

# Gráficos 3D



Jeferson Souza (thejefecomp), Ph.D. Candidate  
[thejefecomp@neartword.com](mailto:thejefecomp@neartword.com)



# Introdução aos Gráficos 3D

## Introdução

No Scilab é possível, a partir de valores de vetores e matrizes, gerados ou não por funções, desenhar gráficos em três dimensões. Esses gráficos possibilitam uma melhor visualização dos dados em questão, a implicar em projeções tridimensionais resultantes da conjugação das diferentes malhas de pontos presentes em cada um dos planos  $x$ ,  $y$ , e  $z$ .

# Gráficos 3D

## Gráficos 3D

Existe um conjunto de funções que podem ser utilizadas para o desenho de gráficos tridimensionais no Scilab. Essas funções permitem desde a transformação de vetores em malhas de pontos (matrizes associadas a um determinado plano), até a própria projeção das referidas malhas em formato visual.

# A Função Utilitária `meshgrid()`

## A Função utilitária `meshgrid()`

A função utilitária `meshgrid()` permite a transformação de um conjunto de dois vetores (nos eixos  $x$  e  $y$ ) em matrizes. O intuito é permitir a geração das malhas de pontos nos referidos planos  $x$  e  $y$ , a implicar o seu uso na representação tridimensional que deseja ser visualizada.

# A Função Utilitária `meshgrid()`

## A Função utilitária `meshgrid()`

A sintaxe básica da função é a seguinte:

`[matrizX, matrizY] = meshgrid(vetorX, vetorY)`

Onde:

`matrizX` representa o `vetorX`, onde os valores de `vetorX` estão distribuídos nas linhas da `matrizX`, a representar a malha de pontos no plano presente no eixo x;

`matrizY` representa o `vetorY`, onde os valores de `vetorY` estão distribuídos nas colunas da `matrizY`, a representar a malha de pontos presente no plano do eixo y.

# A Função Utilitária `meshgrid()`

## Exemplo

```
vetorX = -1:0.1:1
```

```
vetorY = -1:0.1:1
```

```
[matrizX, matrizY] = meshgrid(vetorX, vetorY)
```

# A Função `mesh()`

## A Função `mesh()`

A função `mesh()` permite o desenho de malhas tridimensionais retangulares com base em conjuntos de pontos especificados por matrizes a representar a conjugação dos planos  $x$ ,  $y$ , e  $z$ . Caso somente a matriz de pontos do plano  $z$  seja especificada, a considerar a dependência intrínseca entre as diferentes dimensões, os planos  $x$  e  $y$  conterão os números de pontos equivalentes aos números de linhas e de colunas de  $z$ .

# A Função `mesh()`

## A Função `mesh()`

A sintaxe básica da função é a seguinte:

`mesh(<matrizX, matrizY,> matrizZ)`

Onde:

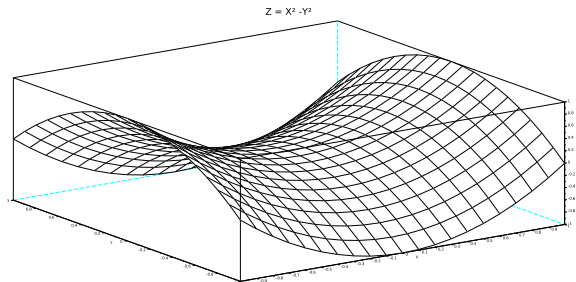
`matrizX` representa o conjunto de pontos no plano x [opcional];

`matrizY` representa o conjunto de pontos no plano y [opcional];

`matrizZ` representa o conjunto de pontos no plano z [obrigatório].

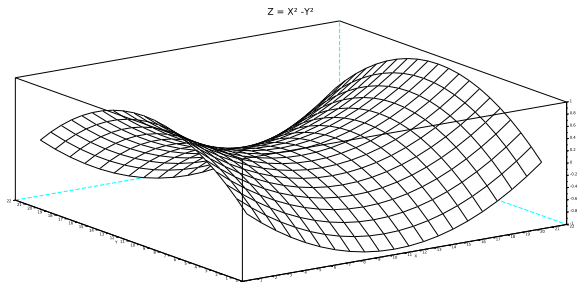


# A Função `mesh()` - Exemplo `mesh(matrizX, matrizY, matrizZ)` com $Z = X^2 - Y^2$



Malha a ser desenhada com `mesh(matrizX, matrizY, matrizZ)`

# A Função `mesh()` - Exemplo `mesh(matrizZ)` com $Z = X^2 - Y^2$



Malha a ser desenhada com `mesh(matrizZ)`

# A Função `plot3d()`

## A Função `plot3d()`

A função `plot3d()` permite o desenho de gráficos tridimensionais com base em conjuntos de pontos especificados por matrizes a permitir a representação gráfica nos planos  $x$ ,  $y$ , e  $Z$ . A função `plot3d()` também permite a visualização do gráfico em ângulos de observação distintos, os quais podem ser especificados por parâmetros adicionais.

# A Função `plot3d()`

## A Função `plot3d()`

A sintaxe básica da função é a seguinte:

```
plot3d(matrizX,matrizY,matrizZ, <parâmetros>)
```

Onde:

`matrizX` representa o conjunto de pontos no plano x;

`matrizY` representa o conjunto de pontos no plano y;

`matrizZ` representa o conjunto de pontos no plano z;

`<parâmetros>` representa o conjunto de parâmetros adicionais que podem ser especificados.

# A Função `plot3d()` - Exemplo `plot3d(matrizX, matrizY, matrizZ)`

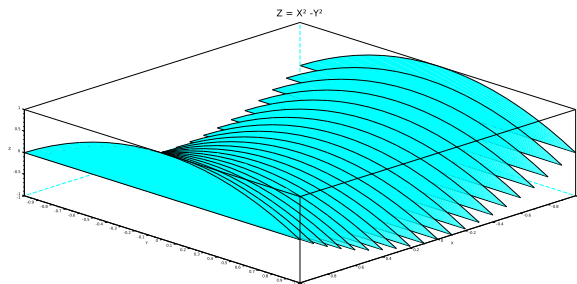


Gráfico tridimensional a ser desenhado com `plot3d(matrizX, matrizY, matrizZ)`

# A Função `plot3d()` - Exemplo `plot3d(matrizX, matrizY, matrizZ, alpha=30, theta=60)`

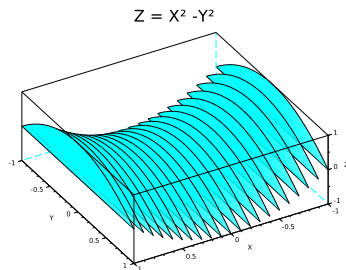
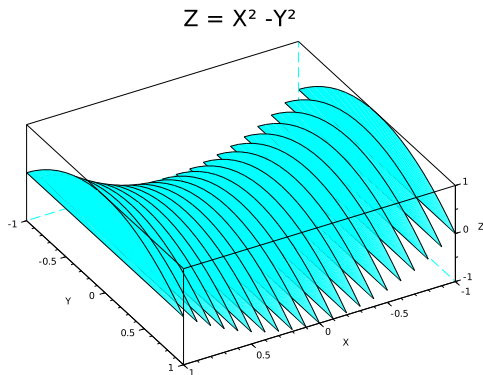


Gráfico tridimensional a ser desenhado com `plot3d(matrizX, matrizY, matrizZ, alpha=30, theta=60)`. O alpha representa o ângulo de observação no planos x e y, e o theta no plano z.

# A Função `plot3d()` - Exemplo `plot3d(matrizX, matrizY, matrizZ, alpha=30, theta=60)`: Sem Legenda



# A Função `plot3d()` - Exemplo

## `plot3d(vetorX,vetorY,matrizZ)`

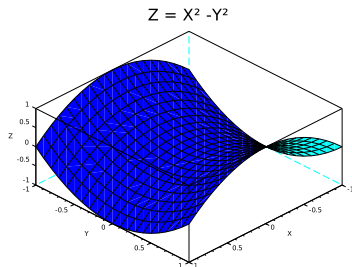
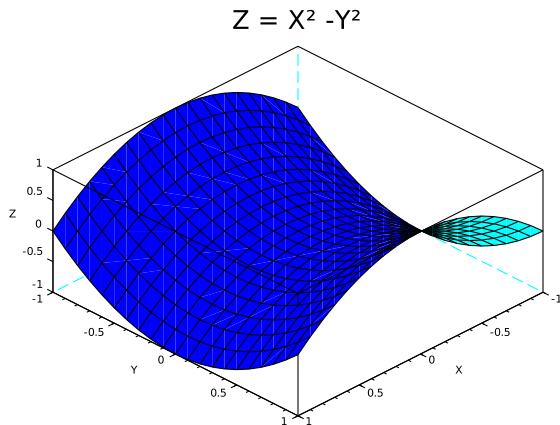


Gráfico tridimensional a ser desenhado com  
`plot3d(vetorX,vetorY,matrizZ)`



# A Função `plot3d()` - Exemplo

## `plot3d(vetorX,vetorY,matrixZ)`: Sem Legenda



## A Função `plot3d()` - Exemplo

`plot3d(vetorX,vetorY,matrizZ, alpha=30, theta=60)`

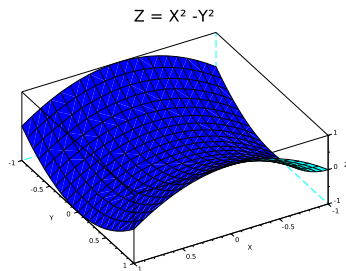
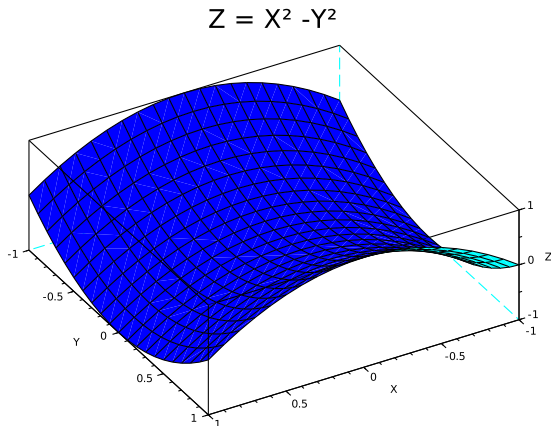


Gráfico tridimensional a ser desenhado com `plot3d(vetorX, vetorY, matrizZ, alpha=30, theta=60)`. O alpha representa o ângulo de observação nos planos x e y, e o theta no plano z.

# A Função `plot3d()` - Exemplo `plot3d(matrizZ, alpha=30, theta=60)`: Sem Legenda



# Gráficos 3D - Detalhes Importantes

## Alguns detalhes importantes

A função `subplot()` também pode ser utilizada com gráficos tridimensionais, a permitir o desenho de mais de um gráfico em uma mesma janela gráfica. Lembre-se que os valores das malhas de pontos nos diferentes planos definem o aspecto gráfico, a coniderar a conjugação das referidas malhas em uma representação gráfica unificada.

# Bibliografia



Gomez, C. and Scilab Enterprises. *"Scilab for very beginners"*, 2013.

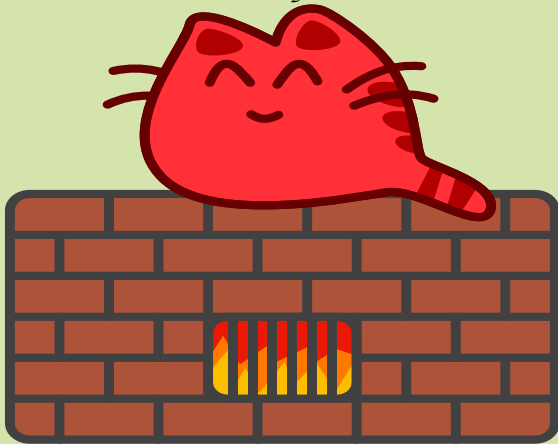


Kühlkamp, N. *"Matrizes e Sistema de Equações Lineares"*, 2ª edição, 2007.



Rietsch, E. *"An Introduction to Scilab from a Matlab User's Point of View"*. version 2.6-1.0. 2001-2002.

*That's it folks!*



*Thank you for your attention!*