Octave para Cálculo Numérico - 2020
© Gustavo C. Buscaglia

ICMC - Sala 4-219, Ramal 738176, gustavo.buscaglia@gmail.com

Leitura: Capítulo 1 do texto ("O que não se pode ignorar").
Leitura adicional: Slides `intro_matlab.pdf`, do Prof. Afonso Paiva Neto.

Software: Instalar Octave ou Matlab.

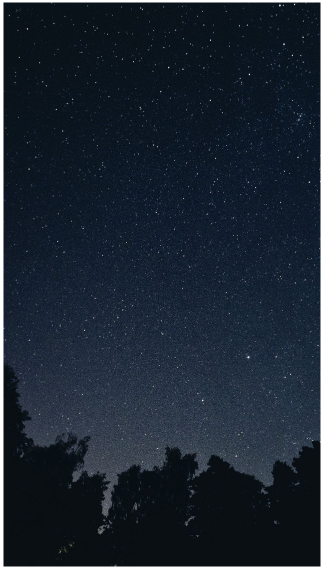
Exemplos com imagens e sequências

1 Operando sobre imagens

1. **Leitura e visualização:** Podemos ler uma matriz de um arquivo de texto usando os comandos `load` ou `dlmread`. A descrição dos comandos é obtida fazendo `help comando`. O arquivo `people.txt` disponível no site. A matriz corresponde a uma imagem, que podemos visualizar com o comando `imshow`.

```
matriz=load("people.txt");
imshow(matriz,[0 256])
```

   Veja os valores da matriz. Veja o que acontece se os limites são variados. Qual pixel da imagem corresponde ao elemento `matriz(1,1)`?

2. **Vetores de índices e operações ponto a ponto:** Usando o comando `size` vemos que a imagem de people.txt tem 625 linhas e 500 colunas. Outra imagem de 625×500 pode ser criada com os seguintes comandos

   ```
   i=1:625;
   j=1:500;
   ij=i'*j;
   ```

   Executar e explicar o resultado do seguinte código, vendo o efeito da multiplicação ponto-a-ponto.

   ```
   ij=ij/(500*625);
   imshow(ij);
   imshow(matriz.*ij,[0 256])
   ```

3. **Exercício:** Explique o resultado das seguintes operações

   - ```
     matriz=load("people.txt");
     matriz=matriz/256;
     imshow(matriz)
     matriz=matriz*1.3;
     imshow(matriz)
     matriz=matriz*1.3;
     imshow(matriz)
     imshow(matriz,[0 1.69]);
     ```
   - ```
     matriz=load("flor.txt");
     imshow(matriz,[0 256])
     matriz=matriz-100;
     imshow(matriz,[0 256])
     ```

4. **Estruturas de repetição:** Programar uma função com as seguintes especificações:

   ```
   function B = suaviz1(A)
   ```

   sendo que a matriz B deve ter, em cada pixel (célula), a média dos valores *das células correspondentes da matriz A e das células adjacentes em horizontal e em vertical*.

   É claro que, sendo A uma matriz $m \times n$, para $i, j > 1$ e $i < m$, $j < n$ a fórmula que define o valor de B_{ij} é

   $$B_{ij} = \frac{1}{5}\left(A_{ij} + A_{i-1,j} + A_{i+1,j} + A_{i,j-1} + A_{i,j+1}\right) \ .$$

   Que pode ser programado como

```
function B = suaviz1(A)
[m n]=size(A);
B=zeros(m,n);
for i=2:m-1
 for j=2:n-1
  B(i,j)=(A(i,j)+A(i-1,j)+A(i+1,j)+A(i,j-1)+A(i,j+1))/5;
 endfor
endfor
end
```

É muito útil e eficiente a versão “vetorizada” do anterior

```
function B = suaviz1(A)
[m n]=size(A);
B=zeros(m,n);
i=2:m-1;
j=2:n-1;
B(i,j)=(A(i,j)+A(i-1,j)+A(i+1,j)+A(i,j-1)+A(i,j+1))/5;
end
```

5. **Exercício:** Porém, a fórmula anterior não pode ser aplicada nas bordas da imagem. Com os códigos acima a borda vale zero. Proponha métodos simples de “completar” a matriz B e implemente. O objetivo é que a borda não destoe com o resto da imagem, mas também que a operação $A \mapsto B = S(A)$ seja linear.

6. **Exercício:** Programar `function B = suaviz2(A)`, onde a matriz B deve ter, em cada pixel (célula), a média dos valores *das células correspondentes da matriz A e das células adjacentes em horizontal, em vertical ou diagonalmente*. Tente vetorizar a função anterior.

7. Ver o efeito sobre a imagem people.txt de utilizar repetidamente as funções de suavização 1 e 2 anteriores.

8. **Discussão:** A operação de suavização descrita acima é linear, isto é,

$$S(\alpha A + \beta B) = \alpha S(A) + \beta S(B) \ .$$

A álgebra linear indica então que, em algum sentido que debemos descobrir, existe uma matriz M tal que $S(A) = M\,A$. Assim, se partirmos da imagem já suavizada $B = S(A)$, poderiamos recuperar A fazendo $M^{-1}B$. Como fazer isto?

Notar que uma matriz $m \times n$ em Octave também funciona como um vetor de mn componentes, armazenado coluna a coluna. Por exemplo, se A é matriz 4×3, então `A(2,2)` é o mesmo que `A(6)`.

9. **Operações lógicas ponto-a-ponto:** Visualizemos o arquivo flor.txt disponível no site. `A=load("flor.txt")`, `imshow(A,[0 256])`. Desejamos uma função

```
function B = fundopreto(A)
```

tal que o comando `imshow(B,[0 256])` mostre a mesma imagem da flor, mas com o fundo totalmente preto. A resposta não é muito fácil, mas podemos começar colocando a zero todos os pixels pouco iluminados, fazendo

```
fl=load("flor.txt");
fl2=fl.*(fl>90);
imshow(fl2)
```

Porém, isto bota preto em partes interiores da flor.

10. **Funções bidimensionais:** Os seguintes comandos

```
x=(0:800)/800;
y=(0:600)/800;
[xx yy]=meshgrid(x,y);
imshow(xx)
```

```
imshow(yy)
imshow(((xx-1/2).^2+(yy-3/8).^2)<.09)
```

mostra em branco um círculo centrado numa imagem de 800×600 pixels. Para isto, identifica os pixels onde a função $f(x, y) = (x - 1/2)^2 + (y - 3/8)^2$ é menor que 0.3^2. Tomar cuidado em que

```
> size(xx)
ans =
   601  801
```

e não `801 601`. De outra maneira o gráfico ficaria "transposto".

Para plotar $f(x, y)$ podemos fazer

```
zz=(xx-1/2).^2+(yy-3/8).^2;
imshow(zz,[min(min(zz)) max(max(zz))]) ## ou
contourf(xx,yy,zz) ## ou
surf(xx,yy,zz)
```

também podemos apenas plotar um subconjunto dos pontos gerando um subconjunto de índices:

```
i2=1:4:size(xx,1); #nesse caso 1 a cada 4
j2=1:4:size(xx,2);
surf(xx(i2,j2),yy(i2,j2),zz(i2,j2))
```

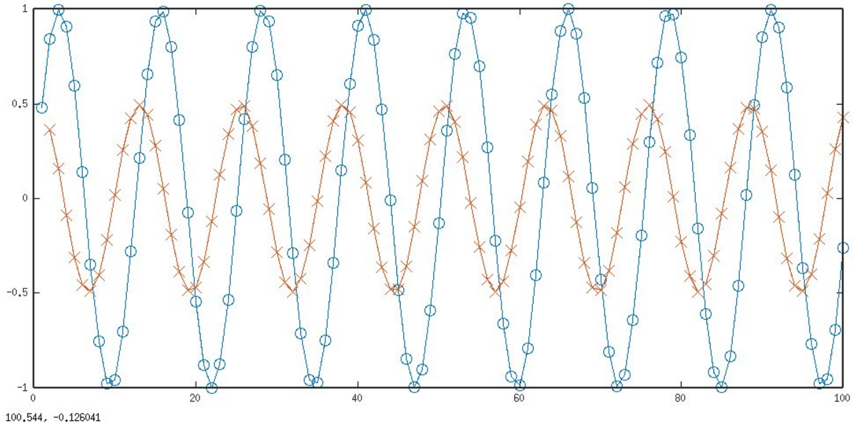
11. **Exercício:** Faça um código que desenhe uma elipse preta centrada sobre fundo cinza. O eixo maior da elipse deve medir 2 e o eixo menor 1. O eixo maior deve estar a 30 graus da horizontal. A imagem total deve corresponder ao retângulo $[0, 3] \times [0, 2]$.

2 Operando sobre sequências

1. É frequente precisarmos fazer operações sobre sequências $\{x^i\}$, que sendo sequências de números reais podemos representar por vetores do tipo `x(1:n,1)`, com mais colunas para o caso em que $x^i \in \mathbb{R}^d$.

2. Exemplo: $x^i = \sin k\, i$, $y^i = x^i - x^{i-1}$

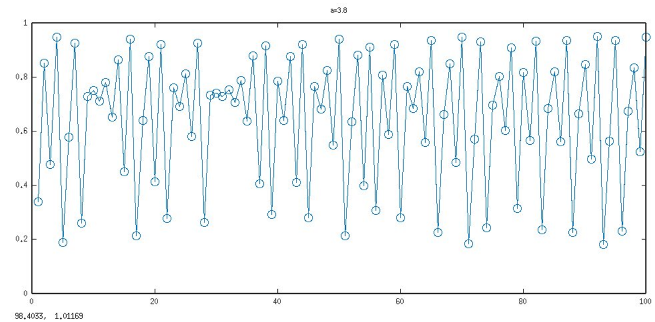
```
n=100;
ii=1:n;
k=.5;
xx=sin(k*ii);
yy=xx(2:n)-xx(1:n-1);
plot(ii,xx,"o-",ii(2:n),yy,"x-")
```



Nota: Os gráficos x-y em geral ficam melhor fazendo `graphics_toolkit(‘‘gnuplot’’)`. O default é `graphics_toolkit(‘‘fltk’’)`.

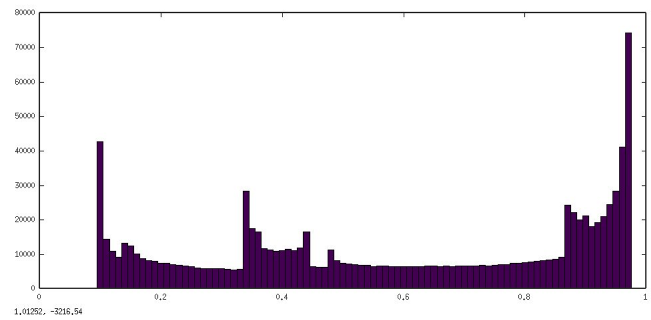
3. Exemplo (comando **hist**): $x^{n+1} = a\, x^n\, (1 - x^n)$. Esse é o **mapa logístico**. Variar a entre 0 e 4. O comportamento para $n \gg 1$ vai variando de constante, para periódico com um ou mais períodos, para caótico a partir de $a \simeq 3.6$.

```
n=10000; a=3.8; x(1)=0.1;
for i=2:n
 x(i)=a*x(i-1)*(1-x(i-1));
end
plot(x(9001:9100),"o-")
title("a=3.8")
```



A sequência gerada é bastante complexa, embora seja “determinística”. Podemos analizar vendo um histograma (ou densidade):

```
dx=0.01; bins=0:dx:1;
figure; hist(x,bins); axis([0 1])
## aproximando a densidade dos pontos
[NN XX]=hist(x,bins,1./dx);
figure; plot(XX,NN)
sum(NN*dx) ## deve ser 1
```



Facilmente podemos investigar diversos aspectos da **série temporal** x assim gerada (que depende de a e de x^1).

```
Nx=10000; a=3.9; x=zeros(1,Nx); x(1)=0.1;
for i=2:Nx
 x(i)=a*x(i-1)*(1-x(i-1));
end
```

- Média, $\overline{x} = \sum_{i=1}^{N_x} x^i / N_x$.

  ```
  xmean = mean(x)
  ```

  Os resultados para alguns valores de x^1 são ($a = 3.9$):

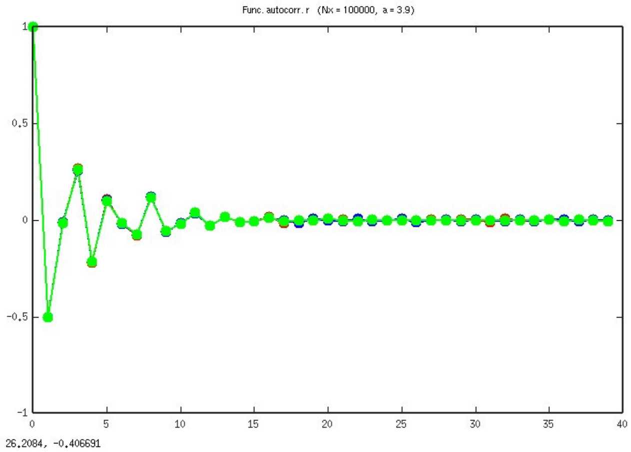
x^1	0.1	0.33	0.54	0.88
$\overline{x}$ ($N_x = 10$)	0.57548	0.58183	0.61336	0.66007
$\overline{x}$ ($N_x = 10^2$)	0.58976	0.59949	0.62498	0.60078
$\overline{x}$ ($N_x = 10^3$)	0.58860	0.59741	0.59028	0.59883
$\overline{x}$ ($N_x = 10^4$)	0.59445	0.59405	0.59300	0.59121
$\overline{x}$ ($N_x = 10^5$)	0.59276	0.59279	0.59231	0.59251

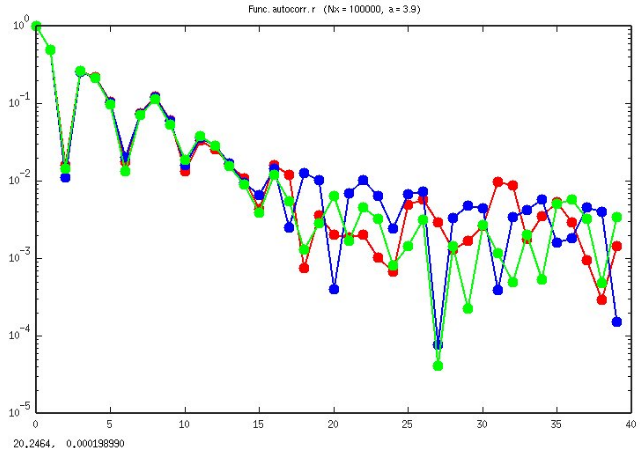
- Variança, $\mathrm{Var}(x) = \sum_i (x_i - \overline{x})^2 / (N_x - 1)$.

x^1	0.1	0.33	0.54	0.88
Var(x) ($N_x = 10^3$)	0.091245	0.087362	0.090611	0.086790
Var(x) ($N_x = 10^4$)	0.088648	0.088827	0.089312	0.090097
Var(x) ($N_x = 10^5$)	0.089405	0.089393	0.089606	0.089520

- O coeficiente de autocorrelação de x,

$$R(k) = \frac{1}{(N_x - k - 1)\mathrm{Var}(x)} \sum_{i=1}^{N_x - k} (x_i - \overline{x})(x_{i+k} - \overline{x})$$





4. **Exercício:**

   - Rodar o código "logistico.m" do site com vários valores de N_x. Dos resultados extrair conclusões sobre a série temporal quando $N_x \to +\infty$.
   - Modificar o código para analizar a série temporal que se encontra na segunda coluna do arquivo "temperatura.dat" do site. Trata-se de 8402 dados de temperatura numa cidade da Argentina. Deveria obter resultados semelhantes aos da figura. Tirar alguma conclusão da análise.