trường lao động đối với chất lượng nguồn nhân lực từ các trường đại học.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Huy Hoàng (2010), *Toán cao cấp*, tập 1, *Đại số tuyến tính*, Nxb Giáo dục Việt Nam.

[2] Nguyễn Đình Trí (2014), *Bài tập Toán cao cấp*, tập 1, Nxb Giáo dục Việt Nam.

[3] Trung Tâm Tin học, Trường Đại học Khoa học Tự nhiên Thành phố Hồ Chí Minh (2019), *Mathematics and Statistics for Datascience*, lưu hành nội bộ.

[4] I.B. Prockyriakob (1978), *Tuyển tập các bài toán về Đại số tuyến tính*, Nxb Khoa học Maxcơva (tiếng Nga).

[5] David C.Lay-Steven R.Lay-Judi J.Mc.Donald (2016), *Linear Algebra and Its Applications*, fifth edition, Pearson.

Ngày nhận bài: 29-10-2021. Ngày biên tập xong: 20-12-2021. Duyệt đăng: 12-01-2022