



Caio Muller Santos Ribeiro
Engenheiro Eletricista | Sistemas de Potência

Com os
códigos!

Passo a passo para editar figuras no MATLAB®

Aprenda a editar figuras no software MATLAB®
para usá-las em publicações científicas.

Brasília, 2022



Sou graduado em Engenharia Elétrica pela Universidade Federal de Goiás (2016), mestre em Engenharia Elétrica na área de Sistemas de Potência pela Universidade de Brasília (2018). Atualmente, faço doutorado em Engenharia Elétrica na área de Sistemas de Potência com linha de pesquisa em Proteção de Sistemas de Potência pela Universidade de Brasília.

Caio Muller Santos Ribeiro



caiomullersribeiro



caiomullersribeiro.com

Dedico este *E-book*
ao meu pai e à minha mãe.

Sumário

Introdução	1
O que é MATLAB®?	2
Passo a passo para editar figuras no MATLAB®	3
Considerações Finais	30

Introdução

O processo de elaboração de uma publicação científica, por exemplo, um artigo, trabalho de conclusão de curso, dissertação de mestrado, tese de doutorado etc. envolve várias etapas. Dentre as quais, uma das mais importantes é a criação das figuras, em que se apresentam as ideias e os resultados da pesquisa.

De modo geral, as figuras em publicações científicas podem ser classificadas em duas categorias: ilustrativas e de resultados. Sendo a primeira formada por desenhos que visam auxiliar na apresentação de ideias. Por outro lado, a segunda categoria é formada, via de regra, por gráficos que são utilizados para apresentar os resultados obtidos no trabalho.

No que se refere à criação de figuras de resultados, existem diversos programas computacionais utilizados para esse fim, sendo o MATLAB® um dos mais utilizados. Milhões de engenheiros e cientistas utilizam esse software para analisar dados, desenvolver algoritmos, criar modelos e realizar testes. Ao final das análises, os resultados são, geralmente, apresentadas na forma de gráficos.

Todavia, observa-se que não é sempre que os autores realizam um processo de edição das figuras após criá-las, de forma que, os elementos visuais fiquem padronizados com o restante do documento, e por consequência, facilite o entendimento do leitor.

Neste *E-book*, apresenta-se um passo a passo sugerido para editar figuras no software MATLAB® e usá-las em publicações científicas. De tal forma a padronizar as figuras do documento, facilitar a interpretação do trabalho e, assim, aumentar a credibilidade da obra. Todo esse processo será detalhado com a apresentação dos códigos feitos no MATLAB®.

O que é MATLAB®?

É uma plataforma de programação e computação numérica. MATLAB® é um acrônimo para *MATrix LABoratory*. O software foi criado no fim dos anos 1970 pelo matemático e cientista da computação estadunidense Clever Moler. Sendo um programa computacional licenciado pela MathWorks Inc.

A linguagem de programação do MATLAB® é considerada de alto nível, isto é, mais próxima do entendimento humano. Os códigos que serão apresentados neste *E-book* foram feitos utilizando a versão MATLAB® R2021b, sendo a licença adquirida pelo autor. Dito isso, a seguir apresenta-se um exemplo com os 23 passos sugeridos para a edição de figuras no MATLAB®.



Figura 1: Logotipo do MATLAB®.

Fonte: [MathWorks Inc.](#)

Passo a passo para editar figuras no MATLAB®

Passo 1: Recomenda-se iniciar um código no MATLAB® aplicando alguns comandos, os quais limpam informações, caso existam, do *Command Window*, apagam variáveis salvas no *Workspace*, fecham todas as figuras abertas, e desativam todos os avisos que, porventura, venham a aparecer ao compilar o código.

% Iniciando o código

```
clc %limpa todo o texto do Command Window  
clear %remove todas as variáveis do Workspace  
close all %fecha todas as figuras com janelas visíveis  
warning('off') %desativa todos os avisos
```

Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/clc.html>

<https://www.mathworks.com/help/matlab/ref/clear.html>

<https://www.mathworks.com/help/matlab/ref/close.html>

<https://www.mathworks.com/help/matlab/ref/warning.html>

Passo 2: Você, provavelmente, já tem ou terá o sinal a ser plotado. Aqui, criaram-se sinais de tensão apenas para exemplificar os procedimentos sugeridos. A saber, define-se plotar como o ato de “imprimir” ou “desenhar” digitalmente um sinal em um determinado ambiente, conforme será apresentado neste *E-book*.

%% Criando o sinal a ser plotado

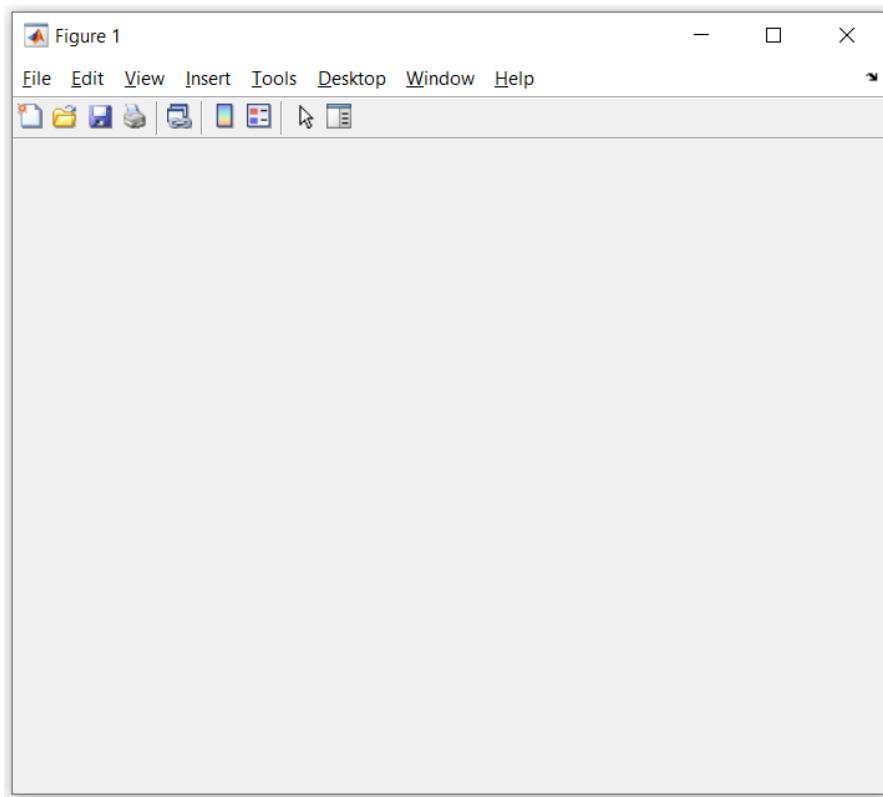
```
wt = 0:pi/100:4*pi; %frequência angular vezes tempo  
v1 = 1*sin(wt); %tensão 1  
v2 = 1*cos(wt); %tensão 2
```

Passo 3: Cria-se o ambiente da figura em que os sinais serão plotados.

%% Criando o ambiente da figura

```
h=figure; %cria o ambiente
```

Apenas para fins didáticos, apresentar-se-ão capturas de tela das imagens obtidas. Porém esse não é o procedimento recomendado para ser feito a fim de salvar a figura para utilizá-la em uma publicação científica. A forma mais adequada será apresentada no final deste *E-book*.



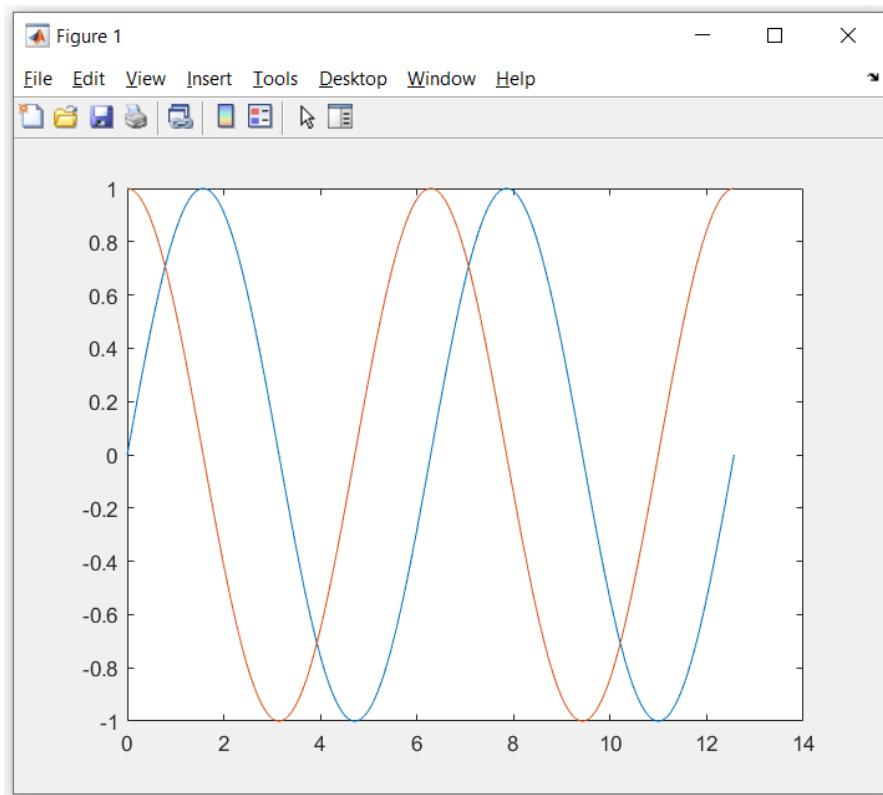
Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/figure.html>

Passo 4: O sinal criado no Passo 2 é plotado no ambiente criado no Passo 3.

% Plotando o gráfico

```
p=plot(wt,v1,... %plota o sinal da tensão 1  
wt,v2); %plota o sinal da tensão 2
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/plot.html>

Passo 5: Recebem-se as propriedades dos eixos e do ambiente da figura. Tal procedimento vai ser importante mais adiante. As variáveis `ax` e `fig` serão usadas para editar as propriedades da figura.

% Recebendo as propriedades

```
ax=gca; %recebe as propriedades dos eixos
```

```
fig=gcf; %recebe as propriedades da figura
```

Para mais informações, veja em:

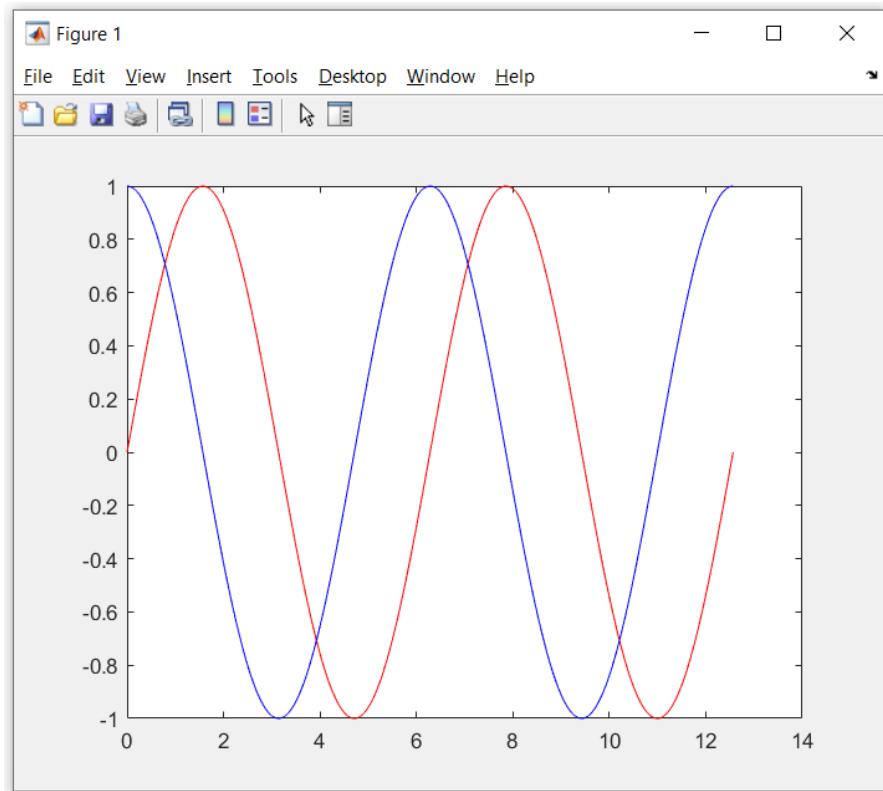
<https://www.mathworks.com/help/matlab/ref/gca.html>

[https://www.mathworks.com/help/matlab/ref\(gcf.html](https://www.mathworks.com/help/matlab/ref(gcf.html)

Passo 6: Alteram-se as cores das curvas. Recomenda-se usar cores que apresentem um bom contraste com o plano de fundo do documento, o qual, em geral, é na cor branca. Portanto, evita-se cores como o amarelo ‘yellow’. Lembre-se, a ausência de contraste pode dificultar a interpretação da imagem.

% Alterando as cores das curvas

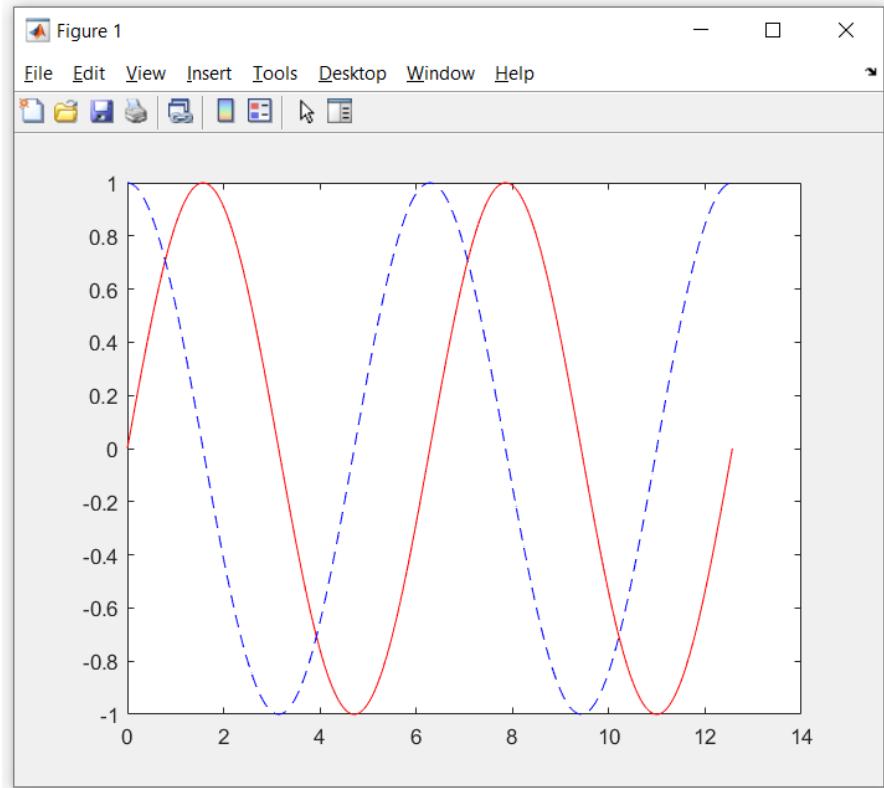
```
p(1).Color='red'; %define a cor vermelha na curva 1  
p(2).Color='blue'; %define a cor azul na curva 2
```



Passo 7: Altera-se o estilo da linha das curvas. Recomenda-se usar diferentes estilos de linhas (contínuo, tracejado, pontilhado ou tracejado-pontilhado) para cada curva. Tal procedimento cria outro fator adicional de diferenciação. Essa prática é importante, pois o leitor pode imprimir o trabalho em preto e branco, o que pode inviabilizar a diferenciação apenas por meio das cores.

% Alterando o estilo das curvas

```
p(1).LineStyle='-' ; %define a curva 1 como contínua  
p(2).LineStyle='--'; %define a curva 2 como tracejada
```



Para mais informações, veja em:

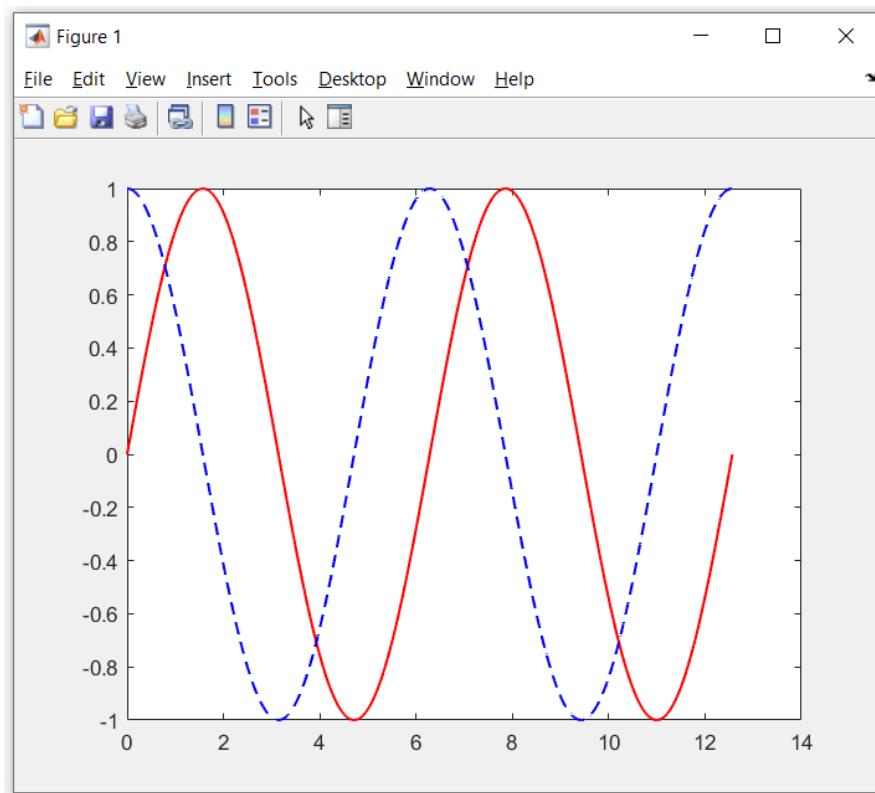
<https://www.mathworks.com/help/matlab/ref/plot.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.line-properties.html>

Passo 8: Pode-se alterar a espessura das linhas das curvas. Por padrão, essa espessura é igual a 0.5. Para o exemplo que está sendo apresentado aqui, escolheu-se um *LineWidth* igual a 1. Porém, escolha um valor de acordo com a figura que você está plotando. Cada caso é um caso.

% Alterando a espessura das linhas das curvas

```
p(1).LineWidth=1; %define a espessura da curva 1  
p(2).LineWidth=1; %define a espessura da curva 2
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/plot.html>

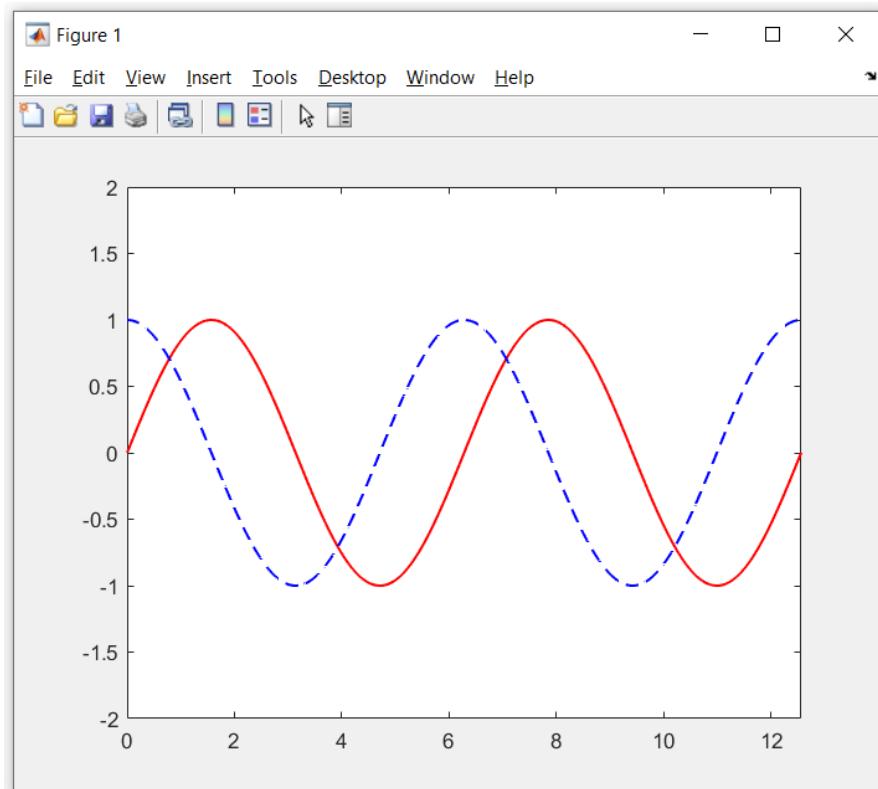
<https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.line-properties.html>

Passo 9: Recomenda-se definir os limites dos eixos x e y para que as curvas fiquem melhor posicionadas no ambiente da figura. Além disso, reserva-se um espaço para a colocação de uma legenda para o gráfico.

```
% Definindo os intervalos limites dos eixos
```

```
xlim([0 4*pi]); %intervalo limite do eixo x
```

```
ylim([-2 2]); %intervalo limite do eixo y
```



Para mais informações, veja em:

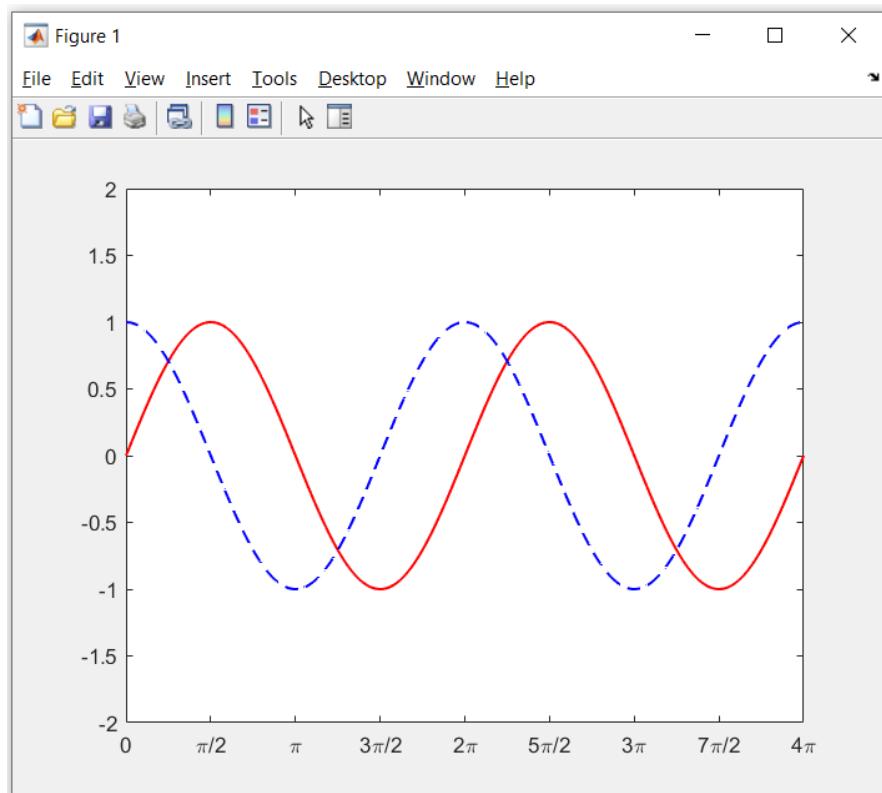
<https://www.mathworks.com/help/matlab/ref/xlim.html>

<https://www.mathworks.com/help/matlab/ref/ylim.html>

Passo 10: Pode-se alterar as marcações dos eixos. Tal procedimento é importante para auxiliar na interpretação do leitor.

% Definindo as marcações e rótulos dos eixos

```
ax.XTick=(0:pi/2:4*pi); %marcações do eixo x
ax.XTickLabel=({'0','\pi/2','\pi','3\pi/2','2\pi',...
    '5\pi/2','3\pi','7\pi/2','4\pi'}); %define os rótulos do eixo x
ax.YTick=(-2:0.5:2); %marcações do eixo y
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/xticks.html>

<https://www.mathworks.com/help/matlab/ref/yticks.html>

<https://www.mathworks.com/help/matlab/ref/xticklabels.html>

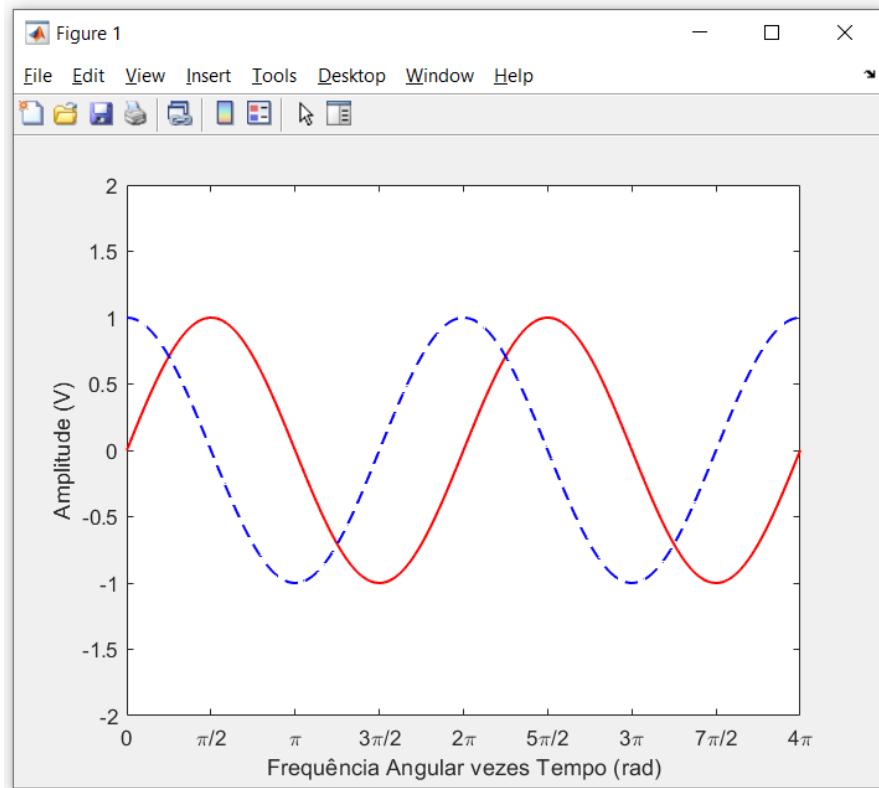
<https://www.mathworks.com/help/matlab/ref/yticklabels.html>

Passo 11: Definem-se os rótulos dos eixos do gráfico.

% Definindo os rótulos dos eixos

```
hx=xlabel('Frequência Angular vezes Tempo (rad)');
%define o rótulo do eixo x

hy ylabel('Amplitude (V)'); %define o rótulo do eixo y
```



Para mais informações, veja em:

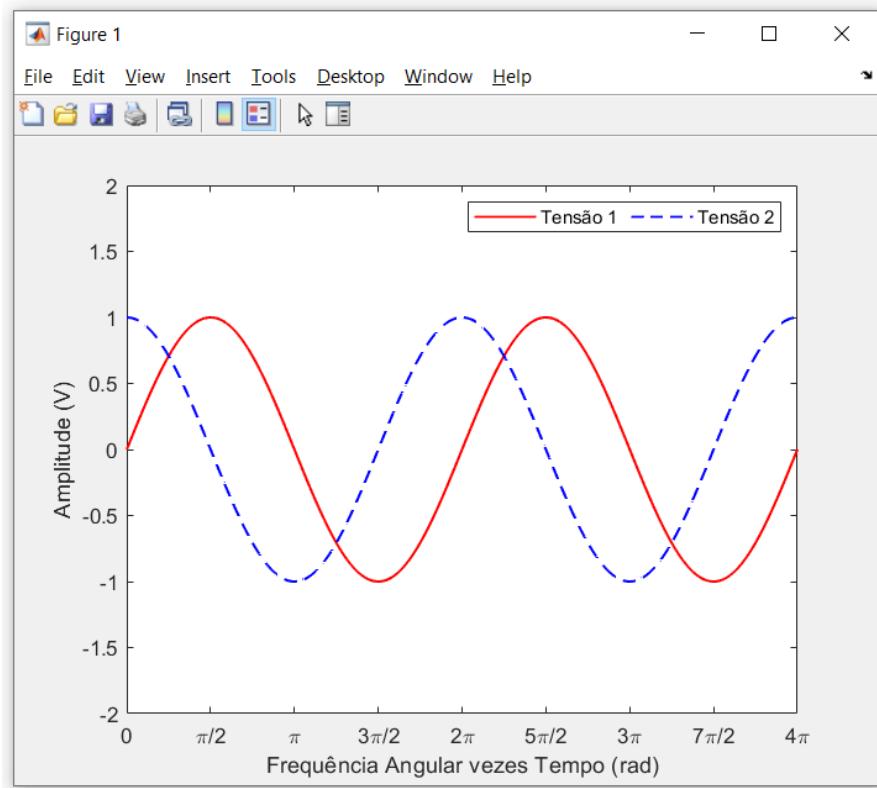
<https://www.mathworks.com/help/matlab/ref/xlabel.html>

<https://www.mathworks.com/help/matlab/ref/ylabel.html>

Passo 12: Define-se a legenda das curvas do gráfico. Para o exemplo que está sendo apresentado, escolheram-se estas configurações de localização e orientação, mas não é uma regra. Avalie como fica melhor para o seu caso. Tome cuidado para a legenda não sobrepor as curvas do gráfico.

% Definindo a legenda das curvas

```
leg=legend('Tensão 1','Tensão 2'); %define a legenda
leg.Location='northeast'; %localização da legenda
leg.Orientation='horizontal'; %orientação da legenda
```



Para mais informações, veja em:

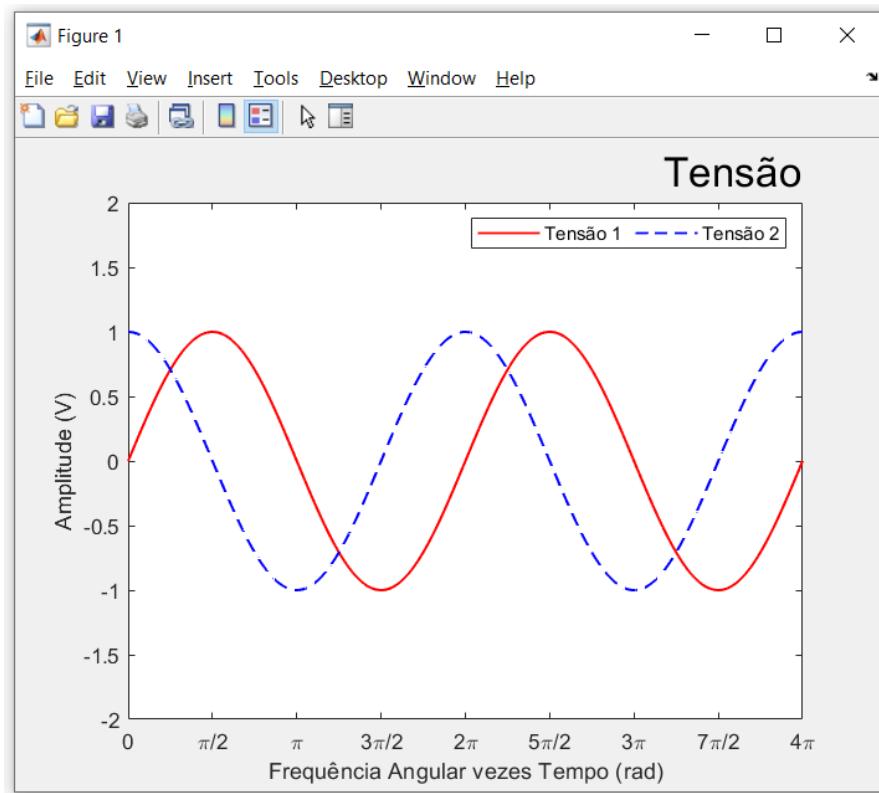
<https://www.mathworks.com/help/matlab/ref/legend.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.illustration.legend-properties.html>

Passo 13: Pode-se definir um título para o gráfico. Porém, não se costuma fazer isso, pois essa informação, geralmente, é colocada na legenda do texto no editor do documento. Dessa forma, esse passo será apenas apresentado neste *E-book*. Portanto, as definições abaixo são só para exemplificar.

% Definindo o título do gráfico

```
tt=title('Tensão');
ax.TitleFontSizeMultiplier=2;
ax.TitleFontWeight='normal';
ax.TitleHorizontalAlignment='right';
```



Para mais informações, veja em:

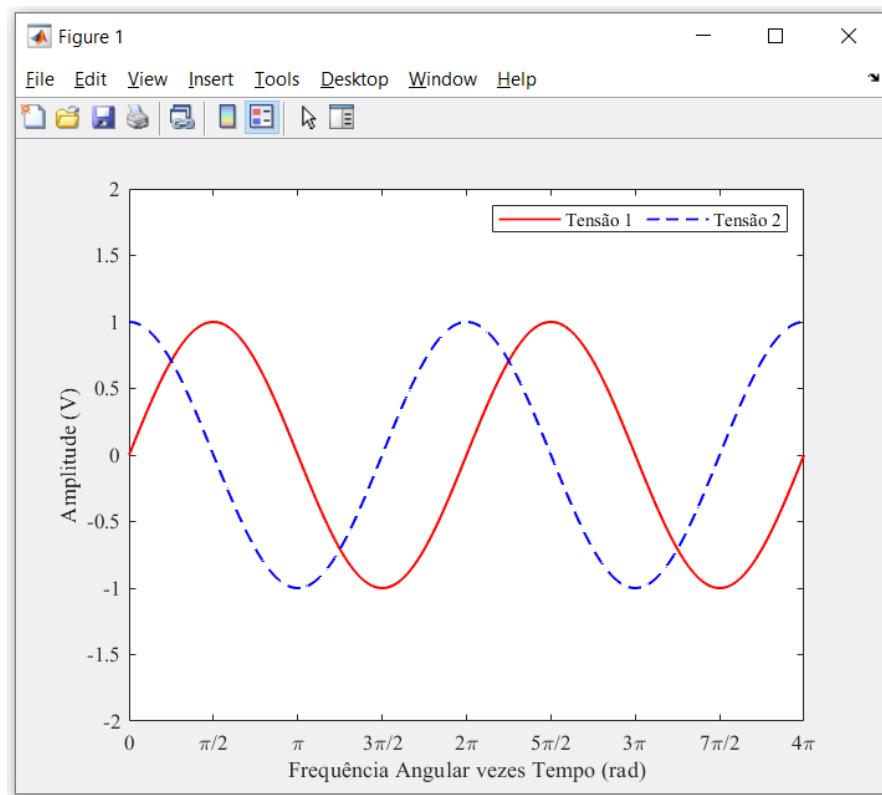
<https://www.mathworks.com/help/matlab/ref/title.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>

Passo 14: Recomenda-se padronizar a fonte do texto da figura com a fonte da publicação científica. Por exemplo, caso o template do artigo que você está escrevendo esteja na fonte *Times New Roman*, defina essa mesma fonte no MATLAB®.

% Definindo a família tipográfica

```
ax.FontName='Times New Roman'; % define a fonte
% ax.FontAngle='italic'; %define a fonte em itálico
% ax.FontWeight='bold'; %define a fonte em negrito
```



Para mais informações, veja em:

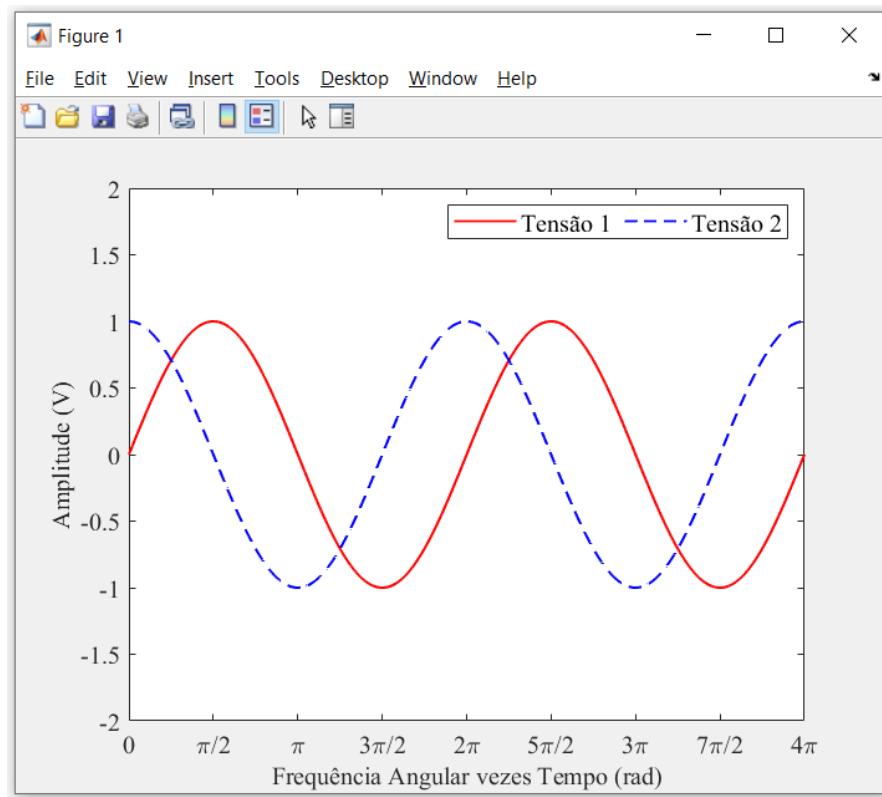
<https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>

Passo 15: Recomenda-se configurar o tamanho da fonte do texto da figura de tal modo que fique cerca de um a dois tamanhos inferior que o tamanho do texto da publicação científica. Por exemplo, caso o *template* do artigo que você está escrevendo esteja com um tamanho de fonte igual a 12, defina uma fonte no MATLAB® para que **visualmente** aparente um tamanho de cerca de 10 a 11. **ATENÇÃO:** Não necessariamente esse vai ser o valor da variável fonte no código. Pode ser que para aparentar essa relação, a variável no MATLAB® precise ser atribuída com um valor menor, igual ou maior. O importante é que a relação entre a fonte do texto da figura e a fonte do texto do documento fique visualmente entre um a dois tamanhos inferior, sendo a fonte da figura a visualmente menor. Sugiro que atribua um valor para a variável fonte, realize o restante dos passos e finalize a figura. Por fim, faça a importação da figura no editor de texto e verifique essa relação de proporção entre as fontes da figura com o restante do texto da publicação científica. Caso não esteja proporcional, altere a variável fonte no código para obter mais ou menos essa proporção.

% Definindo o tamanho da fonte

```
fonte=12; %tamanho da fonte (altere esse valor para obter a proporção recomendada)
```

```
ax.FontSize=fonte; %define o tamanho da fonte  
hx.FontSize=fonte; %tamanho da fonte do eixo x  
hy.FontSize=fonte; %tamanho da fonte do eixo y  
ax.LabelFontSizeMultiplier=1; %fator multiplicador da fonte dos rótulos dos eixos  
leg.FontSize=fonte; %tamanho da fonte da legenda
```



Para mais informações, veja em:

- <https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>
- <https://www.mathworks.com/help/matlab/ref/xlabel.html>
- <https://www.mathworks.com/help/matlab/ref/ylabel.html>
- <https://www.mathworks.com/help/matlab/ref/legend.html>

Passo 16: Caso você esteja editando no LaTeX, pode ser que uma das variáveis que está presente no gráfico foi apresentada no documento como sendo uma variável do *Math Mode* do LaTeX. Sendo assim, pode-se solicitar que o software MATLAB® interprete o texto da figura para o LaTeX. Tal procedimento é feito por meio da variável *interpreter*. Por padrão, essa variável é definida como sendo '`tex`', o que possibilita escrever símbolos como `\alpha`, `\beta`, `\gamma` etc., e visualizar esses símbolos no plot. Porém, variáveis como o tempo, o qual é, geralmente, definido com a letra `t` na fonte do *Math Mode*, necessita-se usar o *interpreter* '`latex`'.

ATENÇÃO: Ao definir o *interpreter* como LaTeX, a fonte do texto que está fora de \$\$ (*Text Mode*) é automaticamente alterada para a *Computer Modern Roman*, e a fonte do texto que está dentro de \$\$ é alterada para a fonte do *Math Mode* do LaTeX, a qual é similar à fonte *Latin Modern Math* em *Math Mode*. Dessa forma, o **Passo 14** perde a validade. Por fim, algumas variáveis definidas nos **Passos 10, 11 e 12** são redefinidas para serem interpretadas em LaTeX.

% Interpretando o texto em LaTeX

```

ax.XTickLabel=({'0','$pi/2$','$pi$','$3\pi/2$',...
    '$2\pi$','$5\pi/2$','$3\pi$','$7\pi/2$','$4\pi$'});
%rótulos do eixo x

hx=xlabel({'$\omega t$ (rad)'}); %rótulo do eixo x

hy.ylabel({'Amplitude (V)'}); %rótulo do eixo y

leg=legend('$v_1$','$v_2$'); %define a legenda

hx.Interpreter='latex'; %rótulo do eixo x em LaTeX

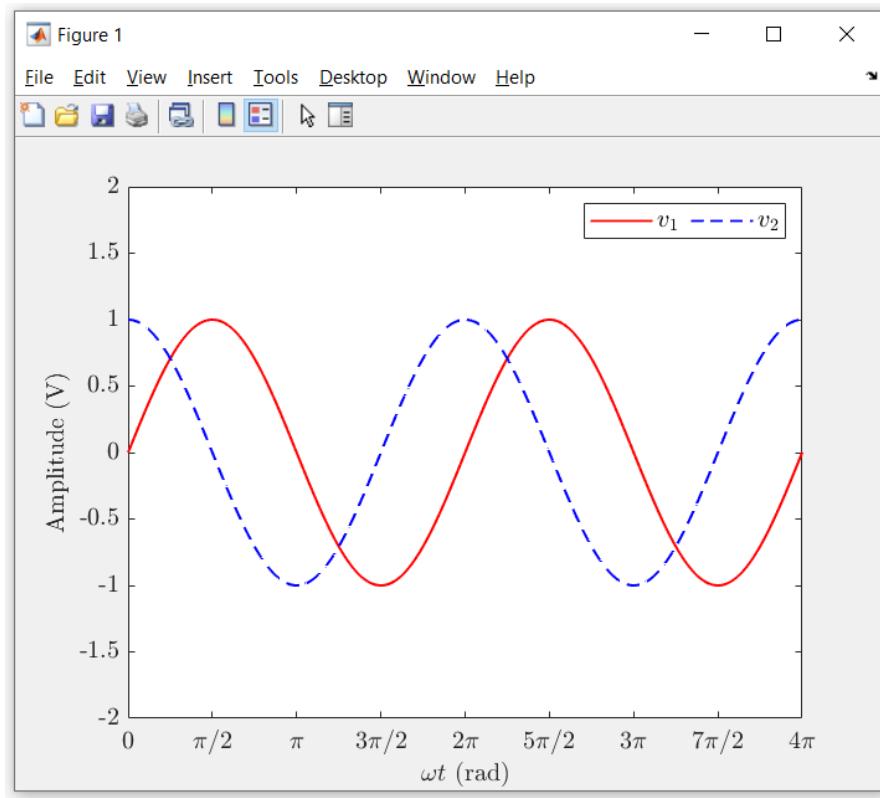
hy.Interpreter='latex'; %rótulo do eixo y em LaTeX

leg.Interpreter='latex'; %legenda em LaTeX

% tt.Interpreter='latex'; %o título em LaTeX (comentado
porque o passo 13 não foi aplicado)

ax.TickLabelInterpreter='latex'; %eixos em LaTeX

```



Para mais informações, veja em:

- <https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>
- <https://www.mathworks.com/help/matlab/ref/xlabel.html>
- <https://www.mathworks.com/help/matlab/ref/ylabel.html>
- <https://www.mathworks.com/help/matlab/ref/legend.html>
- <https://www.mathworks.com/help/matlab/ref/title.html>

Passo 17: Recomenda-se redimensionar a figura para que melhor se aproveite o espaço reservado à ela no documento. Não existe um valor fixo e recomendado para tal, pois dependerá do *template* da publicação científica que você está escrevendo. Por exemplo, se o artigo for de um ou duas colunas, você terá mais ou então menos largura para apresentar sua figura, ou ainda, se você optou por colocar mais de uma figura lado a lado no documento. Enfim, são infinitas as possibilidades de organizar suas figuras na publicação científica. Neste *E-book*, reservar-se-á em mostrar como alterar as propriedades (largura e altura) da figura. Cabe a você testar e avaliar os valores mais adequados para cada figura que está plotando. Por fim, vá ajustando o tamanho da fonte (**Passo 15**) concomitantemente ao ajuste dos valores de largura e altura da figura realizada nesse passo. Lembre-se de avaliar as proporções do tamanho da figura e do tamanho da fonte do texto da figura apenas quando essa já tiver sido importada no editor do documento.

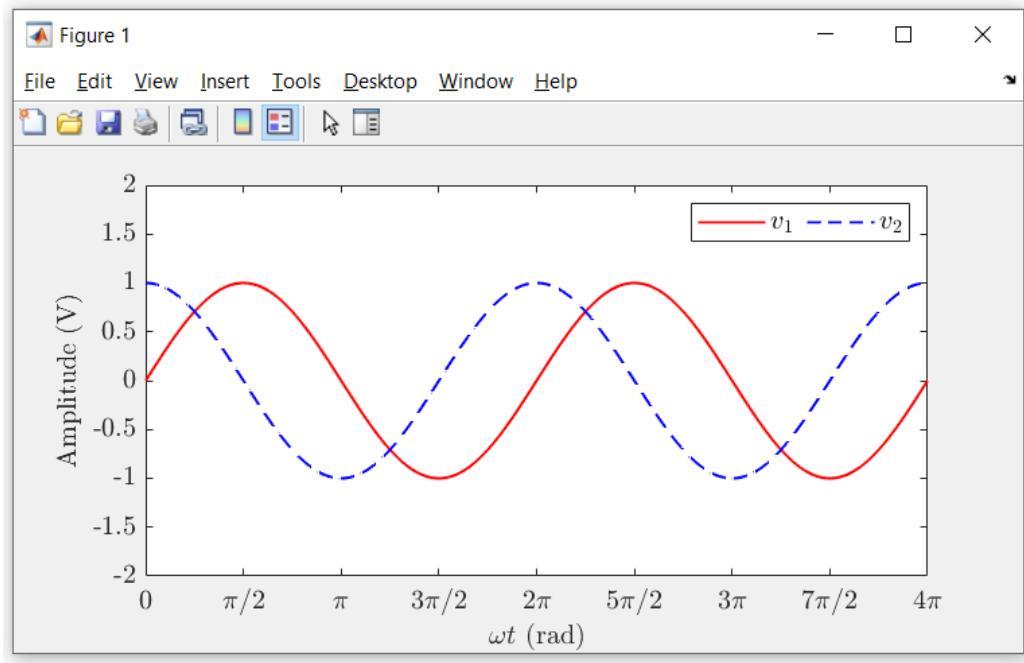
% Definindo o tamanho da figura

`fig.Units='pixels'; %define a unidade das medidas (largura e altura) que serão inseridas no Position. Por padrão, 'pixels' é a unidade, portanto, essa linha pode ser comentada caso queira.`

`fig.Position(3) = 600; %define a largura da figura como sendo 600 pixels. Esse valor é apenas um exemplo, não é uma regra. Avalie cada caso.`

`fig.Position(4) = 300; %define a altura da figura como sendo 300 pixels. Esse valor é apenas um exemplo, não é uma regra. Avalie cada caso.`

`fig.PaperPositionMode = 'auto'; %define que as dimensões da figura a ser salva terão os mesmos valores da figura que está na tela. Por padrão, 'auto' já é definido, portanto, essa linha pode ser comentada caso queira.`



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/figure.html>

<https://www.mathworks.com/help/matlab/ref/gcf.html>

<https://www.mathworks.com/help/matlab/ref/matlab.ui.figure-properties.html>

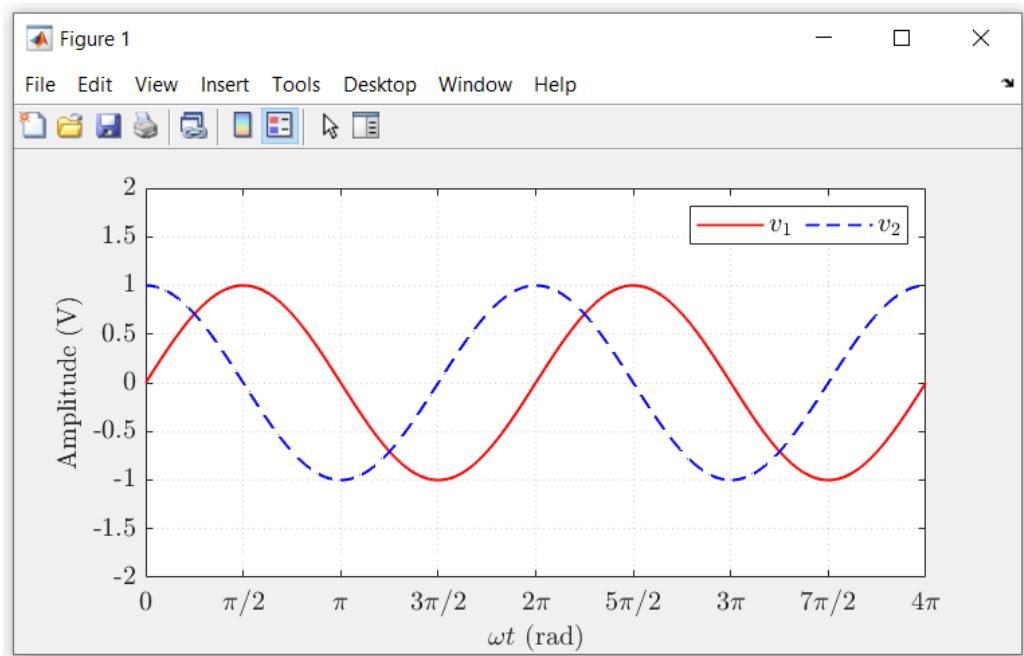
Passo 18: Em alguns casos é interessante adicionar um *grid* para a melhor visualização da figura. Não é uma recomendação. Cada caso deve ser avaliado em particular. Somente para explicação, será adicionado um *grid* no exemplo que está sendo analisado neste *E-book*. A escolha do estilo da linha do *grid* foi só para exemplificar. A transparência do *grid* foi mantida com a configuração padrão (15%) do MATLAB®. Não são regras. Por fim, a cor (*GridColor*) é definido como sendo puramente preta, pois por padrão, a cor definida não é exatamente preta, assim como será visto para outras variáveis no **Passo 19**.

```
% Configurando o grid
grid on; %ativa o grid

ax.GridLineStyle=':'; %define o estilo das linhas do
grid como pontilhado

% ax.GridAlpha=1; %define a transparência do grid com
100%

ax.GridColor='k'; %define a cor do grid como sendo
puramente preta.
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/grid.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>

Passo 19: Assim como foi visto para o *grid* no passo anterior, por padrão, o software MATLAB® não define a cor dos elementos que compõem a figura como sendo exatamente preta, isto é, a cor que visualmente é aceita como preta, não é puramente preta. A cor definida pelo software é, nas escalas RGB e Hexadecimal, como:

Cor preta definida pelo MATLAB®

HEX = #262626 ou RGB = (38,38,38)

Sendo que,

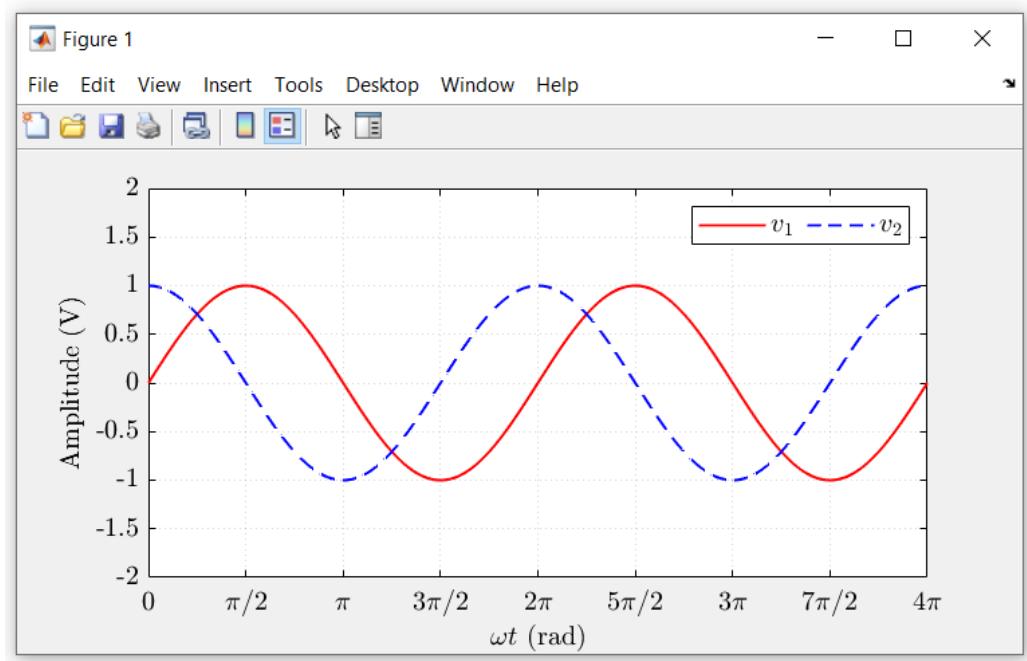
Cor puramente preta

HEX = #000000 ou RGB = (0,0,0)

Por meio do código a seguir, tal característica é corrigida.

%% Redefinindo a cor dos elementos na cor preta

```
ax.XColor='k'; %redefine o eixo x  
ax.YColor='k'; %redefine o eixo y  
hx.Color='k'; %redefine a legenda do eixo x  
hy.Color='k'; %redefine a legenda do eixo y  
leg.EdgeColor='k'; %redefine o box da legenda  
leg.TextColor='k'; %redefine o texto da legenda
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/gca.html>

<https://www.mathworks.com/help/matlab/ref/xlabel.html>

<https://www.mathworks.com/help/matlab/ref/ylabel.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.primitive.text-properties.html>

<https://www.mathworks.com/help/matlab/ref/legend.html>

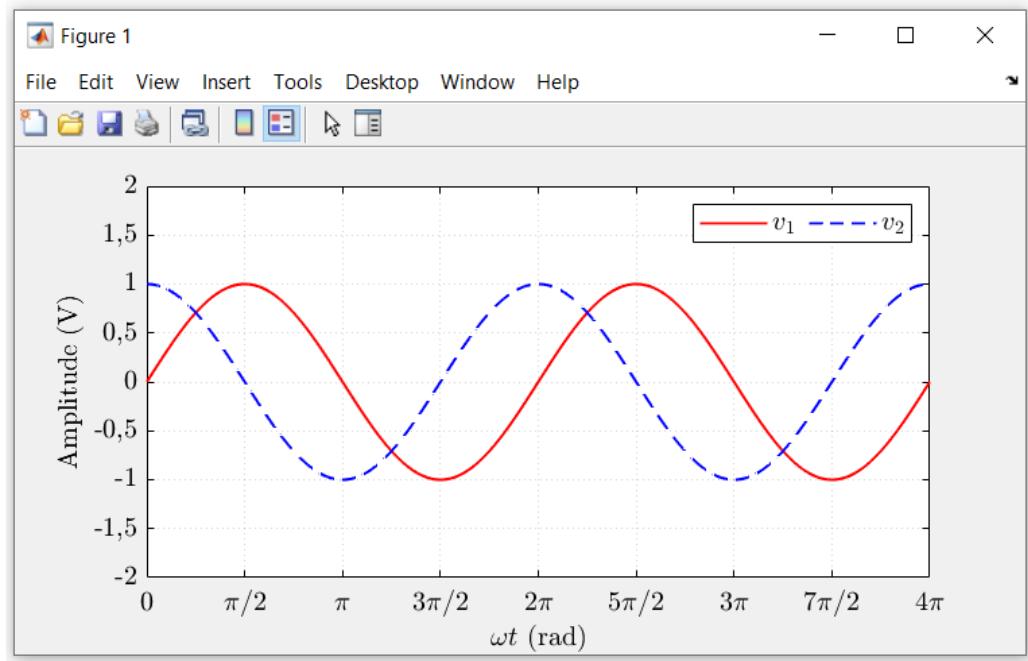
<https://www.mathworks.com/help/matlab/ref/matlab.graphics.illustration.legend-properties.html>

Passo 20: Por padrão, o separador decimal do software MATLAB® é o ponto. Isso porque o software segue o padrão comum em países de língua inglesa. Porém, no Brasil, usa-se a vírgula como separador decimal. Portanto, atente-se para o padrão adotado no template da publicação científica que você está escrevendo. Por exemplo, se o idioma do trabalho for o inglês, utiliza-se, geralmente, o ponto como separador decimal. Se for esse o caso, o MATLAB® já irá te fornecer a figura com o ponto como separador, ou seja, não precisa ser feita nada a mais. Por outro lado, caso a publicação científica esteja em português, recomenda-se utilizar, a vírgula como separador decimal. Para realizar a troca de ponto por vírgula, pode-se usar o código abaixo.

```
%% Trocando o separador decimal

%eixo x
xsepdec=ax.XTickLabel;
for k=1:1:length(xsepdec)
new_xsepdec=strrep(xsepdec(k,:), '.', ',');
new_xsepdec_vetor(k,:)=new_xsepdec;
end
ax.XTickLabel=new_xsepdec_vetor;

%eixo y
ysepdec=ax.YTickLabel;
for k=1:1:length(ysepdec)
new_ysepdec=strrep(ysepdec(k,:), '.', ',');
new_ysepdec_vetor(k,:)=new_ysepdec;
end
ax.YTickLabel=new_ysepdec_vetor;
```



Para mais informações, veja em:

<https://www.mathworks.com/help/matlab/ref/gca.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>

<https://www.mathworks.com/help/matlab/ref/for.html>

<https://www.mathworks.com/help/matlab/ref/strrep.html>

Passo 21: Recomenda-se remover as margens da figura para melhor aproveitamento do espaço do documento.

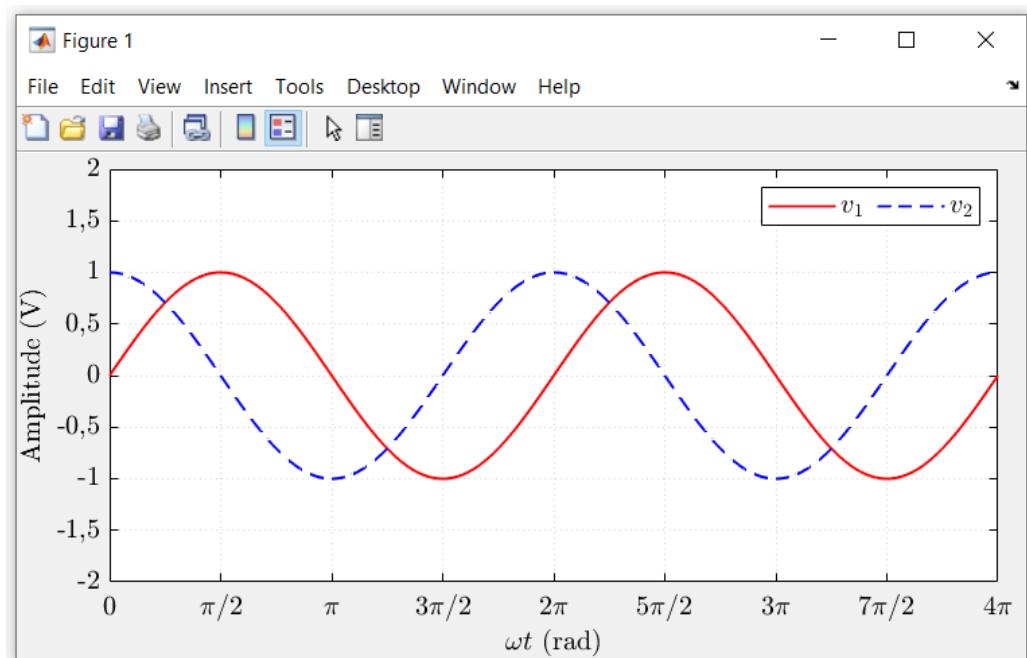
% Removendo as margens da figura

```
ax.LooseInset=[0 0 0 0];
```

Caso o código acima não remova as margens para o seu caso, tente a solução alternativa apresentada abaixo.

% Removendo as margens da figura - Solução alternativa

```
outerpos = ax.OuterPosition;
ti = ax.TightInset;
left = outerpos(1) + ti(1);
bottom = outerpos(2) + ti(2);
ax_width = outerpos(3) - ti(1) - ti(3);
ax_height = outerpos(4) - ti(2) - ti(4);
ax.Position = [left bottom ax_width ax_height];
```



Para mais informações, veja em:

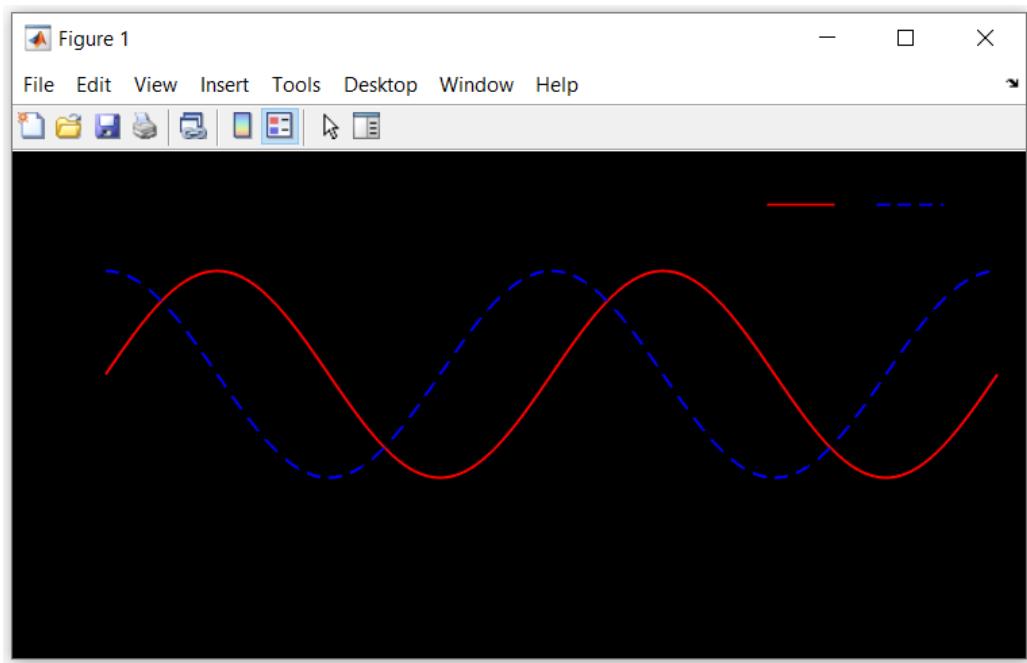
<https://www.mathworks.com/help/matlab/ref/gca.html>

<https://www.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>

Passo 22: Por padrão, as figuras do software MATLAB® são plotadas com fundos brancos. Recomenda-se removê-los. Para tal, utilize o código abaixo. Não se assuste com o resultado. A figura ficará conforme se deseja ao salvar com a extensão recomendada no próximo passo.

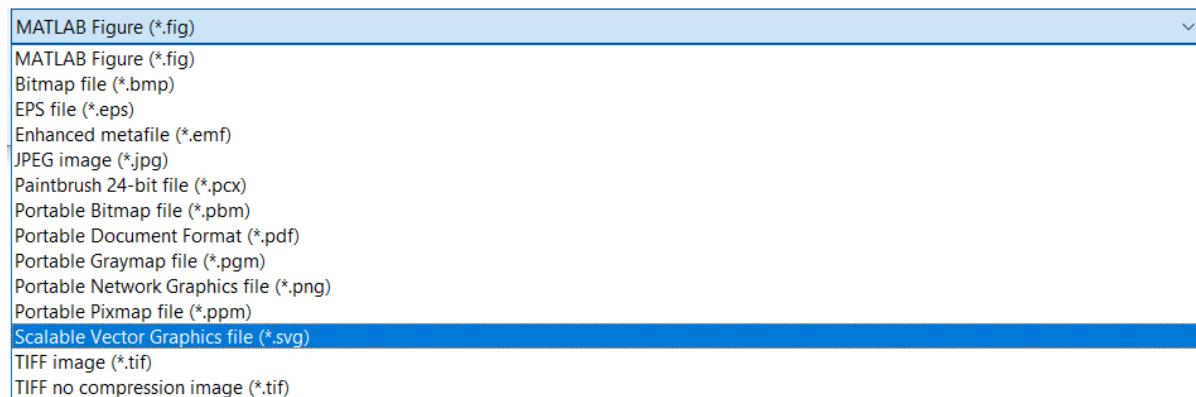
% Removendo o fundo da figura

```
ax.Color='none'; %remove o fundo da parte interna  
fig.Color='none'; %remove o fundo da parte externa  
leg.Color='none'; %remove o fundo do box da legenda  
fig.InvertHardcopy='off'; %mantém as definições ao  
salvar a figura
```



Passo 23: O último passo no processo de edição de figuras para publicações científicas consiste em salvar a imagem para, que, posteriormente, essa seja importada pelo editor de texto.

No software MATLAB® é possível escolher salvar uma figura entre 14 diferentes extensões. Na captura de tela abaixo, apresentam-se essas opções disponíveis. No *E-book “Como criar e editar imagens para publicações científicas?”*, explica-se a diferença entre os tipos e extensões das imagens digitais. Caso você ainda não tenha lido esse *E-book*, adquira-o para mais detalhes.



A figura apresentada como exemplo neste *E-book* foi salva com a extensão .svg e importada por um editor de texto LaTeX. O resultado é apresentado em uma página no final deste *E-book*. Convido-o a observar o resultado dos ajustes que foram inseridos ao longos dos passos sugeridos. Por fim, proponho que amplie (dê um zoom) no documento, mais especificamente, na imagem por meio da utilização de um leitor de PDF e, assim, observar a qualidade da figura. Portanto, seguindo os passos sugeridos, obtém-se uma figura bem editada e com qualidade para que possa ser inserida em uma publicação científica. Por fim, convido-o a refazer o código apresentado neste *E-book*, realizando adaptações para atender as especificidades da figura que você pretende plotar. Dessa forma, você estará praticando os comandos que foram aqui sugeridos.

Considerações Finais

Neste *E-book*, apresentou-se um passo a passo para editar figuras no software MATLAB® com o objetivo de usá-las em publicações científicas. Um total de 23 passos foram sugeridos, dentre ajustes no texto, na cor dos elementos, no tamanho e demais propriedades da figura. As sugestões apresentadas em cada passo não são regras. Avalie se é coerente realizar cada procedimento na figura que você está plotando. Por fim, espero que os comentários apresentados neste *E-book* contribuam para o seu crescimento e desenvolvimento profissional. Sucesso! Abraços!

Passo a passo para editar figuras no MATLAB®

Caio Muller Santos Ribeiro

Brasília, 2022

Resultado do Exemplo do *E-book*

Na Figura 1, apresentam-se as tensões v_1 e v_2 , sendo:

$$v_1 = 1 \cdot \sin(\omega t) , \quad (1)$$

$$v_2 = 1 \cdot \cos(\omega t) , \quad (2)$$

em que ω é a frequênciā angular, e t é o tempo.

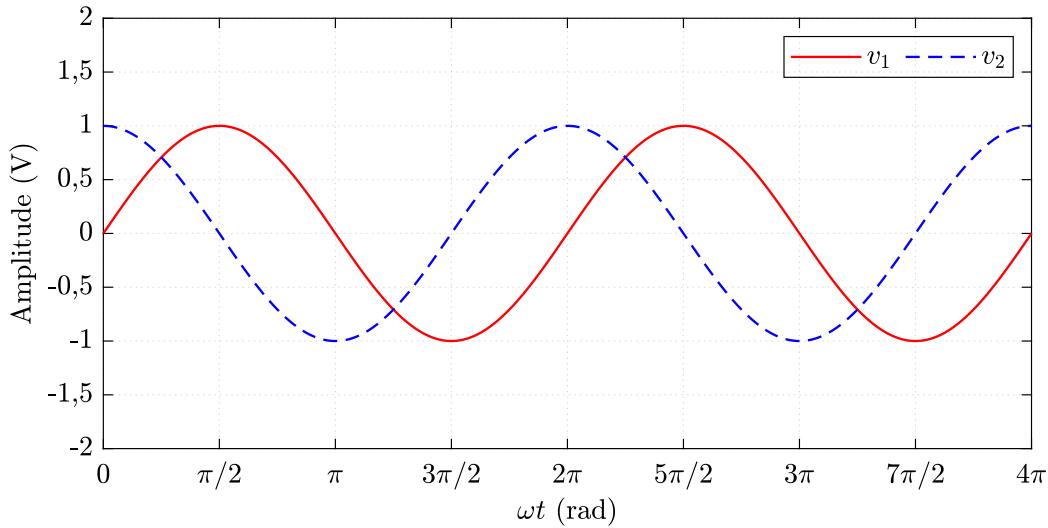


Figure 1: Tensão.