



信阳师范学院  
数学与统计学院  
SCHOOL OF MATHEMATICS AND STATISTICS

# 第8章 MATLAB绘图与可视化

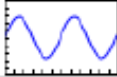
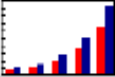

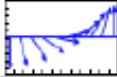


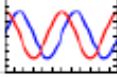
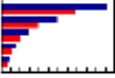

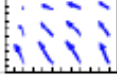


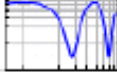
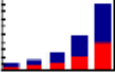



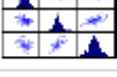

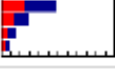
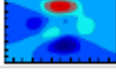

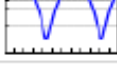

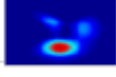


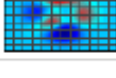

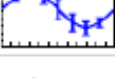

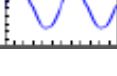



讲授人：牛言涛



日期：2020年3月27日

# 二维绘图函数汇总

Line Graphs	Bar Graphs	Area Graphs	Direction Graphs	Radial Graphs	Scatter Graphs
<a href="#">plot</a> 	<a href="#">bar</a> (grouped) 	<a href="#">area</a> 	<a href="#">feather</a> 	<a href="#">polar</a> 	<a href="#">scatter</a> 
<a href="#">plotyy</a> 	<a href="#">barh</a> (grouped) 	<a href="#">pie</a> 	<a href="#">quiver</a> 	<a href="#">rose</a> 	<a href="#">spy</a> 
<a href="#">loglog</a> 	<a href="#">bar</a> (stacked) 	<a href="#">fill</a> 	<a href="#">comet</a> 	<a href="#">compass</a> 	<a href="#">plotmatrix</a> 
<a href="#">semilogx</a> 	<a href="#">barh</a> (stacked) 	<a href="#">contourf</a> 		<a href="#">ezpolar</a> 	
<a href="#">semilogy</a> 	<a href="#">hist</a> 	<a href="#">image</a> 			
<a href="#">stairs</a> 	<a href="#">pareto</a> 	<a href="#">pcolor</a> 			
<a href="#">contour</a> 	<a href="#">errorbar</a> 	<a href="#">ezcontourf</a> 			
<a href="#">ezplot</a> 	<a href="#">stem</a> 				

# 1. fill填充图

## fill 函数创建彩色多边形。

`fill(X,Y,C)` 根据 `X` 和 `Y` 中的数据创建填充的多边形（顶点颜色由 `C` 指定）。`C` 是一个用作颜色图索引的向量或矩阵。如果 `C` 为行向量，`length(C)` 必须等于 `size(X,2)` 和 `size(Y,2)`；如果 `C` 为列向量，`length(C)` 必须等于 `size(X,1)` 和 `size(Y,1)`。必要时，`fill` 可将最后一个顶点与第一个顶点相连以闭合多边形。`X` 和 `Y` 的值可以是数字、日期时间、持续时间或分类值。

`fill(X,Y,ColorSpec)` 填充 `X` 和 `Y` 指定的二维多边形（颜色由 `ColorSpec` 指定）。

`fill(X1,Y1,C1,X2,Y2,C2,...)` 指定多个二维填充区。

`fill(...,'PropertyName',PropertyValue)` 允许您为补片图形对象指定属性名称和值。

`fill(ax,...)` 将在由 `ax` 指定的坐标区而不是当前坐标区 (`gca`) 中创建多边形。选项 `ax` 可以位于前面的语法中的任何输入参数组合之前。

`h = fill(...)` 返回由补片对象构成的向量。

# 1. fill填充图

%第4章 数值积分：通过桥梁的车流量数据

```
>> x=[0,2,4,5:9,10.5,11.5,12.5,14,16:24];
```

```
>> y = [20,25,30,35,40,50,155,92,60,62,60,43,59,140,120,68,60,80,55,65,20];
```

```
>> subplot(1,2,1)
```

```
>> fill(x,y,'y')
```

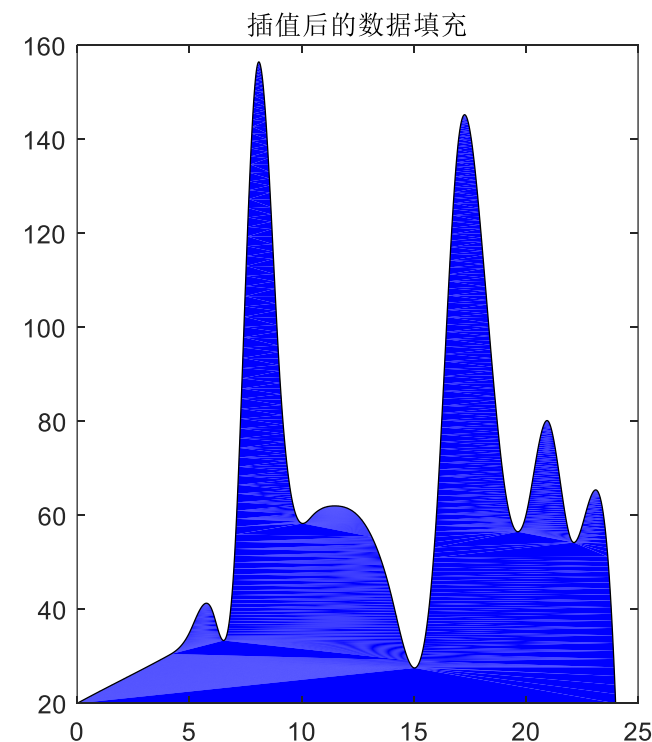
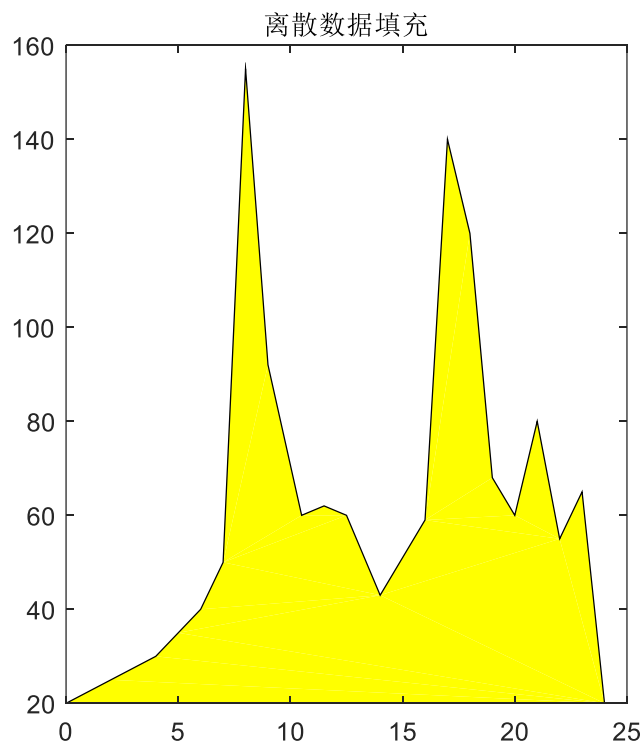
```
>> subplot(1,2,2)
```

```
>> xi = 0:1/60:24; %插值点
```

%通过三次样条插值求插值点函数值

```
>> yi = spline(x,y,xi);
```

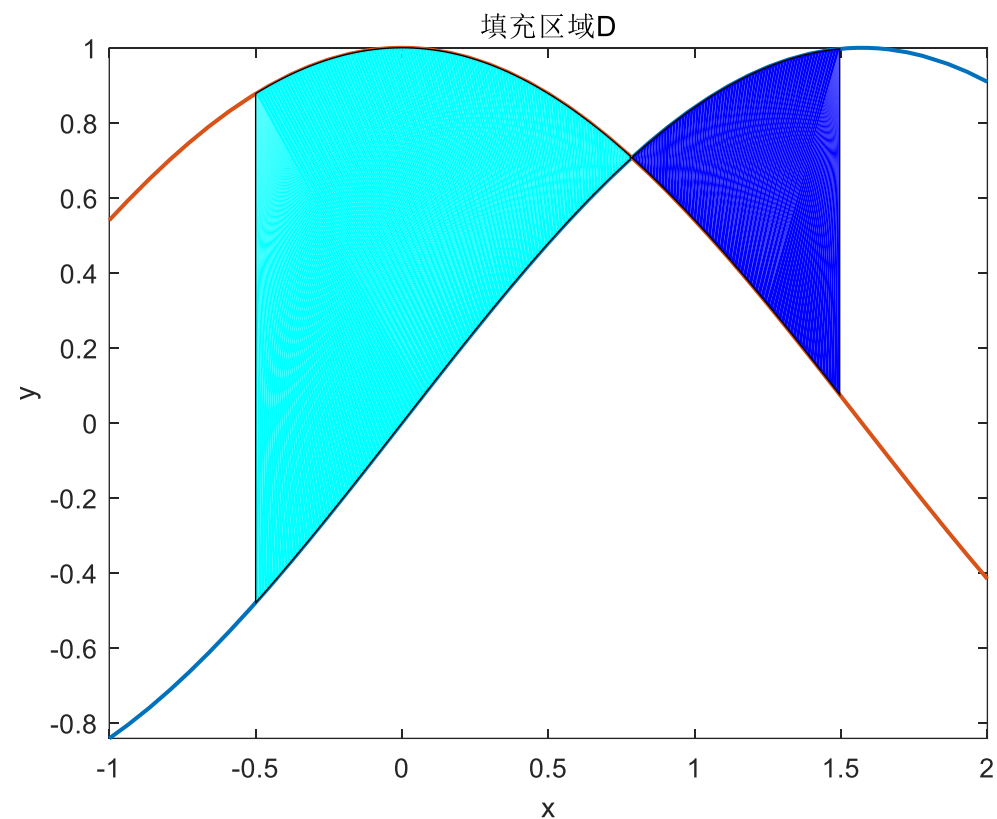
```
>> fill(xi,yi,'b')
```



# 1. fill填充图

由 $y_1 = \sin x$ ,  $y_2 = \cos x$ ,  $x_1 = -0.5$ ,  $x_2 = 1.5$  所围成的平面区域 $D$ 。

```
y1 = @(x)sin(x);  
y2 = @(x)cos(x);  
fplot(y1,[-1,2],'-','LineWidth',1.5)  
hold on  
fplot(y2,[-1,2],'-','LineWidth',1.5)  
[x0,y0]=fzero(@(x)sin(x) - cos(x),[-0.5,1.5]); %求两条线的交点  
x1 = -0.5:0.01:x0;  
x2 = x0:0.01:1.5;  
fill([x1,flipr(x1)],[y1(x1),flipr(y2(x1))],'c') %填充颜色  
fill([x2,flipr(x2)],[y2(x2),flipr(y1(x2))],'b') %填充颜色
```



## 2. 二维饼图pie

`pie(X)` 使用  $X$  中的数据绘制饼图。饼图的每个扇区代表  $X$  中的一个元素。

- 如果  $\text{sum}(X) \leq 1$ ,  $X$  中的值直接指定饼图扇区的面积。如果  $\text{sum}(X) < 1$ , `pie` 仅绘制部分饼图。
- 如果  $\text{sum}(X) > 1$ , 则 `pie` 通过  $X/\text{sum}(X)$  对值进行归一化, 以确定饼图的每个扇区的面积。
- 如果  $X$  为 `categorical` 数据类型, 则扇区对应于类别。每个扇区的面积是类别中的元素数除以  $X$  中的元素数的结果。

`pie(X, explode)` 将扇区从饼图偏移一定位置。`explode` 是一个由与  $X$  对应的零值和非零值组成的向量或矩阵。`pie` 函数仅将对应于 `explode` 中的非零元素的扇区偏移一定的位置。

如果  $X$  为 `categorical` 数据类型, 则 `explode` 可以是由对应于类别的零值和非零值组成的向量, 或者是由要偏移的类别名称组成的元胞数组。

`pie(X, labels)` 指定扇区的文本标签。 $X$  必须是数值数据类型。标签数必须等于  $X$  中的元素数。

`pie(X, explode, labels)` 偏移扇区并指定文本标签。 $X$  可以是数值或分类数据类型。对于数值数据类型的  $X$ , 标签数必须等于  $X$  中的元素数。对于分类数据类型的  $X$ , 标签数必须等于分类数。

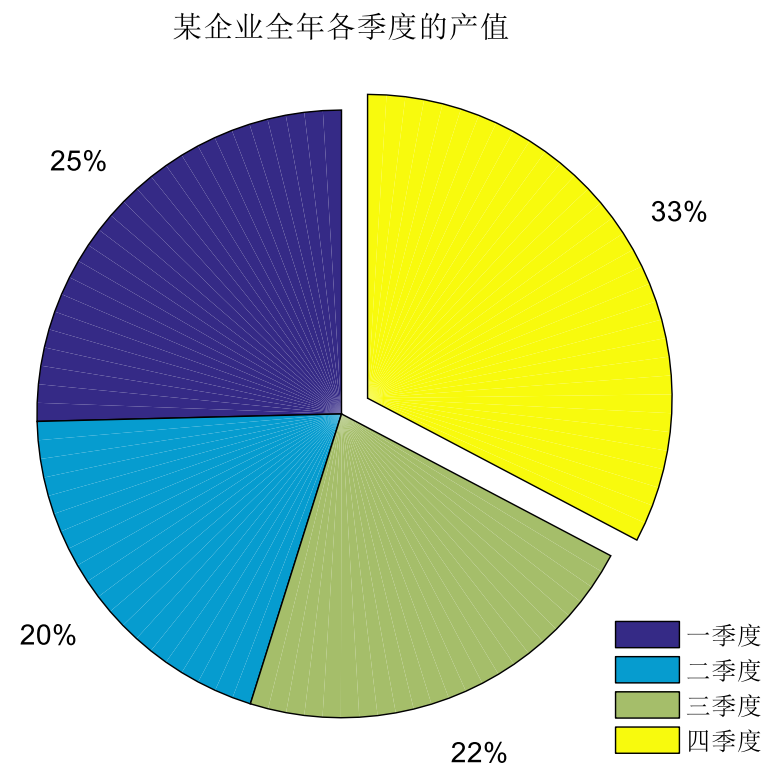
`pie(ax, ____)` 将图形绘制到 `ax` 指定的坐标区中, 而不是当前坐标区 (`gca`) 中。选项 `ax` 可以位于前面的语法中的任何输入参数组合之前。

`p = pie(____)` 返回一个由补片和文本图形对象组成的向量。该输入可以是先前语法中的任意输入参数组合。

## 2. 二维饼图pie

例： 绘制图形，某企业全年各季度的产值(单位：万元)分别为：2347,1827,2043,3025，试用饼图作统计分析。

```
XX = [2347,1827,2043,3025];  
[xmax,ymax] = max(XX);  
explode = zeros(1,4);  
explode(ymax) = 1;  
pie(XX,explode)  
title('某企业全年各季度的产值');  
legend('一季度','二季度','三季度','四季度','Location','best');  
legend('boxoff')
```



### 3. image生成图像

`image(C)` 会将数组 `C` 中的数据显示为图像。`C` 的每个元素指定图像的 1 个像素的颜色。生成的图像是一个  $m \times n$  像素网格，其中  $m$  和  $n$  分别是 `C` 中的行数和列数。这些元素的行索引和列索引确定了对应像素的中心。

`image(x,y,C)` 指定图像位置。使用 `x` 和 `y` 可指定与 `C(1,1)` 和 `C(m,n)` 对应的边角的位置。要同时指定两个边角，请将 `x` 和 `y` 设置为二元素向量。要指定第一个边角并让 `image` 确定另一个，请将 `x` 和 `y` 设为标量值。图像将根据需要进行拉伸和定向。

`image('CData',C)` 将图像添加到当前坐标区中而不替换现有绘图。此语法是 `image(C)` 的低级版本。有关详细信息，请参阅[图像的高级与低级版本](#)。

`image('XData',x,'YData',y,'CData',C)` 指定图像位置。此语法是 `image(x,y,C)` 的低级版本。

`image( ___,Name,Value)` 使用一个或多个名称-值对组参数指定图像属性。您可以使用先前语法中的任意输入参数组合指定图像属性。

`image(ax, ___)` 将在由 `ax` 指定的坐标区中而不是当前坐标区 (`gca`) 中创建图像。选项 `ax` 可以位于前面的语法中的任何输入参数组合之前。

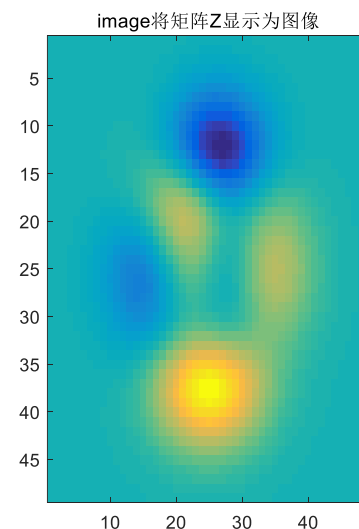
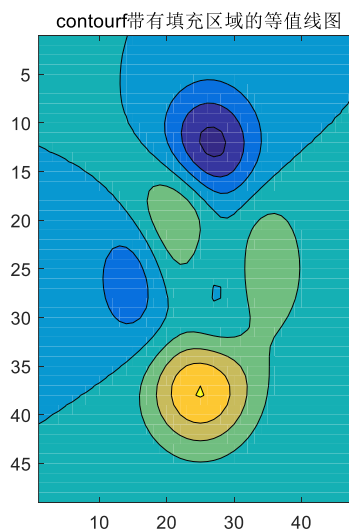
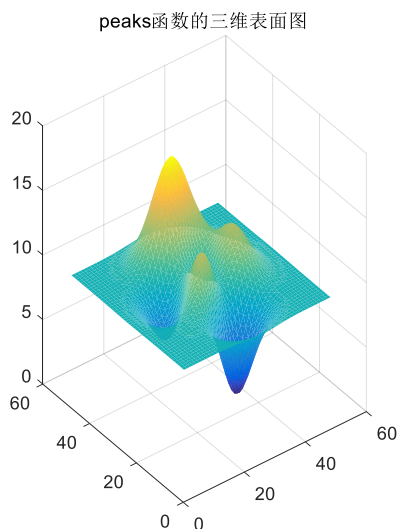
`im = image( ___)` 返回创建的 `Image` 对象。使用 `im` 在创建图像后设置图像的属性。您可以使用先前语法中的任意输入参数组合指定此输出。如需图像属性和说明的列表，请参阅 [Image 属性](#)。

```
subplot(1,3,3)
```

```
%将值的范围缩放到当前颜色图的完整范围
```

```
image(Z,'CDataMapping','scaled')
```

```
title('image将矩阵Z显示为图像')
```





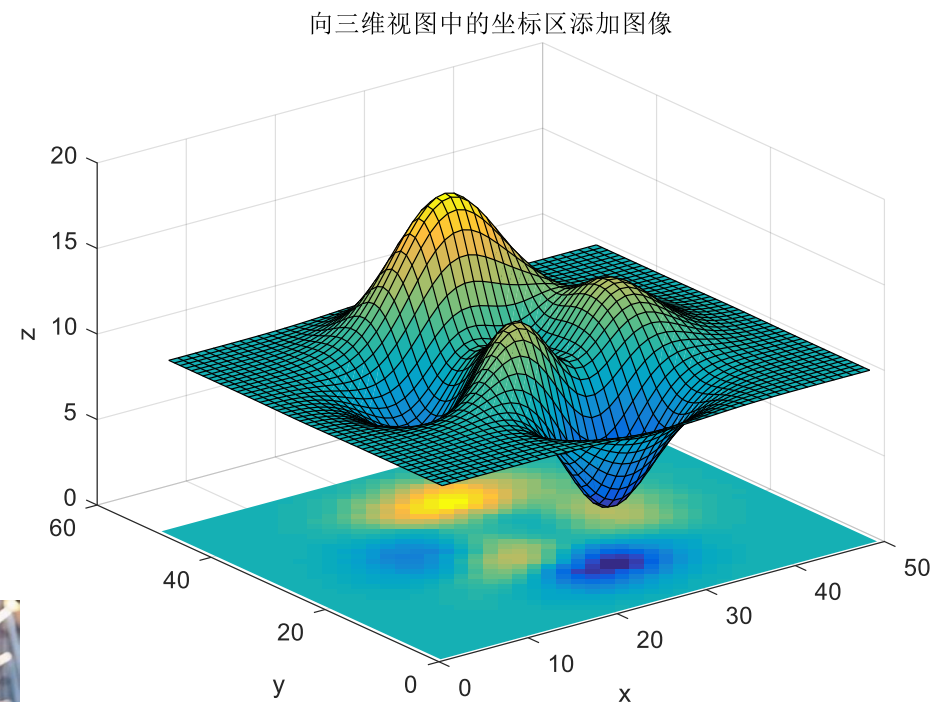
### 3. image生成图像

%创建一个曲面图。然后，在曲面下添加一个图像。image  
在 xy 平面中显示该图像。

```
>> Z = 10 + peaks;  
>> surf(Z)  
>> hold on  
>> image(Z,'CDataMapping','scaled')  
>> title('向三维视图中的坐标区添加图像')
```

%图片读入函数

```
>> img = imread('gaoyuanyuan.png');  
>> image(img)  
>> axis off
```



### 3. image生成图像

```
>> img = imread('gaoyuanyuan.png'); %%图片读入函数
```

```
>> subplot(2,2,1)
```

```
>> imshow(img)
```

```
>> subplot(2,2,2)
```

```
>> img_gray = rgb2gray(img); %图片转为灰度图像
```

```
>> imshow(img_gray)
```

```
>> subplot(2,2,3)
```

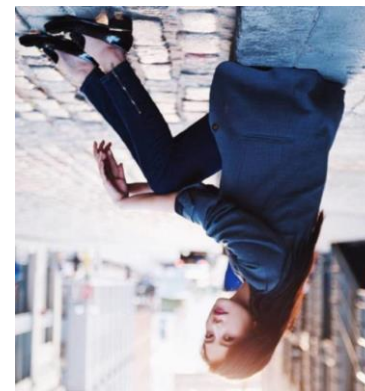
```
>> imgf = fliplr(img);
```

```
>> imshow(imgf)
```

```
>> subplot(2,2,4)
```

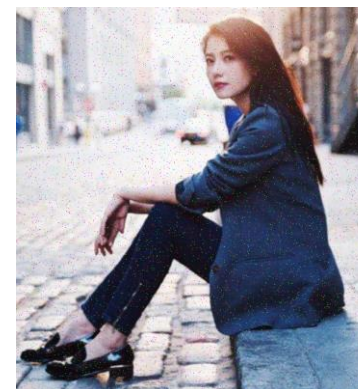
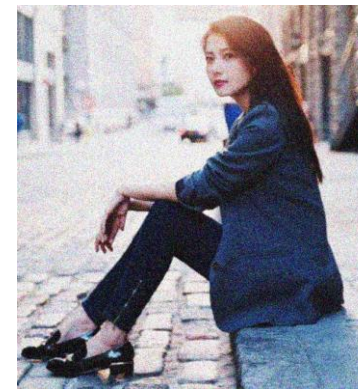
```
>> imgud = flipud(img);
```

```
>> imshow(imgud)
```



### 3. image生成图像

```
>> subplot(2,2,1)
>> imshow(img)
>> subplot(2,2,2)
>> J1 = imnoise(img,'gaussian',0,0.02); %高斯噪音
>> imshow(J1)
>> subplot(2,2,3)
>> J3 = imnoise(img,'speckle',0.02); %乘性噪音
>> imshow(J3)
>> subplot(2,2,4)
>> J2 = imnoise(img,'salt & pepper',0.02); %椒盐噪音
>> imshow(J2)
```



## 4. area区域图

区域图将  $Y$  中的元素显示为一个或多个曲线并填充每个曲线下方的区域。如果  $Y$  为矩阵，则曲线堆叠在一起，显示每行元素占每个  $x$  区间的曲线总高度的相对量。

`area(Y)` 绘制向量  $Y$  或将矩阵  $Y$  中每一列作为单独曲线绘制并堆叠显示。 $x$  轴自动缩放到 `1:size(Y,1)`。 $Y$  中的值可以是数值或持续时间值。

`area(X,Y)` 绘制  $Y$  对  $X$  的图，并填充  $0$  和  $Y$  之间的区域。 $X$  的值可以是数值、日期时间、持续时间或分类值。

- 如果  $Y$  是向量，则将  $X$  指定为由递增值组成的向量，其长度等于  $Y$ 。如果  $X$  的值不增加，则 `area` 将在绘制之前对值进行排序。
- 如果  $Y$  是矩阵，则将  $X$  指定为由递增值组成的向量，其长度等于  $Y$  的行数。`area` 将  $Y$  的列绘制为填充区域。对于每个  $X$ ，最终结果是  $Y$  行的相应值的和。您还可以将  $X$  指定为大小等于  $Y$  的矩阵。为了避免  $X$  为矩阵时出现意外输出，请将  $X$  的列指定为重复列。

`area(...,basevalue)` 指定区域填充的基值。默认 `basevalue` 为  $0$ 。将基值指定为数值。

`area(...,Name,Value)` 使用一个或多个名称-值对组参数修改区域图。

`area(ax,...)` 将图形绘制到  $ax$  坐标区中，而不是当前坐标区 (`gca`) 中。

`ar = area(...)` 返回一个或多个 `Area` 对象。`area` 函数将为向量输入参数创建一个 `Area` 对象。它会为矩阵输入参数的每一列创建一个对象。

创建  $m \times n$  矩阵的区域图会创建  $n$  个区域序列对象（即每列一个），而  $1 \times n$  向量创建一个区域序列对象。

您在单个区域序列对象中设置的一些区域序列对象属性会设置图形中的所有区域序列对象的值。有关特定属性的信息，请参阅 [Area 属性](#)。

## 4. area区域图

```
>> A = [2 6 9;7 5 7;4 9 4;9 9 3;9 7 4]
```

```
A =
```

```
2    6    9
7    5    7
4    9    4
9    9    3
9    7    4
```

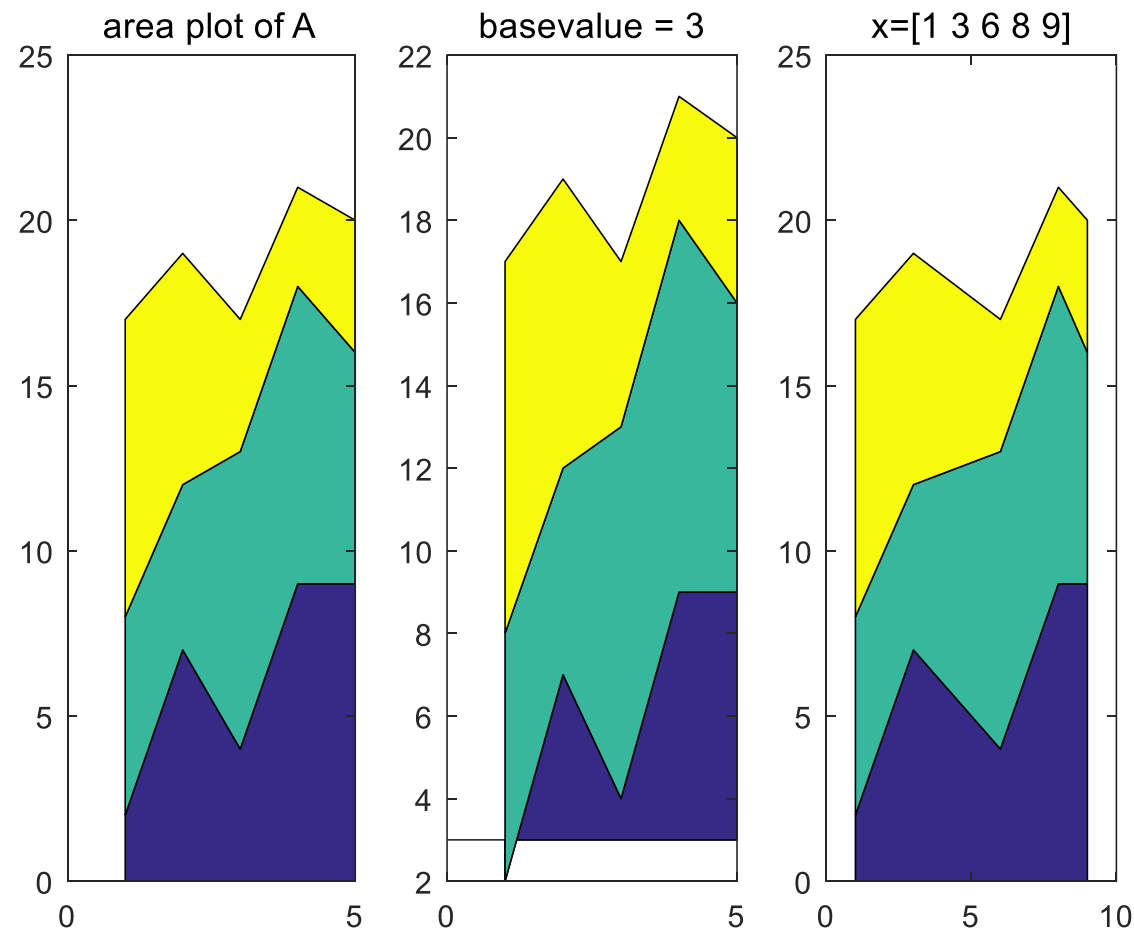
```
>> subplot(1,3,1),area(A),title('area plot of A');
```

```
%基线
```

```
>> subplot(1,3,2),area(A,3),title('basevalue = 3');
```

```
%绘制y对x的图
```

```
>> subplot(1,3,3),area([1 3 6 8 9],A),title('x=[1 3 6 8 9]');
```





## 4. area区域图

`x=1:2:9;` % 注意：自变量要单调变化

`y=magic(5);` % 各因素的相对贡献份额，每一列相当于一个因素

`colormap(spring);` % 控制图的用色

`%area(y)`则以列下标作为自变量,第三个参数为基准线

`area(x,y,4,'LineStyle',':');`

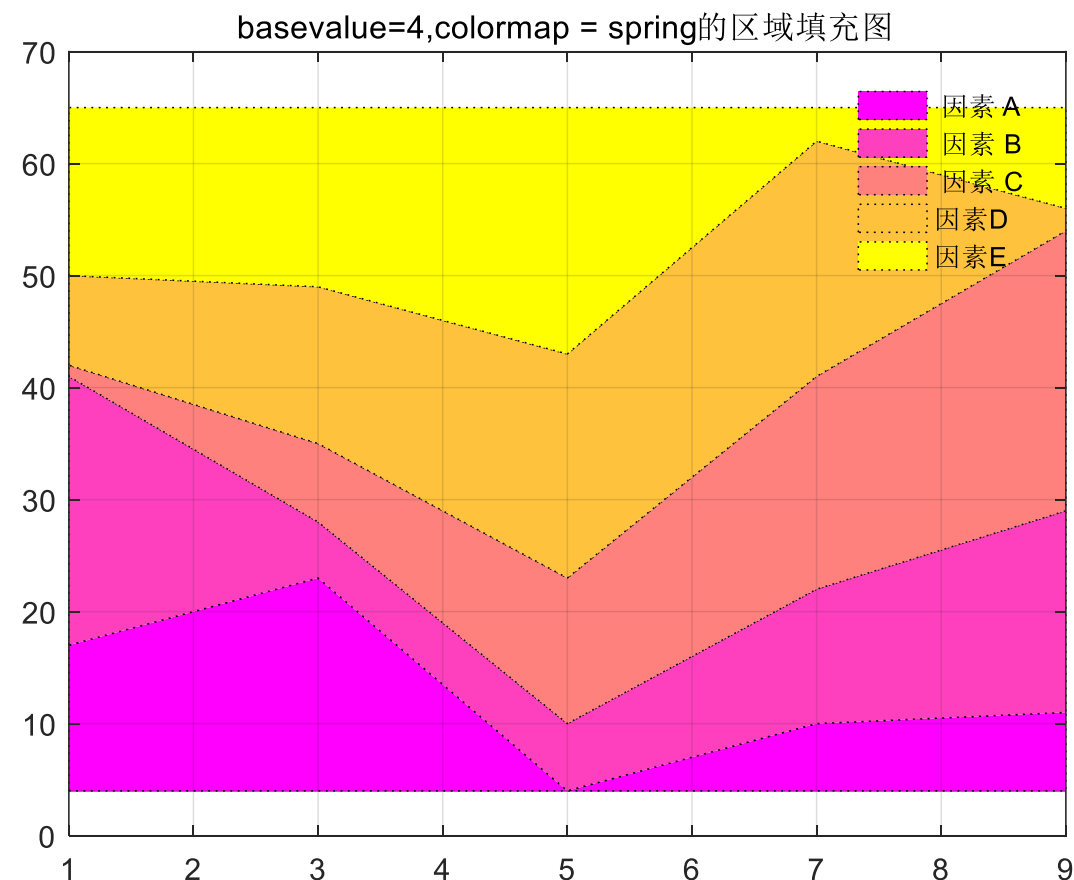
`set(gca,'layer','top');` %图层设置为top层，显示网格

`title('basevalue=4,colormap = spring的区域填充图');`

`legend('因素 A','因素 B','因素 C','因素D','因素E');`

`legend('boxoff')`

`grid on;`



## 5. pcolor伪彩图

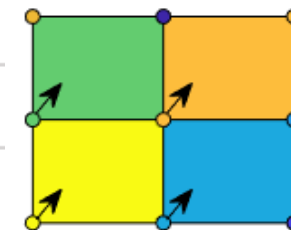
常用于以二维平面图表现三维图形的效果，用颜色表示三维图形的高度。

`pcolor(C)` 使用矩阵 `C` 中的值创建一个伪彩图。伪彩图以彩色单元（称为面）阵列形式显示矩阵数据。MATLAB<sup>®</sup> 在  $x$ - $y$  平面上创建该图的单一着色平面图。该平面由对应于各面的角（即顶点）的  $x$  坐标和  $y$  坐标的网格定义。网格覆盖区域  $X=1:n$  和  $Y=1:m$ ，其中  $[m,n] = \text{size}(C)$ 。矩阵 `C` 指定顶点处的颜色。每个面的颜色取决于其四个周围顶点之一的颜色。在这四个顶点中， $x$ - $y$  网格中出现的第一个顶点决定该面的颜色。

`pcolor(X,Y,C)` 指定顶点的  $x$  坐标和  $y$  坐标。`C` 的大小必须与  $x$ - $y$  坐标网格的大小匹配。例如，如果  $x$  和  $Y$  定义一个  $m \times n$  网格，则 `C` 必须为  $m \times n$  矩阵。

`pcolor(ax, __)` 指定绘图的目标坐标区。指定 `ax` 作为上述任何语法中的第一个参数。

`s = pcolor(__)` 返回 `Surface` 对象。创建绘图后，使用 `s` 设置绘图属性。有关属性列表，请参阅 [Surface 属性](#)。



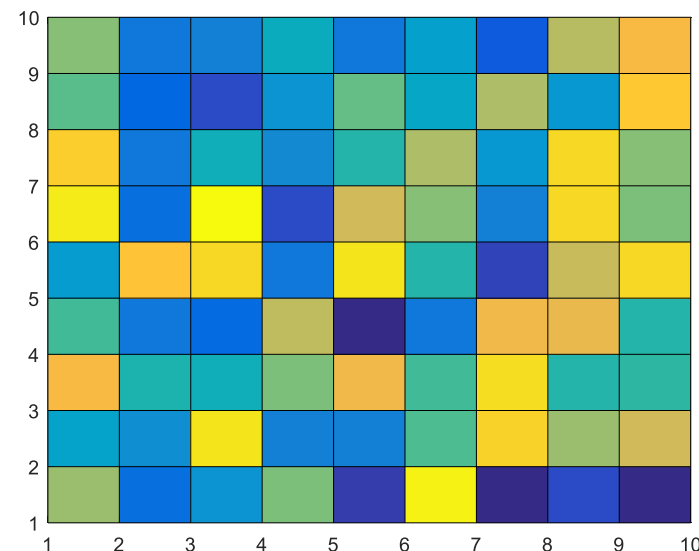
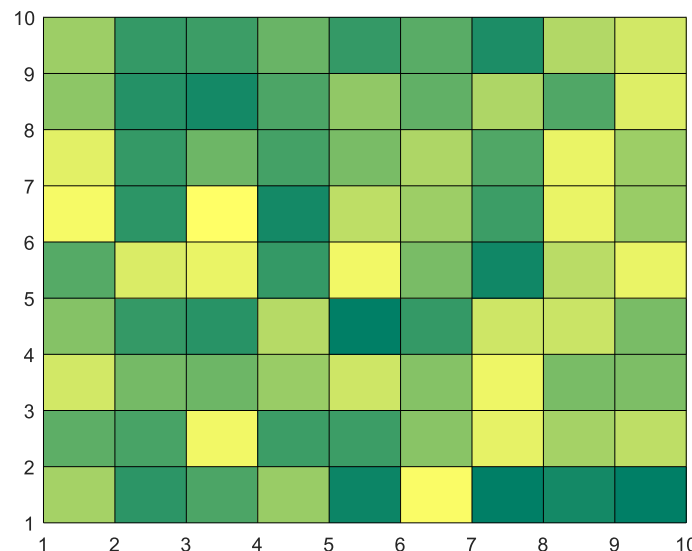
```
>> C = randi([10,100],10);
```

```
>> pcolor(C)
```

```
>> colormap(summer)
```

```
>> pcolor(C)
```

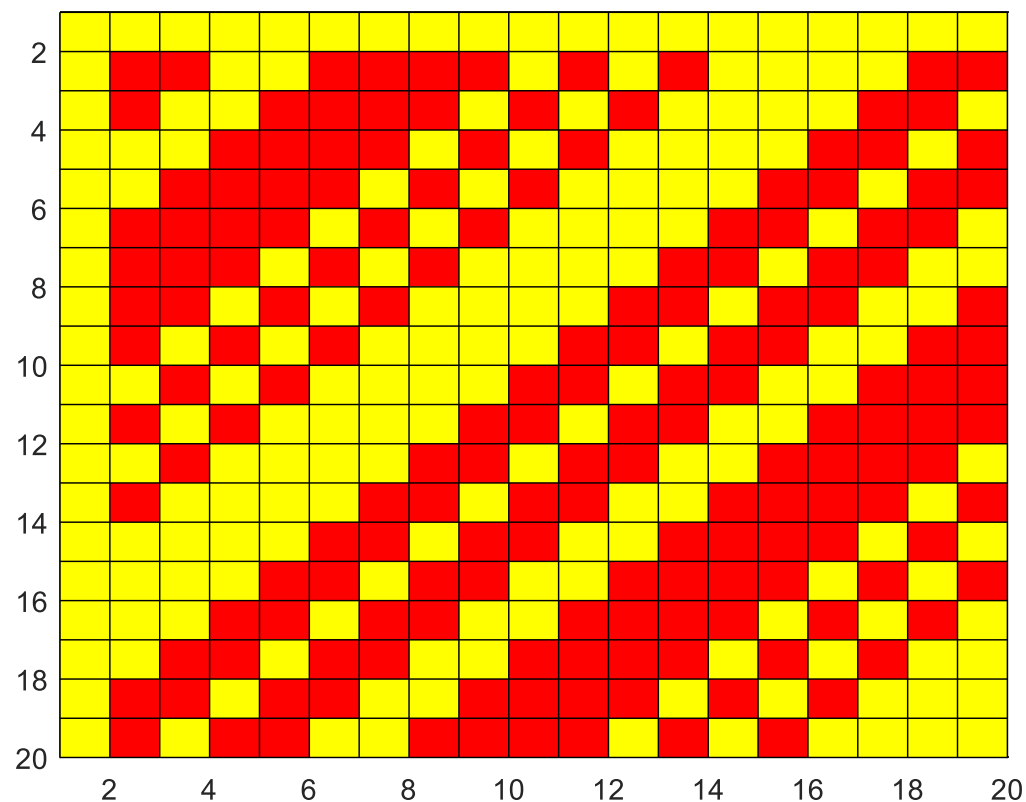
```
>> colormap default
```



## 5. pcolor伪彩图

- Hadamard 矩阵的元素是 1 或 -1。可以使用双色颜色图来可视化此矩阵。
- 创建一个  $20 \times 20$  Hadamard 矩阵。然后用autumn颜色图绘制该矩阵。使用 axis 函数反转 y 轴的方向，并将轴线设置为相等的长度。

```
>> C = hadamard(20);  
>> pcolor(C)  
>> colormap autumn %设置当前颜色图为autumn  
>> axis ij %反转y轴的方向
```





## 5. pcolor伪彩图

```
[X,Y,C] = peaks(30);
```

```
subplot(2,2,1)
```

```
mesh(X,Y,C)
```

```
subplot(2,2,2)
```

```
contourf(X,Y,C,'ShowText','on')
```

```
subplot(2,2,3)
```

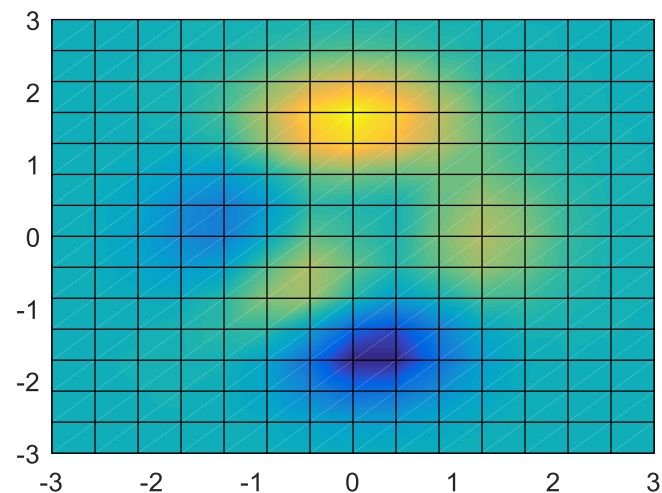
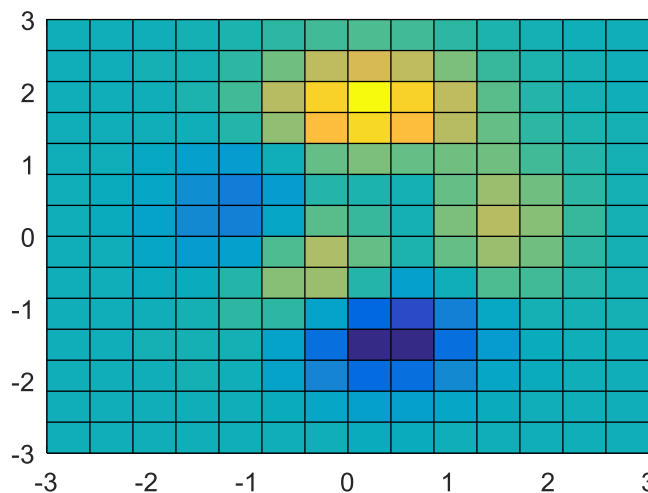
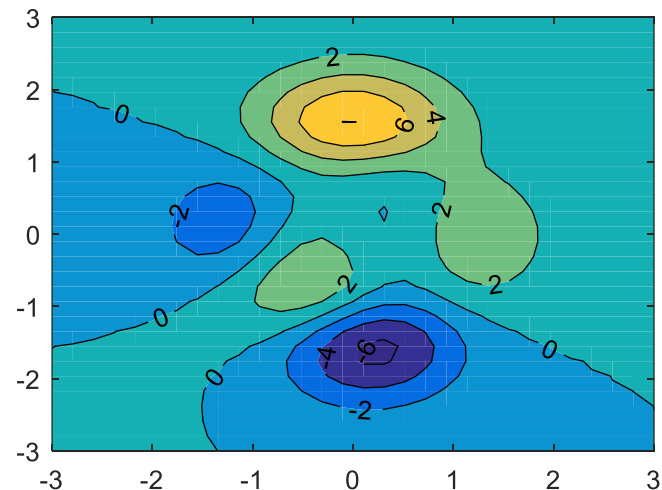
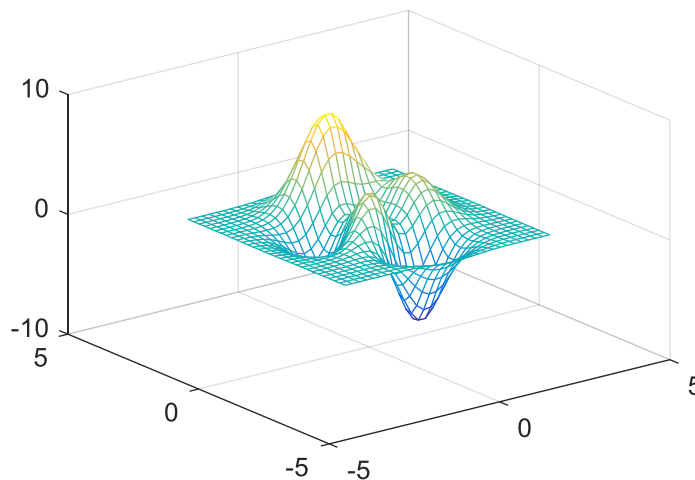
```
[X,Y,C] = peaks(15);
```

```
pcolor(X,Y,C);
```

```
subplot(2,2,4)
```

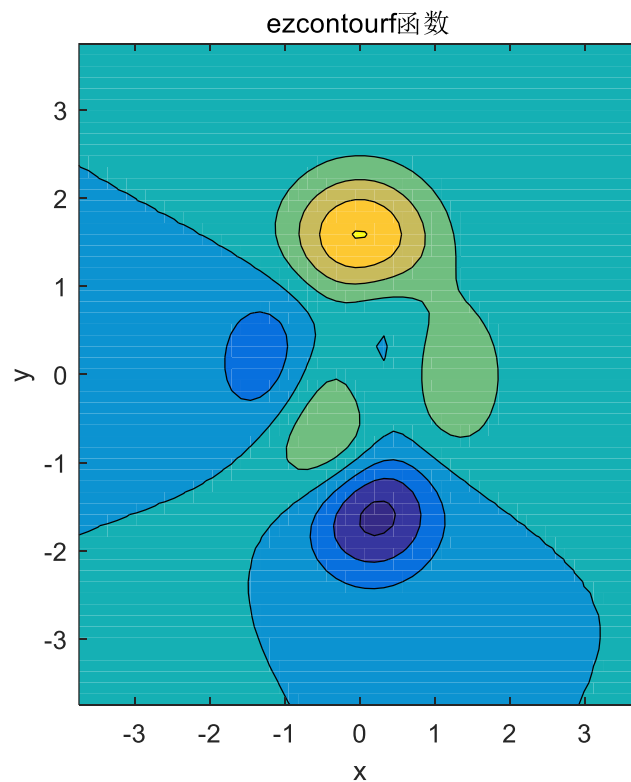
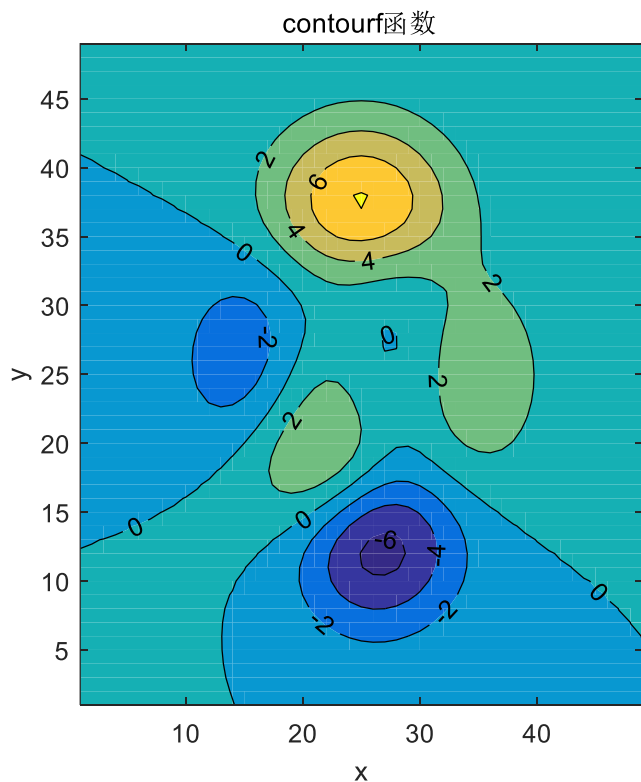
```
h = pcolor(X,Y,C);
```

```
h.FaceColor = 'interp';
```



## 6. contourf和ezcontourf等值线填充图

```
>> subplot(1,2,1)
>> contourf(peaks,'ShowText','on'); title('contourf函数')
>> subplot(1,2,2)
>> f = @(x,y) 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) - 1/3*exp(-(x+1).^2 - y.^2);
>> ezcontourf(f); title('ezcontourf函数')
```





---

# 感谢聆听

---