

# 几个 **MATLAB** 函数句柄作图的 **Demo**

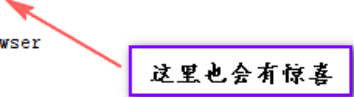
首先，大家要学会使用 MATLAB 的帮助文档 (多 help、doc、lookfor，常看看其后的 see also)，要能做到举一反三。

```
>> help gca
GCA Get handle to current axis.

H = GCA returns the handle to the current axis in the current
figure. The current axis is the axis that graphics commands
like PLOT, TITLE, SURF, etc. draw to if issued.
Use the commands AXES or SUBPLOT to change the current axis
to a different axis, or to create new ones.

See also axes, subplot, delete, cla, hold, gcf, gcbo, gco, gcbf.

Reference page in Help browser
doc gca
```



要熟练掌握 MATLAB 中常用的二维、三维作图函数的使用，对绘制出的图形能在图形窗口中进行手工编辑，明白每个菜单的定义。

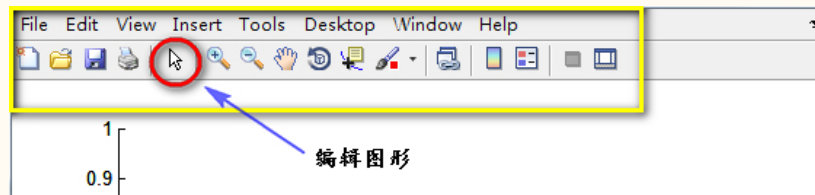


Figure 1: 了解绘图窗口各项菜单的含义

# 利用句柄控制图形特征

首先看一个简单的例子，运行完原始代码后，在图形窗口点击 File -> Generate M-File..., 就可以查看 MATLAB 自动生成的代码。

原始代码

```
x = 0:0.1:pi;
plot(x, sin(x), 'o-', x, cos(x), ':+');
```

MATLAB 自动生成的代码

```
function createfigure(X1, YMatrix1)
%CREATEFIGURE(X1,YMATRIX1)
% X1: vector of x data
% YMATRIX1: matrix of y data
% Create figure
figure1 = figure('PaperSize',[20.98 29.68]);
% Create axes
axes('Parent',figure1);
box('on');
hold('all');
% Create multiple lines using matrix input to plot
plot1 = plot(X1,YMatrix1);
set(plot1(1),'Marker','o');
set(plot1(2),'Marker','+', 'LineStyle',':');
```

执行完一个绘图命令后，MATLAB 背后做了更多具体的工作，可以看出 MATLAB 完整的绘制一个图形是分为三步的：

- 1) 建立图形窗口 `figure1 = figure('PaperSize',[20.98 29.68]);`;
- 2) 在图形窗口上建立坐标系 `axes('Parent',figure1);`;
- 3) 在坐标系上绘制图形 `plot1 = plot(X1,YMatrix1);`;

进一步，如果把图形窗口，坐标系都看成容器的话，则坐标系为图形窗口（作为 'Parent'）的子容器，同时又为线条（作为 'Children'）的父容器。图形窗口、坐标系以及线条对象都有对应的入口地址，称之为函数句柄（图形窗口为整数，坐标轴及线条对象为双精度浮点数 double）。

要控制绘制出图形的特征，就要获得这些句柄，常见的函数句柄如下：

<code>gco</code>	当前对象的句柄（get current object）
<code>gcf</code>	当前图形的句柄（get current figure）
<code>gca</code>	当前坐标轴的句柄（get current axes）
<code>h = plot(...)</code>	a column vector of handles to lineseries objects, one handle per line
<code>new_handle = copyobj(handle, parent)</code>	拷贝由句柄 handle 指定的对象到父容器 parent 中，并返回这个新生成对象的句柄 new_handle

每个句柄中都包含很多参数-值对，要了解这些参数-值对的话，可以先生成一个句柄，然后使用 `get` 函数查看，通过 `find` 函数查找特定的参数，并通过 `set` 函数修改这些参数的值。

原始代码

```

clc,clear,close all;

x = 0:0.4:2*pi;

figure1 = figure(1);
h = plot(x, exp(x));

% 查看参数-值对
get(gca);

% 更改 Y 轴颜色
set(gca, 'YColor', 'red');

% YLabel 为 gca 的子句柄, 查看其信息
get(get(gca, 'YLabel'));

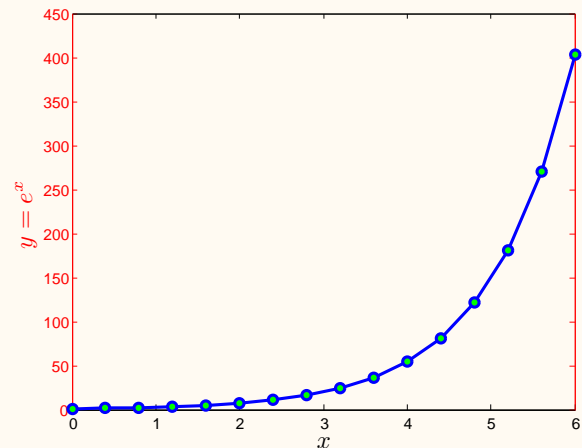
% 更改 YLabel
set(get(gca, 'YLabel'), 'String', '$y = e^x$', ...
    'Interpreter', 'latex', 'FontSize', 16);

% 函数 xlabel 能达到同样的效果
xlabel('$x$', 'Interpreter', 'latex', 'FontSize', 16);

% 获取线条对象的参数信息
get(h);

% 对一些常见的特征进行修改
% set(findobj('Type','line'),'LineWidth', 2)
set(h, 'Marker', 'o', 'MarkerFaceColor', 'green', ...
    'MarkerEdgeColor', 'blue', 'Markersize', 6, ...
    'LineStyle', '-', 'LineWidth', 2);

```



由上面的代码 `get(gca)` 可以看出坐标轴既有 Parent 又有 Children, 其值分别与 `figure1` 和 `h` 相等。

函数句柄基本就这些知识, 把上面的代码都弄明白的话, 在批量绘图的情况下, 就能一次到位, 达到很好的效果。此外如果绘制少量图形, 嫌使用句柄麻烦, 可以手工编辑或者使用 `inspect` 函数, 直接跳出编辑的 GUI。不知道怎样用的可以先 `help inspect` 一下。编辑后再用 **MATLAB** 自动生成一下代码, 看官方是怎么设置的。

原始代码

```

clc,clear,close all;

x = 0:0.4:2*pi;

figure1 = figure(1);
h = plot(x, exp(x));

% 编辑坐标轴
inspect(gca);

% 更改 YLabel
inspect(get(gca, 'YLabel'));

% 更改 XLabel
inspect(get(gca, 'XLabel'));

% 对一些常见的特征进行修改
inspect(h);

```

懂得了怎样使用函数句柄, 再把以下几个 demo 看懂基本上就没问题了。

## ● 双坐标轴绘图

原始代码

```
clc,clear,close all;

x = 0:0.4:2*pi;

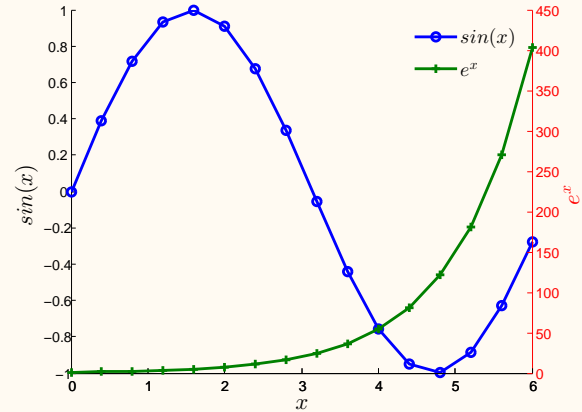
figure1 = figure;
[AX,Ha,Hb]=plotyy(x, sin(x), x, exp(x));

% 设置坐标轴颜色
set(AX(1), 'YColor', 'k', 'Box', 'off');
set(AX(2), 'YColor', 'r', 'Box', 'off');

% 设置线条宽度及 Marker
set(Ha,'LineWidth',2,'LineStyle','-','Marker','o');
set(Hb,'LineWidth',2,'LineStyle','-','Marker','+');

% 生成 legend
HL = legend([Ha,Hb],'$sin(x)$','$e^x$');
get(HL);
set(HL,'Interpreter','latex','Box','off','FontSize',14);

% 设置横纵坐标说明
set(get(AX(1), 'XLabel'), 'String', '$x$', ...
    'Interpreter', 'latex', 'FontSize',16);
set(get(AX(1), 'YLabel'), 'String', '$sin(x)$', ...
    'Interpreter', 'latex', 'FontSize',16);
set(get(AX(2), 'YLabel'), 'String', '$e^x$', ...
    'Interpreter', 'latex', 'FontSize',16);
```



## ● 复杂的双坐标绘图

原始代码

```
clc, clear, close all;

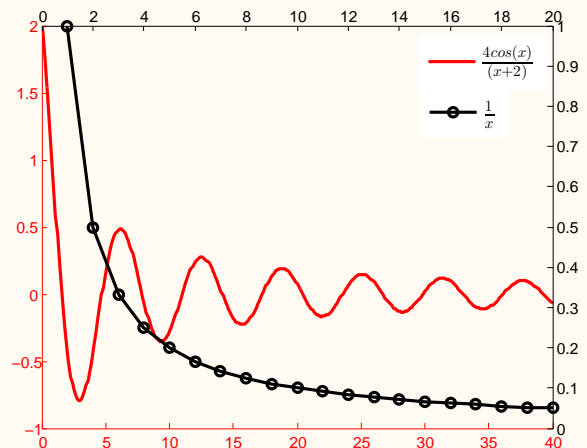
x1 = 0:0.1:40;
y1 = 4 * cos(x1) ./ (x1 + 2);
x2 = 1:1:20;
y2 = 1 ./ x2;

figure1 = figure(1);
h1 = plot(x1, y1, '-r', 'LineWidth', 2);
axes1 = gca;
set(axes1, 'XColor', 'r', 'YColor', 'r', 'box', 'off');
legend1 = legend('$\frac{4\cos(x)}{(x+2)}$');
set(legend1,'EdgeColor','white', ...
    'Interpreter','latex', 'FontSize', 14);

axes2 = axes;
h2 = plot(axes2, x2, y2, '-ko', 'LineWidth', 2);

set(axes2, 'Position', get(axes1,'Position'),...
    'XAxisLocation', 'top',...
    'YAxisLocation', 'right',...
    'Color', 'none', 'box', 'off', ...
    'XColor', 'k', 'YColor', 'k');

legend2 = legend('$\frac{1}{x}$', 'location', 'Best');
set(legend2, 'EdgeColor', 'white', 'Color', [1,1,1], ...
    'Interpreter', 'latex', 'FontSize', 14)
```



## ● 绘制图上图

原始代码

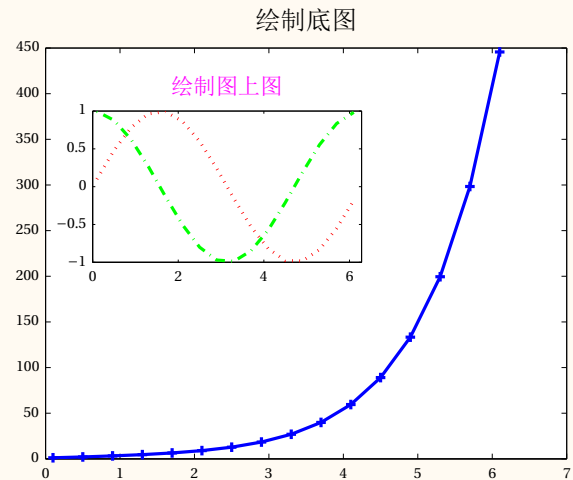
```
clc, clear, close all;

t = 0.1:0.4:2*pi;
y = sin(t) + exp(t);

figure1 = figure(1);

axes1 = axes;
h1 = plot(t, y, '+-');
set(h1, 'LineWidth', 2);
title(' 绘制底图','fontName',' 楷体','fontSize',16);

% axes('position', [left, bottom, width, height])
% 新建坐标系的原点坐标为 (left, bottom), 相对于当前图形
% 窗口的左下角 (0,0); 图形窗口右上角坐标为 (1,1)
% 位置不满意的话, 可以编辑状态下拖动
axes2 = axes('position',[0.2, 0.5, 0.4, 0.3]);
set(axes2, 'FontSize', 10);
h2 = plot(axes2, t, sin(t), 'r:', t, cos(t), 'g-');
set(h2, 'LineWidth', 2);
title(' 绘制图上图','fontName',' 楷体',...
'fontSize',14,'color','Magenta')
xlim([0, 2*pi]);
```



原始代码

```
clc, clear, close all;

% Create data
t = linspace(0, 2*pi, 40);
y = sin(t);

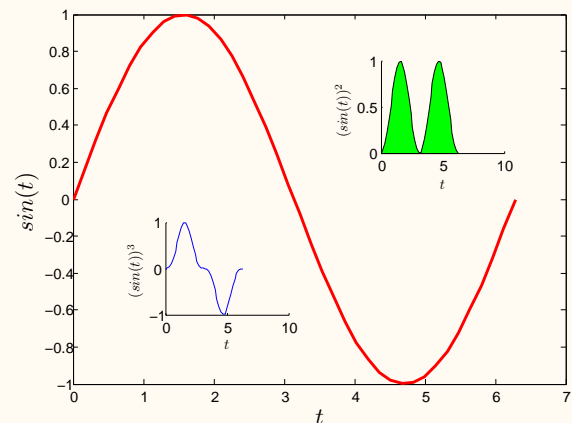
% Place axes at (0.1,0.1) with width and height of 0.8
figure1 = figure(1);
axes1 = axes('position', [0.1 0.1 0.8 0.8]);

% Main plot
plot(axes1, t, y, 'r', 'LineWidth', 2);
xlabel('$t$', 'Interpreter', 'latex');
ylabel('$sin(t)$', 'Interpreter', 'latex');
set(axes1, 'box', 'on');

% Adjust XY label font
xlabel1 = get(gca, 'xlabel');
set(xlabel1, 'fontSize', 16);
ylabel1 = get(gca, 'ylabel');
set(ylabel1, 'fontSize', 16);

% Place second set of axes on same plot
axes2 = axes('position', [0.6 0.6 0.2 0.2]);
fill(t, y.^2, 'g');
set(axes2, 'box', 'off');
xlabel('$t$', 'Interpreter', 'latex');
ylabel('$sin(t)^2$', 'Interpreter', 'latex');

% Add another set of axes
axes3 = axes('position', [0.25 0.25 0.2 0.2]);
plot(t, y.^3);
set(axes3, 'box','off');
xlabel('$t$', 'Interpreter', 'latex');
ylabel('$sin(t)^3$', 'Interpreter', 'latex');
```



## ● 使用 subplot

原始代码

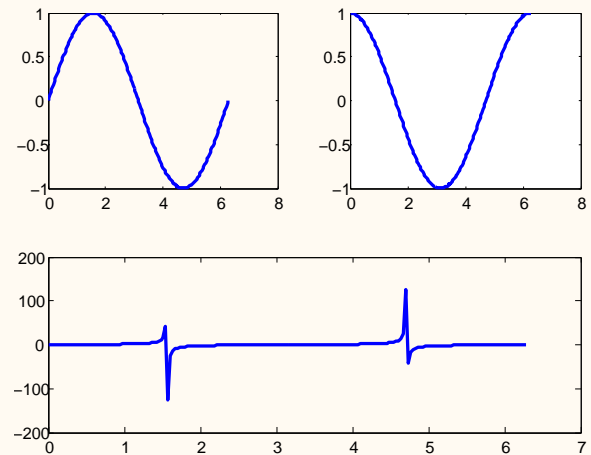
```
clc, clear, close all;

t=linspace(0, 2*pi, 200);
figure1 = figure(1);
% 返回的是一个坐标轴句柄
h1 = subplot(2, 2, 1);
plot(t, sin(t));
ax1 = gca;
% true
disp(ax1 == h1);

subplot(2, 2, 2);
plot(t, cos(t))
ax2 = gca;

subplot(2, 2, [3, 4]);
plot(t, tan(t));
ax3 = gca;

AX = get(figure1, 'Children');
% true
disp(all(AX == [ax3; ax2; ax1]));
```



## ● 多列 legend

原始代码

```
clc, clear, close all;

x = 1:10;
y = [x; 2 * x; x + 4; 0.5 * x - 2; x - 4; x / 3];
colors={'-r.' '-bx' '-k+' '-ms' '-md' '-cv' '-k.' ...
        '-r*' '-g*' '-b*' '-y*' '-m*' '-c*' '-k*'};

figure1 = figure(1);
hold on;

for i = 1:size(y, 1)
    plot(x, y(i, :), colors{i});
end

% First column
% Axes handle 1 (this is the visible axes)
axes1 = gca;
handles = get(gca, 'Children');
set(handles, 'LineWidth', 1);

% Legend at axes 1
% [legend_h,object_h,plot_h,text_strings] = ...
% legend(axes1, handles(1:3),'y1','y2','y3',1);
legend1 = legend(axes1, handles(1:3),{'y1','y2','y3'},1);
set(legend1, 'EdgeColor', 'white');

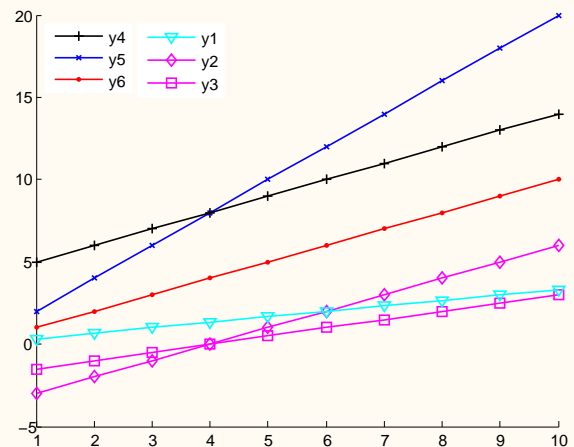
% There are three ways to add the second column

%% 1. Use copyobj function(copy axes)
axes2 = copyobj(axes1, figure1);
set(axes2, 'Color', 'none', 'Visible', 'off');
legend(axes2, handles(4:6),{'y4','y5','y6'},2);

%% 2. Use copyobj function(copy legend)
copyobj(legend1, figure1);
legend(handles(4:6),{'y4','y5','y6'},3);

%% 3. Add a invisible axes
% Axes handle 2 (invisible, only for place the second legend)
axes3 = axes('position',get(axes1,'position'), 'Visible','off');

% Legend at axes 2
legend2 = legend(axes3, handles(4:6),{'y4','y5','y6'},4);
set(legend2, 'EdgeColor', 'white');
```





## 最后给大家说明几点

- 学会使用 [Google](#), 找不到的一些问题尝试用英文检索
- MATLAB 官网的 [File Exchange](#) 是个好地方, 一些算法, 常见问题都可能在这找到答案 (当然, 要把检索词换成英文)
- 更多漂亮、专业的绘图示例在这里, [MATLAB Plot Gallery](#)