

The need to solve linear systems arises in an estimated 75% of all scientific computed problems (Dahlquist & Björck in *Numerical Methods*, 1974).

### Cálculo Numérico - Sistemas Lineares

Uma equação é linear se cada um de seus termos contém não mais do que uma variável e se cada variável aparece na primeira potência. Por exemplo  $3x+2y+3z=8$  é linear, mas  $3x+2yz+3w=27$  não é, pois o segundo termo tem  $yz$  e também  $2x^2+4y+5z=90$  também não é linear, pois aparece  $x^2$ .

Vamos considerar  $n$  equações lineares com  $n$  incógnitas e vamos nos referir a esse conjunto como um Sistema Linear de ordem  $n$ . Uma solução para tal sistema é um conjunto de  $n$  valores para as incógnitas, de maneira a que quando esses valores substituem as variáveis nas equações, todas elas são simultaneamente satisfeitas. Por exemplo, o sistema

$$\begin{cases} x + y + z = 1 \\ x - y - z = 1 \\ 2x + 3y - 4z = 9 \end{cases}$$

tem a solução  $x = 1, y = 1$  e  $z = -1$ . Você pode verificar e se certificar disso substituindo estas variáveis no sistema. O mesmo sistema também pode ser escrito na forma matricial

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ 2 & 3 & -4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 9 \end{pmatrix}$$

Para quem, como nós, conhece a multiplicação matricial, o sistema acima também pode ser escrito como

$$A.x = b,$$

onde  $A$  é chamada Matriz de Coeficientes,  $b$  é o termo independente e  $x$  é o vetor solução. Dado um sistema linear qualquer o mesmo pode ser classificado em termos do número de soluções que ele admite:

- Sistema possível com solução única. Por exemplo, seja

$$\begin{cases} x + y = 6 \\ x - y = 2 \end{cases}$$

Este sistema, que pode ser representado na figura

admite uma única solução, representada pelo ponto (4,2).

- Sistema possível com infinitas soluções. Por exemplo, seja

$$\begin{cases} x + y = 1 \\ 2x + 2y = 2 \end{cases}$$

Como se pode ver na figura as soluções são infinitas (sobre toda a linha)

- Sistema impossível, sem solução. Por exemplo

$$\begin{cases} x + y = 1 \\ x + y = 4 \end{cases}$$

Como se pode ver na figura, as retas são paralelas e portanto não têm ponto comum.

### Método do escalonamento de Gauss

Resolver um sistema linear  $A.x = b$  é achar um sistema equivalente  $A'.x = b'$  que tenha solução imediata, ou pelo menos mais simples do que o sistema original. Dois sistemas são ditos equivalentes se, embora diferentes, têm a mesma solução. As operações que são realizadas sobre os sistemas e que não lhes alteram as soluções, são:

- Trocar uma linha (equação) de lugar com outra.
- Multiplicar uma linha (equação) por uma constante
- Trocar uma linha por uma soma de 2 linhas

Seja por exemplo, resolver o sistema

$$\begin{cases} x + 4y + 4z = 23 \\ 3x + 3y + z = 27 \\ 2x + 5y + 2z = 25 \end{cases}$$

Passando para uma matriz mais simples

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 3 & 3 & 1 & 27 \\ 2 & 5 & 2 & 25 \end{array} \right)$$

aplicando à terceira linha a propriedade que diz  $L_3 = L_3 - p.L_1$  e escolhendo  $p$  de modo que surja um zero na primeira posição da linha, fica:  $2, 5, 2, 25 - p.1, 4, 4, 23$  e escolhendo  $p = 2$ , a nova linha passa a ser  $0, -3, -6, -21$  e a matriz fica:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 3 & 3 & 1 & 27 \\ 0 & -3 & -6 & -21 \end{array} \right)$$

aplicando à segunda linha a propriedade que diz  $L_2 = L_2 - p.L_1$  e fazendo  $p = 3$  para que surja um zero na primeira posição e a linha fica:  $3, 3, 1, 27 - p.1, 4, 4, 23$ . Com  $p = 3$ , a nova linha fica sendo  $0, -9, -11, -42$  e a matriz agora é:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 0 & -9 & -11 & -42 \\ 0 & -3 & -6 & -21 \end{array} \right)$$

Finalmente, elimina-se o  $-3$  na matriz acima de maneira a deixar a matriz escalonada:  $L_3 = L_3 - p.L_2$  e fazendo  $p = 0.333333$ , a segunda linha fica sendo  $0, 0, -2.333333, -7$  e a matriz escalonada agora é:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 0 & -9 & -11 & -42 \\ 0 & 0 & -2.33333 & -7 \end{array} \right)$$

Examinando a última linha, percebe-se que  $z = -2.33333 = 3$ . Levando este valor para a linha anterior, acha-se  $y = 1$  e levando os dois valores na primeira linha, acha-se  $x = 7$ . Ou seja, o vetor  $\{7, 1, 3\}$  é a solução procurada do sistema. O algoritmo em Python que resolve este sistema é

```
def gauss(a,b):
    n=len(b)
    x=[0]*n
    for passo in range(0,n-1):
        for i in range(passo+1,n):
            pivot=a[i,passo]/a[passo,passo]
            for j in range(0,n):
                a[i,j]=a[i,j]-(pivot*a[passo,j])
            b[i]=b[i]-pivot*b[passo]
    x[n-1]=b[n-1]/a[n-1,n-1]
    for i in range(i,-1,-1):
        x[i]=b[i]
        for j in range(i+1,n):
            x[i]=x[i]-a[i,j]*x[j]
        x[i]=x[i]/a[i,i]
    return(x)
import numpy as np
a=np.array([[3.0,-1,1],[1,2,-1],[4,-2,-1]],float)
b=np.array([11.0,6,-5],float)
x=gauss(a,b)
print(x)
```

### Método LU

No método LU, deve-se transformar a matriz  $A$  (dos coeficientes) em um produto matricial  $L.U$  onde a matriz  $U$  (de Upper) é a própria matriz  $A$  devidamente escalonada (como visto no método anterior) e a matriz  $L$  (de Lower) é uma matriz que contém unidades na diagonal principal, zeros acima destas posições e que contém os pivots do método anterior abaixo da diagonal principal. No exemplo anterior, as matrizes  $L$  e  $U$  são

$$U = \begin{pmatrix} 1.00 & 4.00 & 4.00 \\ 0 & -9.00 & -11.00 \\ 0 & 0 & -2.33 \end{pmatrix}$$

$$L = \begin{pmatrix} 1.00 & 0 & 0 \\ 3.00 & 1.00 & 0 \\ 2.00 & 0.33 & 1 \end{pmatrix}$$

Note que a multiplicação matricial entre  $L.U$  recupera a matriz  $A$ . Obtendo estas duas matrizes, é fácil calcular  $y$  (o vetor de termos independentes que é calculado junto com a matriz escalonada no método anterior e depois o vetor  $x$  de soluções. Em termos algorítmicos, fica  $[L]\{y\} = \{b\}$ . Agora é praticamente imediato calcular o vetor  $\{y\}$ . Finalmente, o sistema se resolve calculando  $[U].\{x\} = \{y\}$ . Acompanhe o algoritmo

```
def lu1(u,b):
    n=len(u)
    x=[0]*n
    y=[0]*n
    L=np.zeros((n,n))
    for passo in range(0,n):
        for i in range(passo+1,n):
            pivot=u[i,passo]/u[passo,passo]
            for j in range(0,n):
                u[i,j]=u[i,j]-pivot*u[passo,j]
            L[i,passo]=pivot
    for i in range(0,n):
        L[i,i]=1
    y[0]=b[0]
    for i in range(1,n):
        y[i]=b[i]
        for j in range(0,i):
            y[i]=y[i]-L[i,j]*y[j]
    x[n-1]=y[n-1]/u[n-1,n-1]
    for i in range(n-1,-1,-1):
        x[i]=y[i]
        for j in range(i+1,n):
            x[i]=x[i]-u[i,j]*x[j]
        x[i]=x[i]/u[i,i]
    print('-----L-----')
    print(L)
    print('-----y-----')
    print(y)
    print('-----u-----')
    print(u)
    print('-----x-----')
    print(x)
import numpy as np
a=np.array([[3.0,-1,1],[1,2,-1],[4,-2,-1]],float)
b=np.array([11.0,6,-5],float)
lu1(a,b)
```

### Para você fazer

Resolva pelos dois métodos o sistema

$$\begin{pmatrix} 3 & 1 & 1 & 4 & 7 & 28 \\ 7 & 4 & 4 & 4 & 3 & 31 \\ 1 & 3 & 1 & 2 & 4 & 20 \\ 3 & 1 & 2 & 7 & 2 & 31 \\ 4 & 4 & 6 & 1 & 6 & 39 \end{pmatrix}$$

e mostre no primeiro método a matriz escalonada e no segundo método as matrizes  $L$  e  $U$  (pode usar o verso desta folha, além de apresentar os 5 valores do vetor resposta  $x$ , que por óbvio devem ser os mesmos em ambos os métodos

| x | y | z | w | t |
|---|---|---|---|---|
|   |   |   |   |   |



The need to solve linear systems arises in an estimated 75% of all scientific computed problems (Dahlquist & Björck in *Numerical Methods*, 1974).

Cálculo Numérico - Sistemas Lineares

Uma equação é linear se cada um de seus termos contém não mais do que uma variável e se cada variável aparece na primeira potência. Por exemplo  $3x+2y+3z=8$  é linear, mas  $3x+2yz+3w=27$  não é, pois o segundo termo tem  $yz$  e também  $2x^2+4y+5z=90$  também não é linear, pois aparece  $x^2$ .

Vamos considerar  $n$  equações lineares com  $n$  incógnitas e vamos nos referir a esse conjunto como um Sistema Linear de ordem  $n$ . Uma solução para tal sistema é um conjunto de  $n$  valores para as incógnitas, de maneira a que quando esses valores substituem as variáveis nas equações, todas elas são simultaneamente satisfeitas. Por exemplo, o sistema

$$\begin{cases} x + y + z = 1 \\ x - y - z = 1 \\ 2x + 3y - 4z = 9 \end{cases}$$

tem a solução  $x = 1, y = 1$  e  $z = -1$ . Você pode verificar e se certificar disso substituindo estas variáveis no sistema. O mesmo sistema também pode ser escrito na forma matricial

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ 2 & 3 & -4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 9 \end{pmatrix}$$

Para quem, como nós, conhece a multiplicação matricial, o sistema acima também pode ser escrito como

$$A.x = b,$$

onde  $A$  é chamada Matriz de Coeficientes,  $b$  é o termo independente e  $x$  é o vetor solução. Dado um sistema linear qualquer o mesmo pode ser classificado em termos do número de soluções que ele admite:

- Sistema possível com solução única. Por exemplo, seja

$$\begin{cases} x + y = 6 \\ x - y = 2 \end{cases}$$

Este sistema, que pode ser representado na figura

admite uma única solução, representada pelo ponto (4,2).

- Sistema possível com infinitas soluções. Por exemplo, seja

$$\begin{cases} x + y = 1 \\ 2x + 2y = 2 \end{cases}$$

Como se pode ver na figura as soluções são infinitas (sobre toda a linha)

- Sistema impossível, sem solução. Por exemplo

$$\begin{cases} x + y = 1 \\ x + y = 4 \end{cases}$$

Como se pode ver na figura, as retas são paralelas e portanto não têm ponto comum.

Método do escalonamento de Gauss

Resolver um sistema linear  $A.x = b$  é achar um sistema equivalente  $A'.x = b'$  que tenha solução imediata, ou pelo menos mais simples do que o sistema original. Dois sistemas são ditos equivalentes se, embora diferentes, têm a mesma solução. As operações que são realizadas sobre os sistemas e que não lhes alteram as soluções, são:

- Trocar uma linha (equação) de lugar com outra.
- Multiplicar uma linha (equação) por uma constante
- Trocar uma linha por uma soma de 2 linhas

Seja por exemplo, resolver o sistema

$$\begin{cases} x + 4y + 4z = 23 \\ 3x + 3y + z = 27 \\ 2x + 5y + 2z = 25 \end{cases}$$

Passando para uma matriz mais simples

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 3 & 3 & 1 & 27 \\ 2 & 5 & 2 & 25 \end{array} \right)$$

aplicando à terceira linha a propriedade que diz  $L_3 = L_3 - p.L_1$  e escolhendo  $p$  de modo que surja um zero na primeira posição da linha, fica:  $2, 5, 2, 25 - p.1, 4, 4, 23$  e escolhendo  $p = 2$ , a nova linha passa a ser  $0, -3, -6, -21$  e a matriz fica:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 3 & 3 & 1 & 27 \\ 0 & -3 & -6 & -21 \end{array} \right)$$

aplicando à segunda linha a propriedade que diz  $L_2 = L_2 - p.L_1$  e fazendo  $p = 3$  para que surja um zero na primeira posição e a linha fica:  $3, 3, 1, 27 - p.1, 4, 4, 23$ . Com  $p = 3$ , a nova linha fica sendo  $0, -9, -11, -42$  e a matriz agora é:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 0 & -9 & -11 & -42 \\ 0 & -3 & -6 & -21 \end{array} \right)$$

Finalmente, elimina-se o  $-3$  na matriz acima de maneira a deixar a matriz escalonada:  $L_3 = L_3 - p.L_2$  e fazendo  $p = 0.333333$ , a segunda linha fica sendo  $0, 0, -2.333333, -7$  e a matriz escalonada agora é:

$$\left( \begin{array}{ccc|c} 1 & 4 & 4 & 23 \\ 0 & -9 & -11 & -42 \\ 0 & 0 & -2.33333 & -7 \end{array} \right)$$

Examinando a última linha, percebe-se que  $z = -2.3333 = 3$ . Levando este valor para a linha anterior, acha-se  $y = 1$  e levando os dois valores na primeira linha, acha-se  $x = 7$ . Ou seja, o vetor  $\{7, 1, 3\}$  é a solução procurada do sistema. O algoritmo em Python que resolve este sistema é

```
def gauss(a,b):
    n=len(b)
    x=[0]*n
    for passo in range(0,n-1):
        for i in range(passo+1,n):
            pivot=a[i,passo]/a[passo,passo]
            for j in range(0,n):
                a[i,j]=a[i,j]-(pivot*a[passo,j])
            b[i]=b[i]-pivot*b[passo]
    x[n-1]=b[n-1]/a[n-1,n-1]
    for i in range(i,-1,-1):
        x[i]=b[i]
        for j in range(i+1,n):
            x[i]=x[i]-a[i,j]*x[j]
        x[i]=x[i]/a[i,i]
    return(x)
import numpy as np
a=np.array([[3.0,-1,1],[1,2,-1],[4,-2,-1]],float)
b=np.array([11.0,6,-5],float)
x=gauss(a,b)
print(x)
```

Método LU

No método LU, deve-se transformar a matriz  $A$  (dos coeficientes) em um produto matricial  $L.U$  onde a matriz  $U$  (de Upper) é a própria matriz  $A$  devidamente escalonada (como visto no método anterior) e a matriz  $L$  (de Lower) é uma matriz que contém unidades na diagonal principal, zeros acima destas posições e que contém os pivots do método anterior abaixo da diagonal principal. No exemplo anterior, as matrizes  $L$  e  $U$  são

$$U = \begin{pmatrix} 1.00 & 4.00 & 4.00 \\ 0 & -9.00 & -11.00 \\ 0 & 0 & -2.33 \end{pmatrix}$$

$$L = \begin{pmatrix} 1.00 & 0 & 0 \\ 3.00 & 1.00 & 0 \\ 2.00 & 0.33 & 1 \end{pmatrix}$$

Note que a multiplicação matricial entre  $L.U$  recupera a matriz  $A$ . Obtendo estas duas matrizes, é fácil calcular  $y$  (o vetor de termos independentes que é calculado junto com a matriz escalonada no método anterior e depois o vetor  $x$  de soluções. Em termos algorítmicos, fica  $[L]\{y\} = \{b\}$ . Agora é praticamente imediato calcular o vetor  $\{y\}$ . Finalmente, o sistema se resolve calculando  $[U].\{x\} = \{y\}$ . Acompanhe o algoritmo

```
def lu1(u,b):
    n=len(u)
    x=[0]*n
    y=[0]*n
    L=np.zeros((n,n))
    for passo in range(0,n):
        for i in range(passo+1,n):
            pivot=u[i,passo]/u[passo,passo]
            for j in range(0,n):
                u[i,j]=u[i,j]-pivot*u[passo,j]
            L[i,passo]=pivot
    for i in range(0,n):
        L[i,i]=1
    y[0]=b[0]
    for i in range(1,n):
        y[i]=b[i]
        for j in range(0,i):
            y[i]=y[i]-L[i,j]*y[j]
    x[n-1]=y[n-1]/u[n-1,n-1]
    for i in range(n-1,-1,-1):
        x[i]=y[i]
        for j in range(i+1,n):
            x[i]=x[i]-u[i,j]*x[j]
        x[i]=x[i]/u[i,i]
    print('-----L-----')
    print(L)
    print('-----y-----')
    print(y)
    print('-----u-----')
    print(u)
    print('-----x-----')
    print(x)
import numpy as np
a=np.array([[3.0,-1,1],[1,2,-1],[4,-2,-1]],float)
b=np.array([11.0,6,-5],float)
lu1(a,b)
```

Para você fazer

Resolva pelos dois métodos o sistema

$$\begin{pmatrix} 3 & 3 & 1 & 3 & 7 & | & 32 \\ 4 & 1 & 5 & 2 & 7 & | & 33 \\ 6 & 2 & 2 & 3 & 1 & | & 21 \\ 7 & 2 & 7 & 6 & 3 & | & 48 \\ 7 & 1 & 6 & 7 & 1 & | & 38 \end{pmatrix}$$

e mostre no primeiro método a matriz escalonada e no segundo método as matrizes  $L$  e  $U$  (pode usar o verso desta folha, além de apresentar os 5 valores do vetor resposta  $x$ , que por óbvio devem ser os mesmos em ambos os métodos

| x | y | z | w | t |
|---|---|---|---|---|
|   |   |   |   |   |



- 2 - /



===== 28/10/2021 15:26:49.4 =====E=PL735p

|   |    |   |   |   |   |
|---|----|---|---|---|---|
| 1 | -1 | 1 | 4 | 3 | 2 |
| 2 | -1 | 6 | 4 | 2 | 1 |
| 3 | 5  | 3 | 2 | 9 | 4 |