

Aplicações de Matrizes Quadrimensionais

Mário Leite

Normalmente quando se fala sobre matriz, a ideia inicial é uma planilha tal como as do Excel: com largura e comprimento; duas dimensões. Muito raramente se pensa em matrizes tridimensionais, devido ao simples fato de que, apesar do nosso mundo ser 3D, é difícil sair do mundo bidimensional na programação de *arrays* (*linhas* e *colunas*) para mostrar algo que “salta” do papel. Como imprimir (ou mostrar na tela) as três dimensões de uma matriz?! Simplesmente impossível! Então, para uma matriz quadridimensional nem se fala...

Embora os físicos digam que a quarta dimensão é o Tempo - de acordo com a Teoria da Relatividade - para os leigos isto é muito difícil de entender. A Teoria das Cordas, então, é muito mais radical com suas 11 dimensões; aí é de enlouquecer de vez. Mas, matematicamente isto é perfeitamente viável, e fisicamente, na prática, quatro dimensões podem ser simuladas. Podemos considerar esse tipo de matriz como uma estrutura de dados indexada por quatro eixos independentes, geralmente representando tempo + três dimensões físicas ou quatro variáveis correlacionadas. Por exemplo, como o esquema abaixo, onde cada índice responde a **uma pergunta específica**:

Dimensão	Representa
1 ^a	Dia da medição
2 ^a	Altitude
3 ^a	Latitude
4 ^a	Longitude

Isso deixa claro que não é apenas “um *array* grande”, mas um modelo matemático de um fenômeno real; mas, é importante destacar:

- A estrutura da matriz **não muda**
- O que muda é **o significado físico dos dados**
- Isso reforça o conceito de **modelo multidimensional reutilizável**

Embora matrizes quadridimensionais possam parecer abstratas à primeira vista, elas surgem naturalmente sempre que precisamos representar **fenômenos dependentes de múltiplas variáveis simultâneas**, como tempo, espaço, profundidade ou estado. O uso de estruturas **4D** permite organizar os dados de forma lógica, previsível e extensível, aproximando o código computacional do modelo matemático do fenômeno real.

Os três programas abaixo, codificados em Python, mostram exemplos práticos de aplicações com matrizes quadridimensionais, e a **figura 1** mostra os respectivos resultados.

```

'''  

MatrizQuadridimensionalAplicada1.py  

-----  

Simula a aplicação de uma matriz tridimensional para prever a temperatura em uma  

região dividida em:  

5 altitudes diferentes (níveis da atmosfera),  

10 latitudes (faixas no eixo norte-sul),  

15 longitudes (faixas no eixo leste-oeste).  

-----  

'''  

import numpy as np  

#Cria uma matriz com zeros: (dias, altitude, latitude, longitude)  

Temperatura = np.zeros((7, 5, 10, 15)) #7 dias, 5 níveis de altitude, 10 latitudes, 15  

longitudes  

#Simula preenchimento com dados reais  

for dia in range(7):  

    for alt in range(5):  

        for lat in range(10):  

            for lon in range(15):  

                #Temperatura simulada em função da altitude (mais frio em altitudes maiores)  

                Temperatura[dia][alt][lat][lon]=30-(alt*6)+np.random.uniform(-2, 2)  

#Acessa a temperatura no 3º dia, 2º nível de altitude, 4ª latitude e 10ª longitude  

valor = Temperatura [2][1][3][9]  

print(f"Temperatura prevista: {valor:.2f} °C")  

#Fim do programa "MatrizQuadridimensionalAplicada1" -----
-----
```

```

=====  

'''  

MatrizQuadridimensionalAplicada2.py  

-----  

Simula a aplicação de uma matriz tridimensional para simulação de concentração  

de poluentes.  

região dividida em:  

hora horas dos dia (00:00 a 23:59)  

alt altitudes (níveis da atmosfera)  

lat latitude (faixas: norte-sul)  

lon longitude (faixa: leste-oeste)  

-----  

'''  

import numpy as np  

#(horas, altitude, latitude, longitude)  

Poluicao = np.zeros((24, 4, 8, 12))  

for hora in range(24):  

    for alt in range(4):  

        for lat in range(8):  

            for lon in range(12):  

                #Poluição maior em baixas altitudes e horários de pico  

                Poluicao[hora][alt][lat][lon] = (  

                    100 - alt*15 + np.random.uniform(-5, 5)  

                )  

#Consulta: 18h, altitude 1, latitude 3, longitude 7  

nivel = Poluicao[18][1][3][7]  

print(f"Nível de poluição: {nivel:.1f} µg/m³")  

#Fim do programa "MatrizQuadridimensionalAplicada2" -----
-----
```

```

=====
'''  

MatrizQuadridimensionalAplicada3.py  

-----  

Simula a aplicação de uma matriz tridimensional para simulação de concentração  

de poluentes.  

região dividida em:  

mes meses do ano (0 a 11)  

prof profundidade (níveis da superfície)  

lat latitude (faixas: norte-sul)  

lon longitude (faixa: leste-oeste)  

Questão: Cada posição da matriz responde a pergunta: Qual é a temperatura da  

água neste mês, nesta profundidade, nesta latitude e nesta longitude?  

-----  

import numpy as np  

#(meses, profundidade, latitude, longitude)  

TempOceano = np.zeros((12, 6, 10, 10))  

for mes in range(12):  

    for prof in range(6):  

        for lat in range(10):  

            for lon in range(10):  

                #Água mais fria em maiores profundidades  

                TempOceano[mes][prof][lat][lon] = (  

                    25 - prof*3 + np.random.uniform(-1, 1)  

                )  

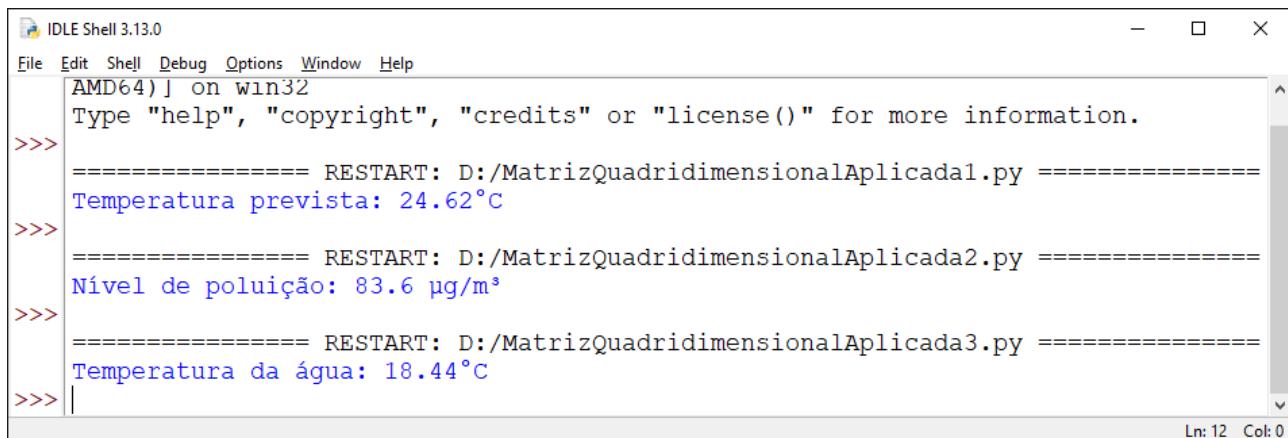
#Consulta: Julho (6), profundidade 2, lat 5, lon 4  

temp = TempOceano[6][2][5][4]  

print(f"Temperatura da água: {temp:.2f}°C")  

#Fim do programa "MatrizQuadridimensionalAplicada3" -----

```



The screenshot shows the IDLE Shell interface with the following content:

- IDLE Shell 3.13.0**
- File Edit Shell Debug Options Window Help**
- AMD64) J on win32**
- Type "help", "copyright", "credits" or "license()" for more information.**
- >>>** ===== RESTART: D:/MatrizQuadridimensionalAplicada1.py =====
- Temperatura prevista: 24.62°C**
- >>>** ===== RESTART: D:/MatrizQuadridimensionalAplicada2.py =====
- Nível de poluição: 83.6 µg/m³**
- >>>** ===== RESTART: D:/MatrizQuadridimensionalAplicada3.py =====
- Temperatura da água: 18.44°C**
- >>> |**
- Ln: 12 Col: 0**

Figura 1 - Saídas dos três programas: respectivamente