Notas de Aula – Métodos Numéricos

Aula XX – Introdução aos Sistemas de Equações Lineares

Prof. Paulo Ribeiro

Sumário

Nesta aula você aprenderá...

1. o que é um sistema de equações lineares, e como pode ser usado para solucionar problemas reais;
2. como classificar sistemas lineares;
3. definir sistemas triangulares superiores e inferiores;
4. solucionar sistemas triangulares por abordagem analítica e computacional, usando a biblioteca `scipy`.

Introdução aos Sistemas de Equações Lineares

Boa parte do material disponível nessas notas são adaptados do livro Cálculo Numérico: Um Livro Colaborativo Versão Python e dos slides de aula do Professor Bordoni, não sendo, portanto, notas totalmente originais. São feitos complementos, principalmente nos exercícios e exemplos resolvidos de outros livros, que serão devidamente citados.

Muitos problemas da engenharia, física e matemática estão associados à solução de sistemas de equações lineares.

Dois exemplos bem ilustrativos e extremamente importantes são a tomografia computadorizada, que faz uso de sistemas de equações lineares para estimar, para cada pixel da imagem obtida no tomógrafo, a densidade do anteparo (no caso, paciente), descobrindo, assim, se esse anteparo tem material mais líquido, mais sólido, se trata-se de músculo, tumor, osso e por aí vai (esse artigo tem uma excelente descrição de como obter um sistema de equações lineares para esse caso) e os sistemas de recomendação, como, por exemplo, o usado pelo Netflix, que faz uso desses sistemas de equações considerando como incógnitas cada um dos parâmetros do sistema (esse artigo mostra de forma bem simplória como funciona o sistema de recomendações da Netflix, exemplificando sua implementação com Excel).

Nessa parte de nosso curso, trataremos de técnicas numéricas empregadas para obter a solução desses sistemas. Veremos como definir e classificar sistemas de equações lineares, as soluções para sistemas triangulares, métodos diretos de solução - Eliminação Gaussiana e fatoração LU, métodos iterativos - Gauss-Seidel e Jacobi e o método de decomposição em valor singular (SVD), importante método usado largamente em sistemas baseados em aprendizado de máquinas.

Começaremos definindo o que são sistemas de equações lineares.

Um **sistema de equações lineares** (abreviadamente, sistema linear) é um conjunto finito de equações lineares aplicadas em um mesmo conjunto, igualmente finito, de variáveis reais ou complexas (nos deteremos somente as variáveis reais).

Uma **solução** para um sistema linear é uma atribuição de números às incógnitas que satisfazem simultaneamente todas as equações do sistema. A palavra *sistema*, nesse contexto, indica que as equações devem ser consideradas em conjunto, e não de forma individual, isto é, cada uma das equações precisa ser satisfeita pelo conjunto de soluções encontradas.

Tipicamente, sistemas de equações são compostos por m equações lineares, cada uma delas com n incógnitas, comumente descritas da forma

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m.
\end{aligned} \tag{1}
$$

Esse formato é chamado de **forma algébrica** do sistema de equações lineares. Esse mesmo sistema pode ser representado na sua forma matricial

$$
Ax = b \tag{2}
$$

em que

$$
A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{3}
$$

sendo A chamada de **matriz dos coeficientes**, x de **vetor das incógnitas** e b de **vetor dos termos independentes**.

Exemplo 1.

Consideremos o seguinte sistema linear

$$
\begin{aligned}
x + y + z &= 1 \\
4x + 4y + 2z &= 2 \\
2x + y - z &= 0.
\end{aligned} \tag{4}
$$

Escreva ela na sua forma matricial.

Na sua forma matricial, este sistema é escrito como

$$
Ax = b \Leftrightarrow \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 4 & 4 & 2 \\ 2 & 1 & -1 \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}}_{b}. \tag{5}
$$

Classificação de Sistemas de Equações Lineares

A solução de um sistema linear é a atribuição de valores às variáveis $x_1, x_2, \ldots, x_n$ de modo a satisfazer todas as equações. O grupo de todas as soluções possíveis é chamado de conjunto-solução.

Um sistema linear pode comportar-se, com relação às suas soluções, em qualquer uma das três formas possíveis:

i. **Sistema Impossível**: é o sistema que não admite uma solução, sendo designado por sistema impossível e suas equações como equações incompatíveis. Por exemplo,

$$\begin{aligned} x_1 + x_2 &= 3 \\ 2x_1 + 2x_2 &= 9 \end{aligned} \tag{6}$$

ii. **Sistema Possível Determinado**: é o sistema que possui apenas uma única solução possível. Esse sistema pode ser chamado de sistema possível e suas equações de equações compatíveis. Por exemplo,

$$\begin{aligned} x_1 + x_2 &= 4 \\ x_1 - x_2 &= 8 \end{aligned} \tag{7}$$

iii. **Sistema Possível Indeterminado**: É o sistema que possui múltiplas soluções. Por exemplo,

$$\begin{aligned} x_1 + x_2 &= 4 \\ 2x_1 + 2x_2 &= 8 \end{aligned} \tag{8}$$

iv. **Sistema Possível Homogêneo**: É o sistema que que possui $b = 0$ e, portanto, tem sempre $x = 0$ como uma possível solução. Por exemplo,

$$\begin{aligned} x_1 + x_2 &= 0 \\ 2x_1 + 3x_2 &= 0 \end{aligned} \tag{9}$$

Solução de Sistemas de Equações Lineares

Durante os anos de estudos básicos (ensino fundamental e médio) nós somos apresentados à sistemas de equações com duas incógnitas (como os mostrados nos exemplos acima, de classificação de sistemas), cujas soluções podem ser feitas por substituição. Bastava isolar uma das incógnitas em uma equação, depois substituir esse incógnita isolada na equação seguinte e encontrar os valores individualmente.

Basicamente, nós podemos fazer isso para qualquer número de equações e incógnitas, bastando, para isso, que o sistema seja **triangular**.

Sistemas de equações triangulares são sistemas cuja matriz dos coeficientes é uma matriz triangular superior ou inferior, podendo ser classificados dessa maneira.

Assim, na forma algébrica, **sistemas triangulares superiores** tem o formato

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{mn}x_n &= b_m \end{aligned} \tag{10}$$

em que

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 0 & a_{mn} \end{bmatrix} \tag{11}$$

é uma **matriz triangular superior** (os elementos abaixo da diagonal principal são nulos, isto é, $a_{ij} = 0, \forall j < i$).

E **sistemas triangulares inferiores** tem, na forma algébrica, o formato

$$\begin{aligned} a_{11}x_1 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \\ a_{21}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{12}$$

em que

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & \cdots & a_{mn} \end{bmatrix} \tag{13}$$

é uma **matriz triangular inferior** (os elementos acima da diagonal principal são nulos, , isto é, $a_{ij} = 0, \forall j < i$).

Para obter a solução desses sistemas, basta recorrermos à substituições retroativas, no caso de sistemas triangulares superiores, ou à substituições progressivas, no caso de sistemas triangulares inferiores, como mostrado a seguir.

Solução de Sistemas Triangulares Superiores

Consideremos o seguinte sistema triangular superior, como exemplo:

$$\begin{aligned} 3x_1 + 4x_2 - 5x_3 + x_4 &= -10 \\ x_2 + x_3 - 2x_4 &= -1 \\ 4x_3 - 5x_4 &= 3 \\ 2x_4 &= 2. \end{aligned} \tag{14}$$

É fácil verificar que podemos encontrar o valor da incógnita x_4 diretamente da última equação. Assim,

$$2x_4 = 2 \quad \Rightarrow \quad x_4 = \frac{2}{2} = 1. \tag{15}$$

De posse do valor de x_4, podemos substituí-lo na penúltima equação e encontrar o valor da incógnita x_3

$$4x_3 - 5x_4 = 3 \quad \Rightarrow \quad 4x_3 - 5 \cdot 1 = 3 \quad \Rightarrow \quad x_3 = 2. \tag{16}$$

Seguindo esse procedimento, podemos obter

$$x_2 + x_3 - 2x_4 = -1 \quad \Rightarrow \quad x_2 + 2 - 2 \cdot 1 = -1 \quad \Rightarrow \quad x_2 = -1 \tag{17}$$

e

$$3x_1 + 4x_2 - 5x_3 + x_4 = -10 \quad \Rightarrow \quad 3x_1 + 4 \cdot (-1) - 5 \cdot 2 + 1 = -10 \quad \Rightarrow \quad x_1 = 1. \tag{18}$$

E, com isso, encontramos a solução para o sistema de equações lineares.

Não é difícil perceber (pode, inclusive, ser verificado) que, se houver garantia de que o sistema é triangular superior, o procedimento pode ser generalizado para qualquer quantidade de equações e incógnitas. Esse procedimento é chamado de **substituição retroativa**, uma vez que as substituições são feitas de "baixo para cima", ou seja da incógnita de maior índice para a de menor.

De forma generalizada, para um sistema descrito de forma algébrica por

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{mn}x_n &= b_m \end{aligned} \tag{19}$$

podemos dizer que as incógnitas do sistema podem ser encontradas por substituição retroativa, descrita por meio da equação

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}, \qquad \forall i = m, \ldots, 1. \tag{20}$$

É interessante perceber que os índices das linhas do sistema são varridos do maior para o menor, como feito no exemplo anterior.

Também é possível mostrar, acerca da soluções, que

i. se $a_{ii} \neq 0$, $\forall i$, temos um sistema possível e determinado;

ii. se $a_{ii} = 0$, para algum valor de i, há dois casos a analisar:

- se $b_i - \displaystyle\sum_{j=i+1}^{n} a_{ij}x_j = 0$, o sistema é possível e indeterminado;
- se $b_i - \displaystyle\sum_{j=i+1}^{n} a_{ij}x_j \neq 0$, o sistema é impossível.

Exercício 1. Verifique, refazendo o exemplo acima, se a equação da substituição retroativa, descrita na equação 20, de fato funciona.

Exercício 2. Faça um programa em Python que calcule a solução de um sistema triangular superior usando substituição retroativa. Para representação das matrizes e vetores use exclusivamente Numpy.

Solução de Sistemas Triangulares Inferiores

A ideia da substituição é exatamente a mesma para sistemas triangulares inferiores, diferenciando-se apenas que, ao invés de começar pela incógnita de maior índice, inicia-se pela de menor índice, sendo esse método, por esse motivo, referido como **substituição progressiva**.

De forma generalizada, considerando o sistema da forma

$$\begin{aligned} a_{11}x_1 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \\ a_{21}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{21}$$

podemos dizer que as incógnitas do sistema podem ser encontradas por substituição retroativa, descrita por meio da equação

$$x_i = \frac{b_i - \sum_{j=i}^{n-1} a_{ij}x_j}{a_{ii}}, \qquad \forall i = 1, \ldots, m. \tag{22}$$

Também é possível mostrar, acerca da soluções, que

i. se $a_{ii} \neq 0$, $\forall i$, temos um sistema possível e determinado;

ii. se $a_{ii} = 0$, para algum valor de i, há dois casos a analisar:

- se $b_i - \displaystyle\sum_{j=i}^{n-1} a_{ij}x_j = 0$, o sistema é possível e indeterminado;
- se $b_i - \displaystyle\sum_{j=i}^{n-1} a_{ij}x_j \neq 0$, o sistema é impossível.

Exercício 3. Faça um programa em Python que calcule a solução de um sistema triangular inferior usando substituição progressiva. Para representação das matrizes e vetores use exclusivamente Numpy.

Solução de Sistemas Triangulares usando `Scipy`

No submódulo `linalg` do `scipy`, existe a função `solve_triangular`, que, conforme diz sua descrição na documentação, resolve a equação $Ax = b$ para x, assumindo que A é uma matriz triangular.

Da documentação da função, temos que destacar os seguintes parâmetros:

- `a`: a matriz triangular dos coeficientes do sistema;
- `b`: o vetor de termos independentes;
- `lower`: indica se a solução é obtida no triangulo inferior de A (sistemas triangulares inferiores) – `True` ou no triângulo superior (sistemas triangulares superiores) – `False`, opção *default*.

Os demais parâmetros são para casos muito particulares, não tratados aqui.

A função retorna, então, um `array` do `numpy`, contendo o vetor de incógnitas x, possuindo a mesma forma do vetor b.

Vejamos dois exemplos de uso.

```
import scipy.linalg as sla
import numpy as np
```

Exemplo 2.

Resolva o exemplo descrito em (14) usando a função `solve_triangular`.

O sistema é dado por

$$\begin{aligned} 3x_1 + 4x_2 - 5x_3 + x_4 &= -10 \\ x_2 + x_3 - 2x_4 &= -1 \\ 4x_3 - 5x_4 &= 3 \\ 2x_4 &= 2 \end{aligned} \tag{23}$$

cuja forma matricial é dada por

$$A = \begin{bmatrix} 3 & 4 & -5 & 1 \\ 0 & 1 & 1 & -2 \\ 0 & 0 & 4 & -5 \\ 0 & 0 & 0 & 2 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad b = \begin{bmatrix} -10 \\ -1 \\ 3 \\ 2 \end{bmatrix} \tag{24}$$

Em python, temos

```
A = np.array([[3, 4, -5, 1],[0, 1, 1, -2],[0, 0, 4, -5],[0, 0, 0, 2]])
b = np.array([-10, -1, 3, 2])
```

Assim,

```
x = sla.solve_triangular(A,b)
```

```
x
```

Saída:

```
array([ 1., -1., 2., 1.])
```

que é o mesmo resultado observado no exemplo. Nós podemos verificar se o resultado está correto resolvendo o sistema, ou seja, se ao multiplicar A por x obtivermos b, tudo certo. Dessa forma, basta fazer

```
A.dot(x)
```

Saída:

```
array([-10., -1., 3., 2.])
```

Exemplo 3.

Ache o conjunto solução para o sistema

$$
\begin{aligned}
3x_1 &= 4 \\
2x_1 + x_2 &= 2 \\
x_1 + x_3 &= 4 \\
x_1 + x_2 + x_3 + x_4 &= 2
\end{aligned}
\tag{25}
$$

A forma matricial do sistema é dada por

$$
A = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad b = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 2 \end{bmatrix} \tag{26}
$$

Em python, temos

```
A = np.array([[3, 0, 0, 0], [2, 1, 0, 0], [1, 0, 1, 0], [1, 1, 1, 1]])
b = np.array([4, 2, 4, 2])
```

No caso, como o sistema é triangular inferior, precisamos especificar isso, ajustando o parâmetro `lower=True` na função `solve_triangular`.

Assim,

```
x = sla.solve_triangular(A, b, lower=True)
x
```

Saída:

```
array([ 1.33333333, -0.66666667, 2.66666667, -1.33333333])
```

Verificando se o resultado está correto, temos

```
A.dot(x)
```

Saída:

```
array([4., 2., 4., 2.])
```

Exercício 4. Compare os resultados das implementações de suas funções para a solução de sistemas triangulares com a função do scipy. Os resultados são diferentes? Experimente com outros exemplos.