

Introdução aos Métodos Numéricos com Octave

Dia 1

Márcio Antônio de Andrade Bortoloti

I Jornada de Mineração / UNEB

Departamento de Ciências Exatas e Tecnológicas - DCET

Universidade Estadual do Sudoeste da Bahia

mbortoloti@uesb.edu.br

www2.uesb.br/professor/mbortoloti

Sumário

Apresentação

O Octave

Matrizes

Expressões

Resolução de Sistemas, Autovalores e Autovetores

Operações elemento por elemento, relacionais e conectivos lógicos

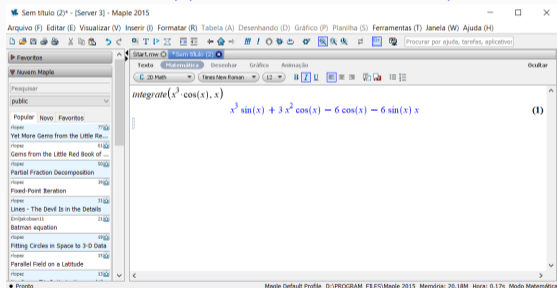
Gráficos

Apresentação

- **Computação Algébrica**

A computação algébrica é a área da computação que lida com a manipulação e solução exata de equações.

Por exemplo: Maple, veja <https://www.maplesoft.com/>



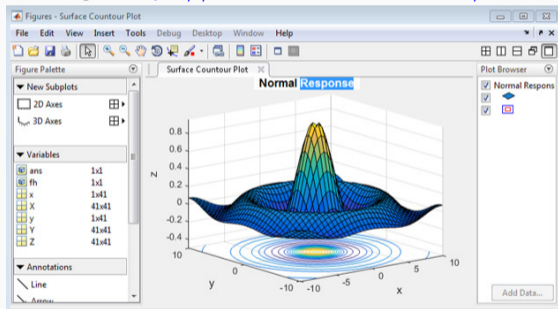
- **Vantagem:** Cálculos são feitos de forma exata;
- **Desvantagem:** Alto custo computacional.

Introdução

- **Computação Numérica**

A computação numérica é ...

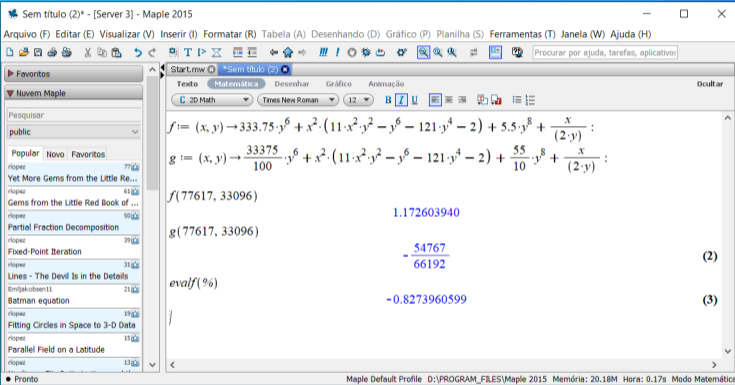
Por exemplo: Matlab, veja <https://www.mathworks.com/>



- **Vantagem:** Baixo custo computacional relativo à comp. algébrica.
- **Desvantagem:** Cálculos **não** são feitos de forma exata.

Cálculo Algébrico × Cálculo Numérico: um exemplo ...

No Maple,



The screenshot shows the Maple 2015 interface with the following content:

Arquivo (F) Editar (E) Visualizar (V) Inserir (I) Formatar (R) Tabela (A) Desenhando (D) Gráfico (P) Planilha (S) Ferramentas (T) Janela (W) Ajuda (H)

2D Math Times New Roman 12 B I U

$$f := (x, y) \rightarrow 333.75 \cdot y^6 + x^2 \cdot (11 \cdot x^2 \cdot y^2 - y^6 - 121 \cdot y^4 - 2) + 5.5 \cdot y^8 + \frac{x}{(2 \cdot y)}$$
$$g := (x, y) \rightarrow \frac{33375}{100} \cdot y^6 + x^2 \cdot (11 \cdot x^2 \cdot y^2 - y^6 - 121 \cdot y^4 - 2) + \frac{55}{10} \cdot y^8 + \frac{x}{(2 \cdot y)}$$

$f(77617, 33096)$ 1.172603940

$g(77617, 33096)$ $-\frac{54767}{66192}$ (2)

$evalf(\%)$ -0.8273960599 (3)

Maple Default Profile D:\PROGRAM_FILES\Maple 2015 Memória: 20.18M Hora: 0.17s Modo Matemática



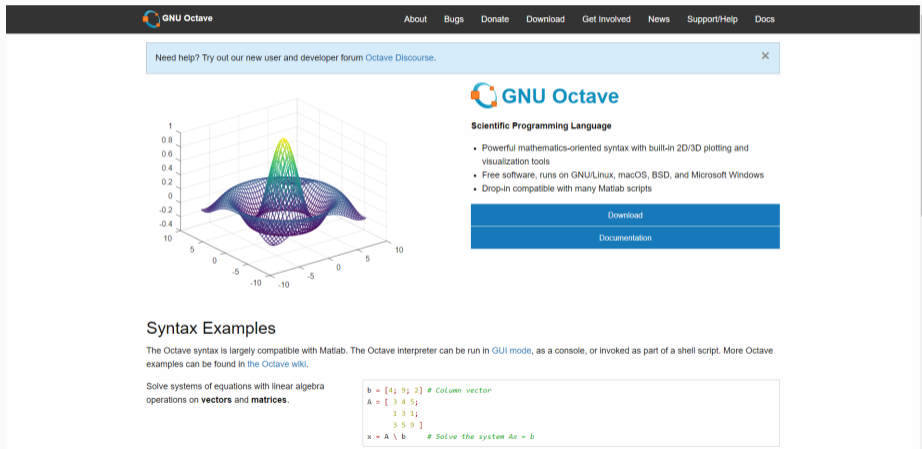
Atkinson, Kendall E.,

An Introduction to Numerical Analysis.

John Wiley and Sons, 1988.

0 Octave

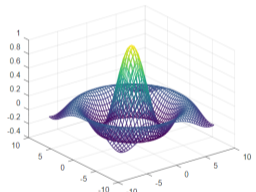
<https://www.gnu.org/software/octave/index>



GNU Octave

About Bugs Donate Download Get Involved News Support/Help Docs

Need help? Try out our new user and developer forum [Octave Discourse](#).



GNU Octave

Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in 2D/3D plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Microsoft Windows
- Drop-in compatible with many Matlab scripts

Download

Documentation

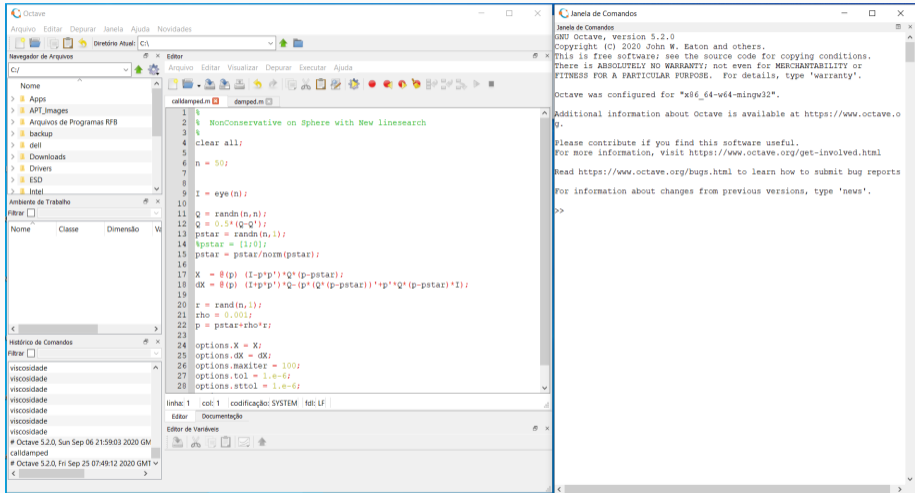
Syntax Examples

The Octave syntax is largely compatible with Matlab. The Octave interpreter can be run in [GUI mode](#), as a console, or invoked as part of a shell script. More Octave examples can be found in [the Octave wiki](#).

Solve systems of equations with linear algebra operations on **vectors** and **matrices**.

```
b = [4; 9; 2] # Column vector
A = [ 3 4 5;
      1 3 1;
      3 5 9 ]
x = A \ b # Solve the system Ax = b
```

Introdução



The screenshot displays the GNU Octave environment. The main window is titled 'Octave' and contains a file explorer on the left, a script editor in the center, and a command window on the right.

Script Editor Content:

```
1 %  
2 % NonConservative on Sphere with New lineearch  
3 %  
4 clear all;  
5  
6 n = 50;  
7  
8  
9 I = eye(n);  
10  
11 Q = randn(n,n);  
12 Q = 0.5*(Q+Q');  
13 pstar = randn(n,1);  
14 %pstar = [1;0];  
15 pstar = pstar/norm(pstar);  
16  
17 X = @(p) (I-p*p')*Q*(p-pstar);  
18 dX = @(p) (I+p*p')*Q-(p'(Q*(p-pstar))+p'*Q*(p-pstar))*I);  
19  
20 r = rand(n,1);  
21 rho = 0.001;  
22 p = pstar+rho*r;  
23  
24 options.X = X;  
25 options.dX = dX;  
26 options.maxiter = 100;  
27 options.tol = 1.e-6;  
28 options.sttol = 1.e-6;
```

Command Window Content:

```
Janela de Comandos  
GNU Octave, version 5.2.0  
Copyright (C) 2020 John W. Eaton and others.  
This is free software; see the source code for copying conditions.  
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.  
  
Octave was configured for "x86_64-w64-mingw32".  
  
Additional information about Octave is available at https://www.octave.org.  
  
Please contribute if you find this software useful.  
For more information, visit https://www.octave.org/get-involved.html  
  
Read https://www.octave.org/bugs.html to learn how to submit bug reports  
  
For information about changes from previous versions, type 'news'.  
  
>>
```

Download: <https://www.gnu.org/software/octave/index>

Matrices

1. O Octave trabalha com um tipo de variável: matrizes (um escalar é uma matrix 1×1);
2. Não é necessário declarar as variáveis para iniciá-las. Isto é feito automaticamente.

```
a = 1.8
```

```
b = 2+3i
```

Comandos Básicos - Matrizes

Entrada:

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

O Octave mostra a matrix da seguinte forma:

```
A =  
  16   3   2  13  
   5  10  11   8  
   9   6   7  12  
   4  15  14   1
```

Transposta de uma Matriz

O Comando

`A'`

Produz a saída

`ans =`

```
16    5    9    4
 3   10    6   15
 2   11    7   14
13    8   12    1
```

A Diagonal de uma Matriz

O Comando

`diag(A)`

Produz a saída

`ans =`

```
16
10
 7
 1
```

Subscripts

O acesso a uma entrada de uma matriz pode ser feito da seguinte forma:

$A(i,j)$

Exemplo:

$A(1,4) + A(2,4) + A(3,4) + A(4,4)$

ans =

34

Se algum índice da matriz for extrapolado um erro será informado:

$t = A(4,5)$

Index exceeds matrix dimensions

Por outro lado é possível armazenar um valor em um elemento "fora" de uma matriz. Esta, nesse caso será expandida. Exemplo:

$X = A;$

$X(4,5) = 17$

$X =$

16 3 2 13 0

5 10 11 8 0

9 6 7 12 0

4 15 14 1 17

O Operador "dois pontos" :

- `1:10` → 1 2 3 4 5 6 7 8 9 10
- `100:-7:50` → 100 93 86 79 72 65 58 51
- `0:pi/4:pi` → 0.00000 0.78540 1.57080 2.35619 3.14159
- `A(1:k,j)`

A função Magic

```
B = magic(3)
```

```
B =
```

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

Como um exercício, mostre que se a matriz B for dada pelo comando `magic(n)` então a referida soma é dada por $(n^3 + n)/2$.

Trabalhando com Matrizes

Algumas matrizes que são bastante utilizadas

`Z = zeros(2,4)`

```
Z =  0  0  0  0
     0  0  0  0
```

`F = 5*ones(3,3)`

```
F =
 5  5  5
 5  5  5
 5  5  5
```

`R = randn(4,4)`

```
R =
 1.0668  0.2944 -0.6918 -1.4410
 0.0593 -1.3362  0.8580  0.5711
-0.0956  0.7143  1.2540 -0.3999
-0.8323  1.6236 -1.5937  0.6900
```

Gerando Vetores

```
>> x = 0:0.2:3;  
>> y = exp(-x)+sin(x)  
>> z = [x' y']
```

Criamos uma matriz onde a primeira coluna pode ser vista como o domínio e a segunda como os valores de imagem da função $f(x) = e^{-x} + \text{sen}(x)$.

Matrizes a partir de matrizes

```
>> A = [ 1 2 3;4 5 6;7 8 9];  
>> b = [0 1 1];  
>> B = [A;b]
```

```
B =  
1   2   3  
4   5   6  
7   8   9  
0   1   1
```

Gerando Matrizes

```
>> A = [1 2 3 4; 5 6 7 8 ; 1 0 0 0;0 1 1 1];
```

```
>> B = A(2,[1 3])
```

```
B =
```

```
5    7
```

```
>> C=A(1:3,2:3)
```

```
C =
```

```
2    3
```

```
6    7
```

```
0    0
```

Atenção com a compatibilidade entre as matrizes

```
>> A = rand(3,3);
```

```
>> B = eye(3);
```

```
>> b = [1 1 1];
```

```
>> S = A+B;
```

```
>> C = A*B;
```

```
>> x = A*b';
```

Execute os seguintes comandos e interprete os resultados:

- `>> 2500/20`
- `>> 2500/20;`
- `>> b = [1 2 3 4 5 6 7 8 9]`
- `>> c = [1 2 3; 4 5 6; 7 8 9]`
- `>> c = [c; [10 11 12]]`
- `>> c(2,1)`
- `>> [m,n] = size(b)`
- `>> [m,n] = size(c)`

Execute os seguintes comandos e interprete os resultados:

- `>> dir`
- `>> a = 2^3;`
- `>> a = 4/3`
- `>> format long`
- `>> a = 4/3`
- `>> format short`
- `>> clear`
- `>> a = [1 2 3; 4 5 6;7 8 9];`
- `>> b = a'`
- `>> c = a+b`

Exercícios

Execute os seguintes comandos e interprete os resultados:

- `>> a(1,:) = [1 3 5]`
- `>> c = a(:,2)`
- `>> c = a(2:3,2:3)`
- `>> x = [-1 0 2];`
- `>> y = [-5 2 1];`
- `>> x*y`
- `>> c = x + 2`
- `>> a = [1 0 2; 0 3 4; 5 6 0];`
- `>> size(a)`
- `>> b = inv(a);`
- `>> c = b*a`
- `>> c = b/a`
- `>> c = b\a`
- `>> clear a b x y`
- `>> whos`

Expressões

- Variáveis
- Operadores
- Funções

- Nenhum tipo de declaração é requerido
- Quando o Octave encontra um novo nome de uma variável ele automaticamente cria a variável e aloca seu conteúdo.
- Se a variável já existe o Octave altera seu conteúdo.
- `a = 1`

+	Adição	$2+3$
-	Subtração	$5-2$
*	Multiplicação	$4*3.5$
/	Divisão	$3/2$
\	Divisão "esquerda"	$3 \setminus 2$
^	Potência	$2^{(-1)}$
'	Transposta conjugada complexa	A'
()	Especificador de avaliação de ordem	$2*(3+\cos(\pi))$

Formatos de saída

Comando	Saída	Comentários
format long	33.5000000000000000	16 dígitos
format short e	3.3500e + 001	5 dígitos mais expoente
format long e	33.5000000000000000e + 01	16 dígitos mais expoente
format bank	33.50	2 dígitos decimais
format rat	67/2	racional
format short	33.5000	4 dígitos decimais

Constantes

ans	Resposta mais recente, que não foi atribuída a nenhuma variável
eps	Precisão da máquina
realmax	Maior número de ponto flutuante
realmin	Menor número de ponto flutuante
pi	3.141592653589793
i,j	Unidade imaginária
NaN	Not a Number (indeterminação)
inf	Infinito
computer	tipo de computador
version	Versão do Octave

O Octave possui algumas funções básicas já implementadas, por exemplo,

- `sin(x)`, `cos(x)`, ...
- `log(x)`, `exp(x)`, ...
- `fprintf(ifile,"Hello, Octave!");`

```
>> rho = (1+sqrt(5))/2
>> rho =
>> 1.6180
```

Também é possível que o usuário defina suas próprias funções, por exemplo,

```
function v = volume(a,b,c)
    v = a*b*c;
end
x = volume(1,2,3);
```

Algumas funções do Octave

sin	seno		
cos	cosseno		
tan	tangente		
asin	arcoseno	max	maior componente de um vetor
acos	arcocosseno	min	menor componente de um vetor
atan	arcotangente	size	dimensões de uma matriz
exp	exponencial	sum	soma das componentes de um vetor
log	logaritmo natural	prod	produto das componentes de um vetor
sqrt	raiz quadrada	norm	norma euclidiana de vetor
rem	resto	sort	ordena as componentes de um vetor em ordem crescente
sign	sinal		
abs	valor absoluto		

Apagando o Workspace

- O comando `clear` + "Nome da Variável" apaga a variável do workspace.
- O comando `clear` apaga todas as variáveis do workspace.

```
>> clear
```

```
>> A=[1 1;2 1];
```

```
>> B=[0 0;1 1];
```

```
>> clear A
```

Resolução de Sistemas, Autovalores e Autovetores

Resolução de Sistemas $Ax = b$

```
>> A = [ 1 2 3; 4 5 6; 7 8 9];  
>> b = [1 1 1]';  
>> x = A\b  
x =  
   -5.0000e-01  
    1.1102e-16  
    5.0000e-01  
>> A*x  
ans =  
    1.00000  
    1.00000  
    1.00000
```

Se a matriz A for inversível então

$$x = A^{-1}b$$

```
>> x = inv(A)*b;
```

```
x =  
   -5.0000e-01  
    1.1102e-16  
    5.0000e-01
```

Obs: O comando `rank(A)` retorna o posto de A .

Fatoração LU

Considere o sistema $Ax = b$.

Se conseguirmos escrever $A = CD$ então o sistema passa a ser escrito como $CDx = b$.

Se fizermos $y = Dx$ teremos $Cy = b$.

Assim resolver o sistema $Ax = b$ é equivalente a resolver o sistema

$$\begin{cases} Cy = b \\ Dx = y \end{cases}$$

A decomposição LU da matriz A é dada por $A = LU$ onde L é uma matriz triangular inferior e U uma matriz triangular superior. Assim,

$$\begin{cases} Uy = b \\ Lx = y \end{cases}$$

Decomposição LU

```
>> A =[1 2 3;6 7 9; 2 5 7];
```

```
L =
```

```
1.00000 0.00000 0.00000
```

```
0.33333 1.00000 0.00000
```

```
0.16667 0.31250 1.00000
```

```
U =
```

```
6.00000 7.00000 9.00000
```

```
0.00000 2.66667 4.00000
```

```
0.00000 0.00000 0.25000
```

```
P =
```

```
Permutation Matrix
```

```
0 1 0
```

```
0 0 1
```

```
1 0 0
```

$$PA = LU$$

Autovalores e Autovetores

O comando $[V,D] = \text{eig}(A)$ retorna duas matrizes V e D que contem os autovetores de A e os autovalores de A , respectivamente.

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

```
>> [V,D]=eig(A)
```

```
V =
```

```
-0.231971    -0.785830    0.408248  
-0.525322    -0.086751   -0.816497  
-0.818673     0.612328    0.408248
```

```
D =
```

```
1.6117e+01         0         0  
0         -1.1168e+00         0  
0         0         -1.3037e-15
```

Lembre-se que, dada uma matriz A , dizemos que v é um autovetor de A e λ um autovalor de A associado a v , se $Av = \lambda v$.

Operações elemento por elemento, relacionais e conectivos lógicos

Operações elemento por elemento

Símbolo	Operação
.*	multiplicação
./ ou .\ .\	divisão
.^	potenciação

```
>> v = [ 1 1 2 3 4];
```

```
>> v.^2
```

```
ans =
```

```
1     1     4     9    16
```

Operadores relacionados e conectivos lógicos

Símbolo	Descrição
<	menor que
>	maior que
<=	menor ou igual a
>=	maior ou igual a
==	igual
~=	diferente
&	e
	ou
~	não

```
>> 1+5 == 6
```

```
ans = 1
```

```
>> 1+5 == 8
```

```
ans = 0
```

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

```
>> A > 4
```

```
ans =
```

```
0 0 0
```

```
0 1 1
```

```
1 1 1
```

Operadores relacionados e conectivos lógicos

```
>> B = [1 2 3 4];
```

```
>> any(B>3)
```

```
ans = 1
```

```
>> any(B>6)
```

```
ans = 0
```

```
>> any(B>3)
```

```
ans = 1
```

```
>> any(B>6)
```

```
ans = 0
```

```
>> all(B>2)
```

```
ans = 0
```

Gráficos

A função `plot` pode receber um número variável de argumentos.

Sua forma mais geral é

`plot(x1, y1, formato1, x2, y2, formato2, ···, xn, yn, formaton)`

Desse modo é possível traçar mais de uma curva no mesmo gráfico.

Gráficos de Funções

Tabela de "Cores"

Símbolo	Cor do Gráfico
y	amarela
m	roxa
c	azul claro
r	vermelha
g	verde
b	azul
w	branca
k	preta

Formatos de Pontos/linhas

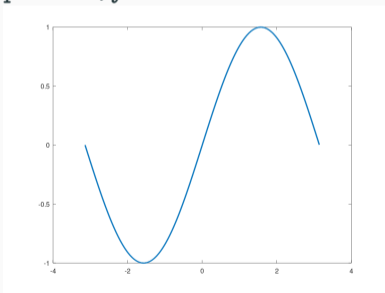
Símbolo	Formato
--	tracejado
.	ponto
o	círculo
×	×
+	cruz
-	sólida
*	estrela
:	pontilhada
-.	traço ponto

Outros Comandos

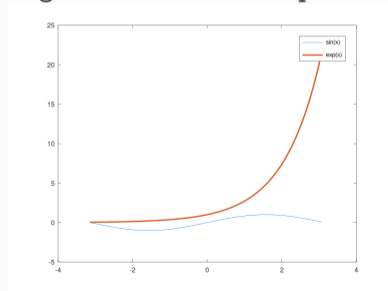
Comando	Descrição
title	título do gráfico
text	escr. no loc. espec.
semilogx	gráf. log em x
loglog	gráf. log
xlabel	nome do eixo x
ylabel	nome do eixo y
grid	des. linhas de grade
semilogy	gráf. log em y

Comando plot (plot(x,y))

```
x=-pi:0.01:pi;  
y = sin(x);  
plot(x,y)
```



```
x = -pi:0.1:pi;  
y1 = sin(x);  
y2 = exp(x);  
plot(x,y1,x,y2,'LineWidth',2)  
legend('sin(x)', 'exp(x)')
```



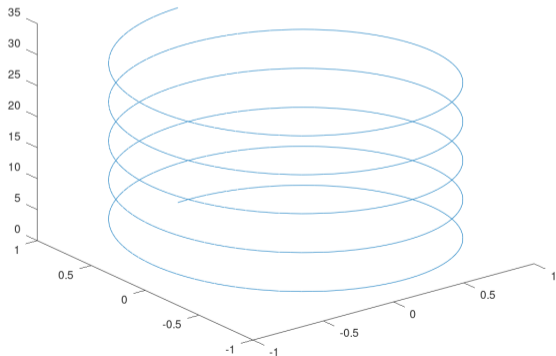
Gráficos 3D

Curvas		Superfícies	
plot3	Plota curvas em 3D	surf, surfc, surfl	superfícies 3D
contour	Curvas de nível	mesh, meshc, meshz	cria linhas em perspectiva 3D

Gráficos de Funções

Exemplo: Desenhe o gráfico da curva $z(t) = (\sin(t), \cos(t), t)$ (hélice circular), utilizando o comando plot3.

```
>> t = 0:pi/50:10*pi;  
>> plot3(sin(t),cos(t),t)
```



Para esboçar o gráfico de $z = f(x, y)$, primeiro devemos gerar matrizes X e Y contendo linhas e colunas repetidas, respectivamente, para funcionarem como o domínio da função.

A função **meshgrid** transforma o domínio especificado por dois vetores x e y em duas matrizes X e Y .

Essas matrizes então são usadas para avaliar a função de 2 variáveis.

Gráficos de Funções

```
tx = ty = linspace (-8, 8, 41)';  
[xx, yy] = meshgrid (tx, ty);  
r = sqrt (xx .^ 2 + yy .^ 2) + eps;  
tz = sin (r) ./ r;  
mesh (tx, ty, tz);  
xlabel ("tx");  
ylabel ("ty");  
zlabel ("tz");  
title ("3-D Sombrero plot");
```

