

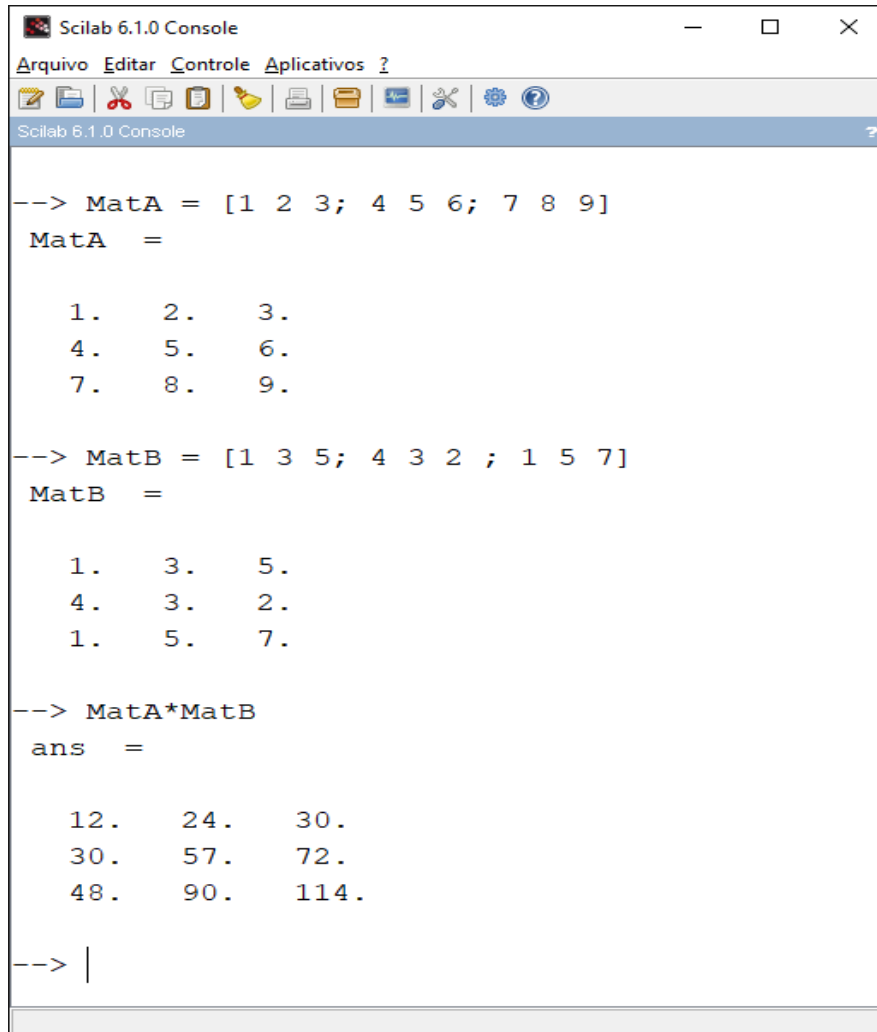
Multiplicando Matrizes

Mário Leite

...

Uma matriz pode ser considerada como “um conjunto de elemento (números ou letras) dispostos na forma de tabela (linhas e colunas)”. Na verdade, pelo menos teoricamente, qualquer elemento pode ser considerado elemento de uma matriz. O SciLab (ferramenta de cálculo numérico (da família do MatLab)) trata qualquer elemento numérico como elemento de uma matriz $m \times n$; assim no caso número (dígito) 7, por exemplo, este seria um elemento numa matriz 1×1 : uma linha e uma coluna.

O uso de matrizes facilita muito o nosso dia a dia quando temos que trabalhar com tabelas numéricas, de onde é possível extrair resultados que envolvem seus elementos. Seja na matemática teórica quanto na aplicação prática, como na Engenharia, Economia, Estatística, etc, a manipulação de matrizes torna-se muito importante. Por exemplo, calcular as médias em diversas disciplinas de vários alunos em vários meses. Deste modo, podemos executar vários tipos de cálculos com seus elementos: somar, subtrair e multiplicar; até mesmo dividir, quando trabalhamos com matriz transposta. Entre as operações fundamentais uma é um tanto problemática, e deve ser tratada com cuidado: a multiplicação; pois esta operação não é comutativa entre matrizes. Deste modo, dadas as matrizes **A** e **B**, o produto **A*B** é diferente de **B*A**. Para ilustrar esse tipo de operação, o programa “**MultiplicaMatrizes**”, codificado em Python mostra como multiplicar duas matrizes, verificando as duas possibilidades: **A*B** e **B*A**, dependendo de suas dimensões. A **figura 1** mostra como multiplicar duas matrizes com o SciLab, e a **figura 2** a saída do programa no *IDLE* do Python 3.10 .



```
--> MatA = [1 2 3; 4 5 6; 7 8 9]
MatA =

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

--> MatB = [1 3 5; 4 3 2 ; 1 5 7]
MatB =

    1.    3.    5.
    4.    3.    2.
    1.    5.    7.

--> MatA*MatB
ans =

    12.    24.    30.
    30.    57.    72.
    48.    90.   114.

--> |
```

Figura 1 - Multiplicando matrizes com o SciLab

Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: D:/Testtext.py

Digite o número de linhas da matriz A: 3

Digite o número de colunas da matriz A: 3

Digite o número de linhas da matriz B: 3

Digite o número de colunas da matriz B: 3

Digite os elementos da matriz A:

Entre com o elemento [1,1] da matriz A: 1

Entre com o elemento [1,2] da matriz A: 2

Entre com o elemento [1,3] da matriz A: 3

Entre com o elemento [2,1] da matriz A: 4

Entre com o elemento [2,2] da matriz A: 5

Entre com o elemento [2,3] da matriz A: 6

Entre com o elemento [3,1] da matriz A: 7

Entre com o elemento [3,2] da matriz A: 8

Entre com o elemento [3,3] da matriz A: 9

Digite os elementos da matriz B:

Entre com o elemento [1,1] da matriz B: 1

Entre com o elemento [1,2] da matriz B: 3

Entre com o elemento [1,3] da matriz B: 5

Entre com o elemento [2,1] da matriz B: 4

Entre com o elemento [2,2] da matriz B: 3

Entre com o elemento [2,3] da matriz B: 2

Entre com o elemento [3,1] da matriz B: 1

Entre com o elemento [3,2] da matriz B: 5

Entre com o elemento [3,3] da matriz B: 7

Matriz A:

1 2 3

4 5 6

7 8 9

Matriz B:

1 3 5

4 3 2

1 5 7

Matriz C:

12 24 30

30 57 72

48 90 114

Figura 2 - Um exemplo de entrada/saída do programa "MultiplicaMarizes.py"

```

'''
MultiplicaMatrizes.py
-----
Faz o produto de duas matrizes: AxB ou BxA
-----
'''
def LerMatriz(matriz, linhas, colunas):
    Lstmatriz = [] #inicia uma matriz-lsta
    print(f"\nDigite os elementos da matriz {matriz}:")
    for i in range(1, linhas+1):
        LstLin = []
        for j in range(1, colunas+1):
            elem = int(input(f"Entre com o elemento [{i},{j}] da matriz {matriz}: "))
            LstLin.append(elem)
        Lstmatriz.append(LstLin)
    print()
    return Lstmatriz

#-----
def ExibirMatriz(LstMatriz, nomeMatriz):
    print(f"\nMatriz {nomeMatriz}:")
    for lin in LstMatriz:
        print(" ".join(map(str, lin)))

#-----
def MultiplicarMatrizes():
    global LstMatA, LstMatB, LstMatC
    linA, colA = 0, 0
    while((linA < 2) or (linA > 10)):
        linA = abs(int(input("Digite o número de linhas da matriz A: ")))
    while((colA < 2) or (colA > 10)):
        colA = abs(int(input("Digite o número de colunas da matriz A: ")))
    print()
    linB, colB = 0, 0
    while((linB < 2) or (linB > 10)):
        linB = abs(int(input("Digite o número de linhas da matriz B: ")))
    while((colB < 2) or (colB > 10)):
        colB = abs(int(input("Digite o número de colunas da matriz B: ")))
    #Verifica se é possível fazer a multiplicação das matrizes
    if((colA != linB) or (colB != linA)):
        return False
    else:
        if(colB == linA):
            #Inverte as dimensões das matrizes usando "Método da Boilha"
            aux = linA
            linA = colB
            colB = aux
        LstMatA = LerMatriz("A", linA, colA)
        LstMatB = LerMatriz("B", linB, colB)
        LstMatC = [[0.0 for _ in range(colB)] for _ in range(linA)]
        for i in range(1, linA+1):
            for j in range(1, colB+1):
                somaProd = 0
                for k in range(1, colA+1):
                    somaProd += LstMatA[i-1][k-1] * LstMatB[k-1][j-1]
                LstMatC[i-1][j-1] = int(somaProd)
        return True

#=====
#Programa principal
if(MultiplicarMatrizes()):
    ExibirMatriz(LstMatA, "A")
    ExibirMatriz(LstMatB, "B")
    ExibirMatriz(LstMatC, "C")
else:
    print()
    print("Impossível multiplicar as matrizes!")
#Fim do programa "MultiplicaMatrizes" -----

```