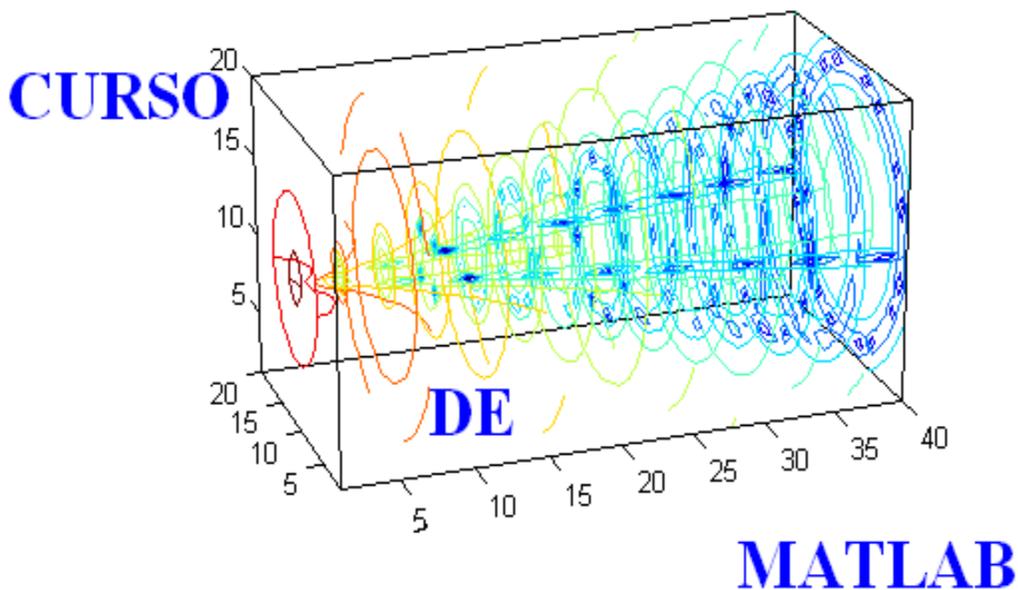


**UNIVERSIDADE FEDERAL DO PARÁ
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PROGRAMA DE EDUCAÇÃO TUTORIAL
SEMANA DOS 40 ANOS DE ENGENHARIA ELÉTRICA**



**NÍVEL BÁSICO
CAPÍTULO III**



**PROGRAMA DE EDUCAÇÃO TUTORIAL
ENGENHARIA ELÉTRICA -UFPA
BELÉM-2004**

CAPÍTULO 3 – GRÁFICOS NO MATLAB

3.1- INTRODUÇÃO:

O sistema gráfico do MATLAB incorpora uma variedade de técnicas sofisticadas para apresentação e visualização de dados. Este sistema é construído através de um conjunto de objetos gráficos, tais como linhas e superfícies, cuja presença pode ser controlada pelo ajuste dos valores das propriedades de cada objeto. Neste capítulo mostraremos os recursos disponíveis no MATLAB de criação e manipulação de figuras para apresentação de resultados em formato gráfico.

Engenheiros usam gráficos para analisar e resolver problemas e situações. Por isso é muito importante aprendermos a interpretar e gerar gráficos e suas formas. Vamos aprender como o MATLAB pode nos ajudar a gerar gráficos.

3.2 – GRÁFICOS 2-D

É muito comum engenheiros e cientistas usarem gráficos 2-D. Os dados que nós plotamos são usualmente lidos por um arquivo ou calculados em nossos programas. Geralmente assumimos que valores de x representam variáveis independentes e que valores de y representam variáveis dependentes. Os comandos para plotar gráficos lineares e logarítmico são:

Plot(x,y) - é uma função básica de geração de gráficos no MATLAB. Gera gráfico bidimensional com eixos x e y lineares.

Semilogx(x,y) – este comando gera gráfico usando escala linear para y e escala logarítmica para x.

Semilogy(x,y) – este comando gera gráfico usando escala linear para x e escala logarítmica para y.

Loglog(x,y) - este comando gera gráficos com escala logarítmica para ambos os eixos x e y.

Exemplo:

```
>>x=[-10:1:10];  
>>y=sin(x);  
>>plot(x,y)
```

⇒Detalhamento da função plot:

- **Gráficos múltiplos** ⇒ Para plotar curvas múltiplas no mesmo gráfico deve-se usar vários argumentos no comando plotar como a seguir:

Plot(x,y,w,z);

Quando se executa este comando a curva correspondente a x, y e a curva correspondente a w,z são plotados no mesmo gráfico. O MATLAB diferencia as curvas por cores diferentes.

- **Subplot** => Este comando é usado quando se quer visualizar dois ou mais gráficos ao mesmo tempo .

Subplot(2,1,1),plot(x,y)

Subplot(2,1,2),plot(w,z)

Este comando significa que teremos dois gráficos sendo o primeiro plot(x,y) colocado na parte superior da tela e o segundo plot(w,z) colocado na parte inferior da tela. Os comandos para adicionar títulos, linhas de grade e inserir textos estão relacionados a seguir:

Title ('texto') - Este comando escreve títulos no topo do gráfico plotado.

Xlabel('texto') - Este comando escreve um texto abaixo do eixo x do gráfico plotado.

Ylabel('texto') - Este comando escreve um texto ao lado do eixo y do gráfico plotado.

Text(x,y,'text') - Este comando escreve um texto na tela do gráfico no ponto específico das coordenadas (x,y) usando os eixos dos gráficos. Se x e y são vetores o texto é escrito a cada ponto.

Legend(texto1,...,texton) – Cria texto de legenda no gráfico.

Os comandos para fazer atributos de eixos estão relacionados a seguir :

Axis([Xmin Xmax Ymin Ymax]) - controla escala dos eixos.

Zoom(zfactor) – Modifica escala de exibição da janela.

Grid – habilita/desabilita exibição de grade.

Gráficos polares são úteis quando valores são representados por ângulo e grandeza (magnitude). Por exemplo se medirmos a intensidade luminosa ao redor de uma fonte de luz, podemos representar a informação com um ângulo fixando eixos e magnitude representando intensidade.

⇒ Coordenadas polares: Um ponto é representado por coordenadas polares por um ângulo θ e uma magnitude r . O valor de θ é geralmente dado entre 0 e 2π . A magnitude é um valor positivo que representa a distância do eixo que fornece o ângulo até o ponto.

O comando no MATLAB para gerar gráficos polares é:

Polar(θ , r) – Este comando generaliza gráficos polares com ângulo θ (em radiano) e magnitude r correspondente.

Exemplo:

```
>>theta=0:2*pi/100:2*pi;
>>r=theta/(2*pi);
>>polar(theta,r)
```

O MATLAB também trabalha com alguns tipos especiais de gráficos, tais como:

Bar(x,y) – Este comando gera gráficos de barras com elementos do vetor y localizados no vetor x, contém o mesmo espaço entre os vetores.

Stairs(x,y) – Este comando gera um gráfico de degraus com os elementos do vetor y localizados no vetor x, contendo o mesmo espaço entre os valores.

Stairs(y) - Este comando gera um gráfico de degraus com os elementos do vetor y.

⇒ Quando se trata de gráficos 2-D, é interessante citarmos a função:

comet(y) – Dados x e y, ele gera gráfico com exibição de trajetória animada.

3.3 – GRÁFICOS 3-D

Para gerar um arquivo que representa uma superfície 3-D, primeiramente calculamos os valores de x e y que representam as variáveis independentes e depois calculamos os valores de z que representa os valores da superfície. Nesta seção apresentaremos as principais funções de criação e manipulação de gráficos tridimensionais.

⇒ Tratando-se de dimensões espaciais começaremos com a função:

plot3(f(x,y),g(x,y)): Cria uma curva no espaço.

Exemplo:

```
>>x=-4*pi:0.1:4*pi;
>>y=-4*pi:0.1:4*pi;
>>plot3(cos(x),sin(y),(x+y))
```

O comando **meshgrid** tem os argumentos do vetor x e y., ou seja, transforma o domínio especificado pelos vetores x e y em vetores que podem ser usados em cálculos de função de duas variáveis e construção de gráfico 3D.

Meshgrid – dados dois vetores x e y, ele retorna duas matrizes, Mx e My, contendo a ‘malha’ de coordenadas necessária para geração de superfícies.

Exemplo:

```
>>x=1:3;
>>y=1:3;
>>[Mx,My]=meshgrid(x,y)
>>Mx =
     1     2     3
     1     2     3
     1     2     3
My =
     1     1     1
     2     2     2
     3     3     3
```

A função acima citada é utilizada para gerar dados para todas as outras funções 3D, tais como:

Mesh(Mx,My,Mz): Gera malha a partir de coordenadas 3D

Surf(Mx,My,Mz): gera superfície a partir de coordenadas 3D.

⇒ Gráficos 3 D Especiais:

cylinder(Raio,N): cria um cilindro de altura 1. Onde os parâmetros de entrada são o raio e o número de faces (N).

sphere(N): Cria uma esfera de raio 1. N é o número de faces.

⇒ Outras funções importantes acerca da criação de gráficos no MATLAB, podemos citamos as seguintes funções:

ezplot3(x(t),y(t),z(t)): Gera gráfico 3D a partir das equações paramétricas x(t), y(t) e z(t).

Exemplo:

```
>>ezplot3('cos(4*t)','sin(2*t)','1./(t+1)')
```

Ezcontour, ezcontourf, ezmesh, ezmeshc, ezsurf, ezsurf: Geram gráficos 3D a partir de equações de funções com duas variáveis.

Exemplo:

```
>>equação= 'cos(x)*y';  
>>subplot(2,3,1)  
>>ezcontour (equação)      % curva de nível de superfície  
>>subplot(2,3,2)  
>> ezcontourf (equação)    % area de nível de superfície  
>> subplot(2,3,3)  
>>ezmesh(equação)         % malha  
>> subplot(2,3,4)  
>>ezmeshc(equação)        % malha com curva de nível  
>> subplot(2,3,5)  
>>ezsurf(equação)         % superfície  
>> subplot(2,3,6)  
>>ezsurf(equação)         % superfície com curva de nível
```