

4 Matrizes Unidimensionais

Neste capítulo você vai aprender como utilizar matrizes, também conhecidas como vetores, que são estruturas apropriadas para armazenar coleções de dados do mesmo tipo que são em geral processados da mesma maneira. Você também vai aprender como utilizar alguns métodos específicos para a manipulação de matrizes, utilizados para selecionar subgrupos de dados, unir informações e ordenar seus elementos, entre outras.

Matrizes são estruturas de dados que agregam muitas informações sob um mesmo identificador. As diferentes informações podem ser acessadas através de um índice que as seleciona. O exemplo abaixo cria duas matrizes unidimensionais, guarda nelas as posições x e y de 4 partículas em um plano e imprime os pontos no documento:

Listagem:

```
<script>
var x = [1,3,5,7];
var y = [-3,-1,1,3];
document.write(x[0]+ "," + y[0] + "<br>");
document.write(x[1]+ "," + y[1] + "<br>");
document.write(x[2]+ "," + y[2] + "<br>");
document.write(x[3]+ "," + y[3] + "<br>");
```

Resultado:

```
1,-3
3,-1
5,1
7,3
```

Em JavaScript, assim como em C, C++ e Java, o primeiro elemento de uma matriz tem índice 0, diferentemente do que acontece em FORTRAN (linguagem ainda muito utilizada em alguns campos de pesquisa em física), cujo primeiro elemento de uma matriz tem índice 1.

O mesmo resultado seria obtido primeiro antecipando que as variáveis x e y são matrizes (de tamanho indefinido) e depois atribuindo valores a cada um dos seus elementos individualmente:

Listagem:

```
<script>
var x = [];
var y = [];
x[0] = 1; y[0] = -3;
x[1] = 3; y[1] = -1;
x[2] = 5; y[2] = 1;
x[3] = 7; y[3] = 3;
for (var i=0; i<x.length; i++) {
    document.write(x[i] + "," + y[i] + "<br>");
}
</script>
```

No exemplo acima, a propriedade `length` (comprimento) da matriz foi utilizada. No caso, `x.length` vale 4, e o laço `for` é executado 4 vezes, para i valendo 0, 1, 2, e 3. Esta propriedade torna-se particularmente importante quando queremos colocar no script matrizes de tamanhos variáveis, pois não precisamos nos preocupar em tomar conta disto.

`var x = new Array(1,3,5,7)` também é uma maneira de criar e inicializar uma matriz, sendo equivalente a `var x = [1,3,5,7]`. Quando apenas um argumento numérico n é utilizado, por exemplo em `var x = new Array(n)`, é criada uma matriz com n elementos, todos indefinidos. Quando nenhum argumento é utilizado, por exemplo em `var x = new Array()`, é criada uma matriz de tamanho indefinido, sendo equivalente a `var x = []`.

Sua vez.. (4-1)

Modifique o exemplo anterior de modo que inclua uma terceira matriz z , correspondente à terceira coordenada espacial, inicializada com 2, 4, 6, 8. O script também deve modificar a forma de imprimir os resultados, agrupando as coordenadas em parênteses:

```
(1, -3, 2)
(3, -1, 4)
(5, 1, 6)
(7, 3, 8)
```

O exemplo a seguir utiliza matrizes para calcular a posição do centro de massa de um sistema unidimensional com 5 partículas posicionadas em diferentes pontos ao longo do eixo x . Para sistemas discretos, a coordenada x_{CM} do centro de massa é dada por:

$$x_{CM} = \frac{x_1 m_1 + x_2 m_2 + \dots + x_N m_N}{m_1 + m_2 + \dots + m_N} = \frac{\sum_{i=1}^N x_i m_i}{\sum_{i=1}^N m_i}$$

onde x_i e m_i são a coordenada e a massa de cada partícula.

Listagem:

```
<script>
var m = [3,2,1,4,5];
var x = [1,3,2,4,7];
var smx = 0;
var sm = 0;
for (i=0;i<m.length;i++) {
    smx = smx + x[i]*m[i];
    sm += m[i];
}
var xcm = smx/sm;
document.write(xcm.toFixed(2));
</script>
```

Resultado:

4.13

O laço `for` acumula a soma do produto da massa pela posição de cada partícula e a soma de suas massas. Um erro muito comum é esquecer de inicializar com zero os acumuladores, como são chamadas variáveis como `smx` e `sm` definidas neste exemplo.

A instrução `sm += m[i]` é uma abreviação para a instrução `sm = sm + m[i]`. Estruturas semelhantes podem ser construídas com os outros operadores aritméticos, como por exemplo `a *= b` para representar `a = a * b`.

Sua vez.. (4-2)

Modifique o exemplo anterior de modo que inclua valores para a coordenada y de cada partícula e calcule também a coordenada y_{cm} do centro de massa.

O exemplo a seguir calcula o produto escalar $\mathbf{a} \cdot \mathbf{b}$ entre os vetores \mathbf{a} e \mathbf{b} , que é dado por:

$$c = a_1 b_1 + a_2 b_2 + a_3 b_3$$

onde os índices 1, 2 e 3 representam as coordenadas x , y e z de cada vetor.

Listagem:

```

<script>
var a = [3,2,1];
var b = [1,2,3];

var prEsc = 0;

for (i=0;i<3;i++) {
    prEsc = prEsc + a[i] * b[i];
}

document.write("<b>a</b> &middot; <b>b</b> = ");
document.write(prEsc.toFixed(2));
</script>

```

Resultado:

a · b = 10.00

Neste exemplo as novidades estão relacionadas à formatação HTML, que vamos explorar em detalhes mais adiante. Os marcadores `...` delimitam regiões do texto que aparecerão em **negrito**; já a sequência `·` apresenta o símbolo ".", um dos muitos caracteres especiais disponíveis no HTML. Para adiantar, em matemática os mais comuns são `×`, que mostra "×", `<`, que mostra "<", `≤`, que mostra "≤", `>`, que mostra ">" e `≥`, que mostra "≥".

Note que tanto no exemplo do cálculo de centro de massa, apesar de na fórmula terem sido utilizados índices de 1 a N , nos scripts foram utilizados índices de 0 a $N - 1$.

O objeto `Array` dispõe de alguns métodos que podem ser úteis.

concat

O método `concat` retorna uma matriz que é a combinação de duas matrizes, "pendurando" (ou concatenando) uma à outra:

Listagem:

```

<script>
var quarks = ["up","down","charm","strange","bottom","top"];
var leptons = ["electron",&tau;,"muon",
              "eletron-neutrino","tau-neutrino","muon-neutrino"];
document.write(leptons.concat(quarks));
document.write("<br><br>");
var fermions = leptons.concat(quarks);
for (var i=0;i<fermions.length;i++) document.write(fermions[i] + " ");
</script>

```

Resultado:

electron,tau,muon,eletron-neutrino,tau-neutrino,muon-

```
neutrino,up,down,charm, strange, bottom,top
```

```
electron tau muon eletron-neutrino tau-neutrino muon-neutrino up down charm  
strange bottom top
```

A instrução `leptons.concat(quarks)` apenas `quarks` a `leptons` e retorna uma terceira matriz que não é armazenada, mas apenas impressa em bloco, com os elementos automaticamente separados por vírgulas. A instrução `var fermions = leptons.concat(quarks)` apenas a matriz `leptons` à matriz `quarks`, guardando o resultado em uma terceira matriz `fermions`. A instrução seguinte imprime os elementos de `fermions` um a um, separando-os por espaços em branco.

Note o uso da instrução `document.write("

")`, que faz uso de dois elementos HTML `
` para "pular" duas linhas no texto. O uso de elementos CSS (*Cascade Style Sheets*), que serão vistos superficialmente neste texto, pode substituir de maneira mais prática e elegante esse tipo de recurso. A internet é rica em materiais sobre o tema: informe-se!

Sua vez.. (4-3)

Modifique o exemplo anterior de modo que sejam criadas duas matrizes unidimensionais `aditivas` e `subtrativas`, a primeira inicializada com os valores "vermelho", "verde" e "azul" e a segunda inicializada com os valores "ciano", "amarelo" e "magenta", e crie e imprima a matriz `cores` que contém a concatenação das duas anteriores.

join

O método `join` retorna um literal que é a junção de todos os elementos da matriz. O método pode receber como parâmetro um literal que separará os elementos no literal resultante. No exemplo a seguir este parâmetro é um sinal de adição com espaços em branco antes e depois:

Listagem:

```
<script>  
var mat = [1,3,5,7,9];  
var soma = 0;  
for (var i=0;i<mat.length;i++) soma += mat[i];  
var str = mat.join(" + ") + " = " + soma;  
document.write(str);  
</script>
```

Resultado:

```
1 + 3 + 5 + 7 + 9 = 25
```

Sua vez.. (4-4)

Modifique o exemplo anterior de modo que apresente uma expressão para o produto dos elementos da matriz ($1 * 3 * 5 * 7 * 9 = 945$).

reverse

O método `reverse` inverte os elementos de uma matriz:

Listagem:

```
<script>
var mat1 = [1,3,5,7,9];
var mat2 = mat1.reverse();
for (var i in mat2) document.write(mat2[i] + " ");
</script>
```

Resultado:

```
9 7 5 3 1
```

Note o uso da expressão `for (var i in mat2) { ... }` que é uma alternativa ao uso de `for (var i=0;i<mat2.length;i++) { ... }` para percorrer os elementos de uma matriz, particularmente quando nem todos estão definidos.

slice

O método `slice` retorna um pedaço de uma matriz. Recebe dois parâmetros, um obrigatório (o índice do elemento que inicia a sequência a ser extraída) e outro opcional (o índice do elemento que sucede o último elemento a ser extraído), ambos com a contagem iniciada em 0 (zero). Caso o elemento final não seja fornecido, a extração se dá até o final da matriz. No exemplo a seguir, os índices inicial e final são passados quando se deseja extrair as variáveis associadas às coordenadas retangulares, mas apenas o índice inicial é passado quando se deseja extrair as variáveis associadas às coordenadas esféricas:

Listagem:

```
<script>
var coords = ["x","y","z","r","&theta;","&phi;"];
var rets = coords.slice(0,2);
var esfs = coords.slice(3);
document.write(rets)
document.write("<br>");
document.write(esfs);
</script>
```

Resultado:

x,y
r,θ,φ

Note o uso das *entidades* HTML `θ` e `φ` para produzir as letras gregas θ e φ. Faça uma busca por "HTML entities" na internet para ver a enorme gama de símbolos que podem ser produzidos desta (e de outras) maneiras.

Se o segundo parâmetro for negativo, representa um deslocamento a partir do final da matriz. No exemplo a seguir, todos os elementos de uma matriz são copiados para outra, exceto os dois últimos:

Listagem:

```
<script>  
var mat1 = [1,3,5,7,9];  
var mat2 = mat1.slice(0,-2);  
document.write(mat2);  
</script>
```

Resultado:

1,3,5

Sua vez... (4-5)

Modifique o exemplo anterior de modo que extraia e imprima os elementos intermediários (3,5,7) da matriz.

sort

O método `sort` ordena uma matriz:

Listagem:

```
<script>  
var mat1 = ["u","d","s","c","t","b"];  
var mat2 = mat1.sort();  
document.write(mat2);  
</script>
```

Resultado:

b,c,d,s,t,u

O método `sort`, como alguns outros, não necessariamente precisa colocar o resultado em outra matriz, podendo ser utilizado para ordenar uma matriz e deixar o resultado nela mesma:

Listagem:

```
<script>
var mat1 = ["u","d","s","c","t","b"];
mat1.sort();
document.write(mat1);
</script>
```

Resultado:

```
b,c,d,s,t,u
```

Sua vez... (4-6)

Faça um script que defina uma matriz com os 5 primeiros números pares e outra com os 5 primeiros números ímpares. A seguir, o script deve utilizar o método `concat()` para combinar as duas matrizes, o método `sort()` para ordená-las em ordem crescente e a instrução `document.write()` para imprimir a matriz resultante.

O método `sort()` sempre interpreta as informações armazenadas em uma matriz como texto e nunca como números, o que pode levar a resultados indesejáveis. Experimente criar e ordenar duas matrizes, por exemplo `var mat1 = [1,9,11,20,22,99]` e `var mat2 = ["1","9","11","20","22","99"]` e aplicar o método `sort()` em ambas e observe os resultados. Para mais detalhes e sugestões de usos, pesquise algo como "sort javascript" na internet.

Exercícios

1. Faça um script que defina uma matriz com a velocidade de um objeto em diferentes instantes, preencha seus elementos e compute e apresente a média entre eles:

$$v_{\text{med}} = (1/N) \sum_i v_i$$

2. Modifique o script anterior para obter também o desvio quadrático médio da velocidade, dado por:

$$\delta_{\text{rms}}(v, v_{\text{med}}) = (1/N) [\sum_i (v_i - v_{\text{med}})^2]^{1/2}$$

3. Dados dois vetores:

$$\mathbf{a} = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k} \quad \text{e} \quad \mathbf{b} = b_x \mathbf{i} + b_y \mathbf{j} + b_z \mathbf{k}$$

o vetor soma \mathbf{c} é dado por:

$$\mathbf{c} = (a_x + b_x) \mathbf{i} + (a_y + b_y) \mathbf{j} + (a_z + b_z) \mathbf{k}$$

Faça um script que utilize matrizes para representar e somar dois vetores, armazenando o resultado da soma em uma terceira matriz.

4. O produto vetorial é dado por:

$$c_x = a_y b_z - b_y a_z \quad c_y = a_z b_x - b_z a_x \quad c_z = a_x b_y - b_x a_y$$

Faça um script que utilize matrizes para realizar o produto vetorial entre dois vetores, armazenando o resultado em uma terceira matriz.

5. Faça um script que percorra uma matriz com N números inteiros e copie para outra matriz apenas os números ímpares.
6. Faça um script que preencha uma matriz com dez números aleatórios e imprima-os em ordem crescente.
7. Faça um script que preencha uma matriz com uma lista de 20 palavras relacionadas a ciência e tecnologia, coloque-as em ordem alfabética e imprima somente aquelas cuja primeira letra está entre o j (jota) e o n (ene), inclusive.