

## Guide of Matlab

内容概要： 数学建模算法

创建时间： 2022/4/5 12:40

更新时间： 2022/4/17 11:00

作者： TwinkelStar

---

# Matlab 绘图

## Matlab Plot Table

---

---

### 1、操作系统相关环境

#### 1) 硬件环境:

➤ 电脑

#### 2) 软件环境:

➤ Matlab2018a(程序设计软件)

#### 3) 操作系统(2 选 1):

➤ Windows10

### 2、数据可视化

图像是结果的一种可视化表现，它能直观的体现数学计算的结果，并且能体现获得结果的准确性。在做数据分析的时候，将其可视化可以直观多维的展示数据，可以让人们更好的发现并且记住数据的特征，对于数学建模比赛，掌握一些绘图方法是非常重要的，更加直观的对数据进行可视化，便于理解。而使用 Matlab 可以非常简单的进行绘图，本文档介绍一些基本绘图方法的整理，包括一些基础的绘图函数。

### 1) 常用的绘图函数

在 Python 中的可视化库为 `matplotlib`，是 Matlab 软件内置的 API 库，不再做重复讲解。在 Matlab 中集成了许多内置的绘图函数，常用的函数有以下：

- ① `plot`, `plot3`: 建立向量或矩阵的图形；
- ② `Loglog`: x、y 轴都取对数标度建立图形；
- ③ `Semilogx`: x 轴用于对数标度，y 轴线性标度绘制图形；
- ④ `Title`: 给图形加标题；
- ⑤ `Xlabel`, `Ylabel`: 给 x,y 轴加标记；
- ⑥ `Text`: 在图形指定的位置上加文本字符串；
- ⑦ `Gtext`: 在鼠标的位置上加文本字符串；
- ⑧ `Grid`: 打开网格线；

## 3、绘图入门

### 1) `plot` 的基本调用格式：

在 Matlab 里面做二维图像最基础也是最常用的两函数：`plot()`和 `fplot()`函数,其中，`plot` 的经常使用的方法有三种，如图 1 所示：

```
one plot(x)
two plot(x,y,参数)
three plot(x1,y1,x2,y2,...,xn,yn)
```

图 1.plot 函数调用格式  
fig1.plot function call format

首先，第一种方法里面若 `x` 为一维数组，则作出的图像是以其数组长度为横坐标，间隔为 1，以数组中的具体值为纵坐标的。而其也可以为复数变量，如图 2 所示：

```

y = [5,10,15,20];
z = [100, 200, 300, 400];
x = y+z*1i;
%x将变为复数变量
%图像会以y为横坐标。虚部z为纵坐标作图
plot(x)

```

图 2.plot 函数调用  
fig2.plot function call

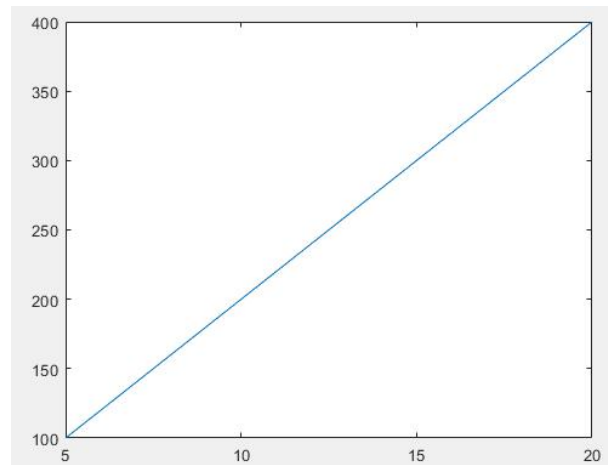


图 3.函数运行效果  
fig3. Function execution effect

对于第二种形式就更好理解了，往往其中的  $x$ 、 $y$  都为二维数组，其实  $y$  也就是  $x$  对应的函数值，后边的参数用于指定曲线的线形、颜色和数据点标记，如图 4 所示：

```

x = [0:0.01:10];
y = sin(x);
plot(x,y, "-r*")

```

图 4.plot 函数调用  
Fig4.plot function call

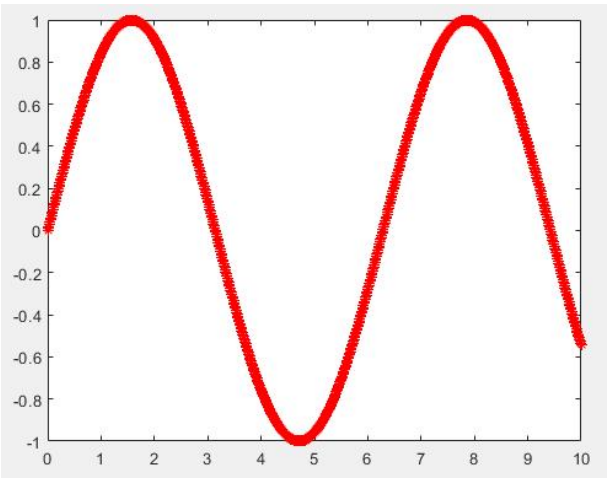


图 5.函数运行效果  
Fig5. Function execution effect

可以看到，x 是一个长度为 1001 的一维数组，y 是和 x 等长的在 sin(x)上的一维数组，后边的-r\*分别为曲线线性、颜色、数据点标记，其中参数的一些具体属性如下表 1 所示：

表 1. 部分参数描述图表  
table1. Some parameter description diagrams

线形	颜色	数据点标记
“_”：实线	“k”：黑色	“*”：星号
“:”：虚线	“b”：蓝色	“o”：圆圈
“-.”：点划线	“r”：红色	“s”：方块
“--”：双划线	“g”：蓝色	“p”：五角星

这些是基本的绘图参数指令，关于参数的使用是在每种方法中都可以添加，三种常用示例只是简单示例。关于第三种使用方法，意思是把 n 个图像做出在一个窗口中，不用再撰写多个 plot。

2) fplot 的基本调用格式如图 6 所示：

```
one fplot(f,lims,参数)
two fplot(funx,funy,tlims,参数)
```

图 6.fplot 函数调用格式  
fig6.fplot function call format

第一个方法 fplot(f,lims,参数), f 代表一个函数，通常采用函数句柄的形式。lims 为 x 轴的取值范围，用二元向量[xmin,xmax]描述，默

认值为[-5,5]。参数定义与plot函数相同。例如使用fplot函数绘制sin(x)图像如下：

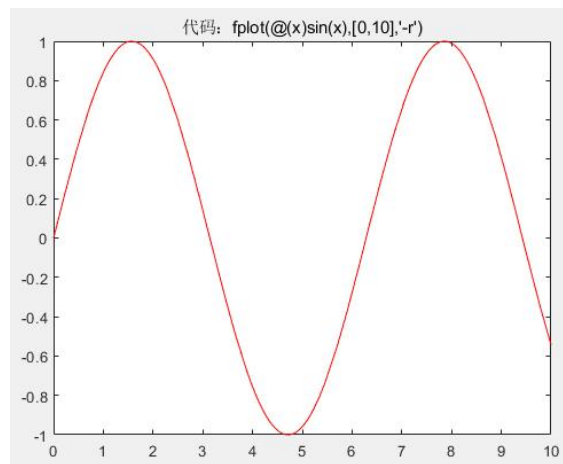


图 7.fplot 绘制正弦曲线  
fig7. fplot plots the sin curve

在第二种方法中 fplot(funx, funy,tlims,参数), funx、funy 代表函数, 通常采用函数句柄的形式。tlims 为参数函数 funx 和 funy 的自变量的取值范围, 用二元向量[tmin,tmax]描述。如绘制参数方程  $(x=t\sin(t), y=t\cos(t))$  曲线如图 8 所示:

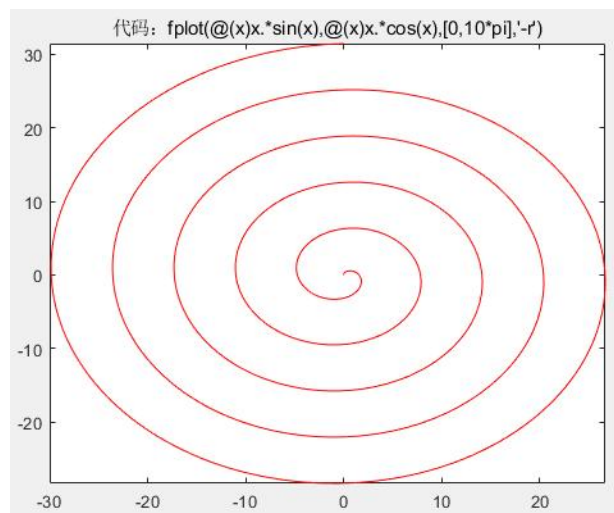


图 8.fplot 绘制参数方程  
fig8. fplot plots parametric equations

### 3) 特殊二、三维绘图函数:

- ① bar(x,y) (barh(x,y),bar3,bar3h): 直方(水平)图;
- ② comet(x,y) (comet3): 建立彗星流动图;

- ③ `errorbar(x,y,l,u)`: 图形加上误差范围;
- ④ `polar(theta,rho)`: 极坐标图;
- ⑤ `hist(y,x)`: 向量统计的直方图,其中  $y$  为要统计的。当  $x$  为标量时,  $x$  指定了统计的区间数; 当  $x$  为向量时, 以该向量中各元素为中心进行统计, 区间数等于  $x$  向量的长度;
- ⑥ `rose(theta)`: 极坐标频数累计柱状图;
- ⑦ `stairs(x,y)`: 阶梯图;
- ⑧ `fill`: 实心图;
- ⑨ `feather` 羽毛图;
- ⑩ `compass`: 罗盘图;

#### 4) 部分代码实例:

彗星图, 使用 `comet (comet)` 函数, 参考 `cometexample.m` 给出的代码, 效果如图 9 所示:

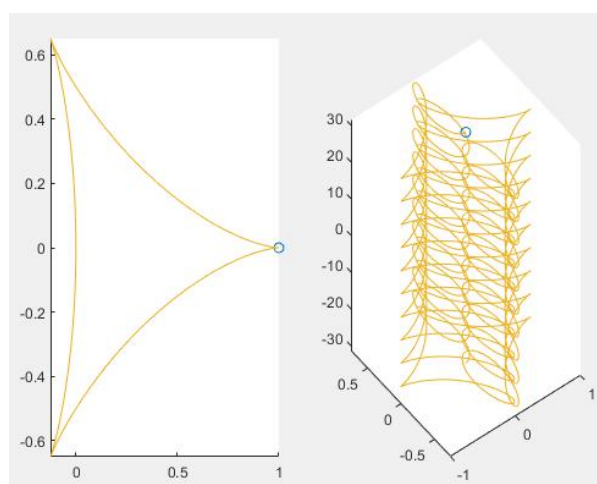


图 9.彗星图

fig9. Comet diagram

等高线三维图, 调用 `contours (contours3)` 函数, 参考 `conplot3.m` 给出的代码, 效果如图 10 所示:

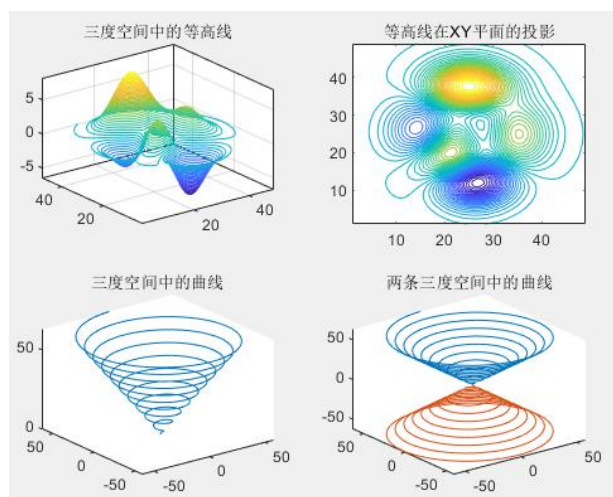


图 10.等高线三维图  
fig10. Three dimensional contour map

#### 4、例题

1) 例题 1: 在某海域测得一些点(x,y)处的水深  $z$  (单位: 英尺)  
由下表给出, 水深数据是在低潮时 测得的。船的吃水深度为 5 英尺,  
问在矩形(75,200)\*(-50,150)里的哪些地方船要避免进入。

表 2. xyz 数据 (单位: 英尺)  
table2. xyz data (Unit: Feet)

x	129.0	140.0	103.5	88.0	185.5	195.0	105.5
Y	7.5	141.5	23.0	147.0	22.5	137.5	85.5
Z	4	8	6	8	6	8	8
X	157.5	107.5	77.0	81.0	162.0	162.0	117.5
Y	-6.5	-81.0	3.0	56.5	-66.5	84.0	-33.5
Z	9	9	8	8	9	4	9

问题求解：

- ① 作出测量点的分布图；
- ② 作出海底地貌图；
- ③ 危险区域平面图；

针对问题①，根据题目所给的  $x$ ,  $y$ ,  $z$  的数据绘制二维图，运行 `example1.m` 文件，结果如图 9 所示：

```
clear;
x=[129 140 103.5 88 185.5 195 105.5 157.5 107.5 77 81 162 162 117.5];
y=[7.5 141.5 23 147 22.5 137.5 85.5 -6.5 -81 3 56.5 -66.5 84 -33.5];
plot(x,y,'+');
title("测量点分布图")
z=[-4 -8 -6 -8 -6 -8 -8 -9 -9 -8 -8 -9 -4 -9];
```

图 9.例题 1 代码  
fig9. Example one code

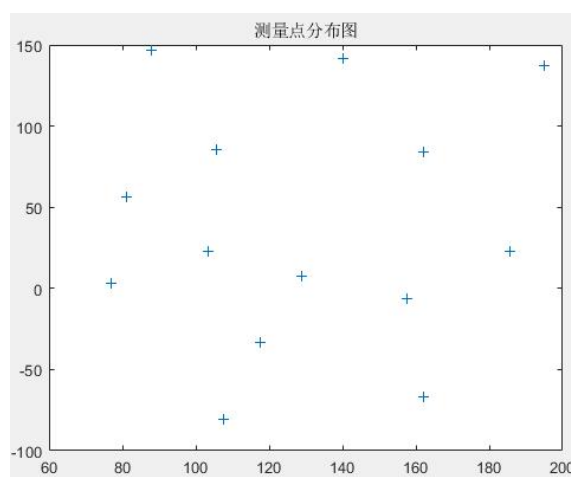


图 10.测量点分布图  
fig10. Distribution map of measuring points

针对问题②，根据题目所给的矩阵范围，调用 `meshgrid` 绘制网格（相当于海域的长宽），再调用 `surf` 绘制曲面图（也就是海域的），因为船最低吃水深度为 5 英尺，需要在水域深度超过 5 英尺的范围才能避免搁浅，所以使用 `find` 函数先找到小于 5 英尺的位置，然后取出相关数据，运行 `example1_1.m` 文件，结果如图 9 所示：



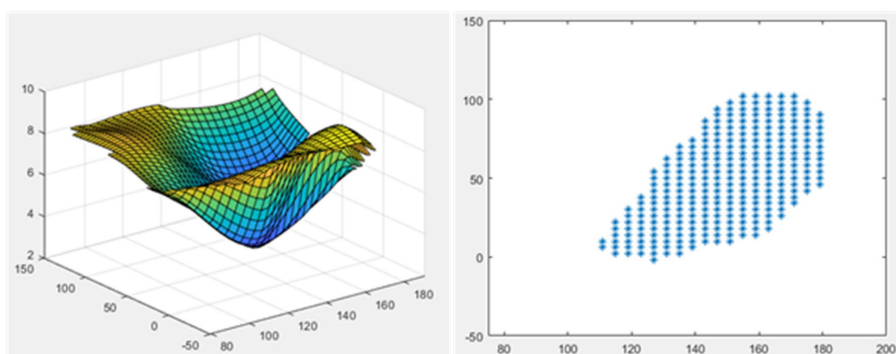


图 11. 海域三维图 (左) 水深危险区域图 (右)  
fig11. 3d map of sea area (left) Danger zone map of water depth (right)