

第 3 章

线形图

线形图用直线或曲线表现数据。根据绘图数据的组数,线形图分为简单线形图和复合线形图;根据坐标系的类型,分为直角坐标系线形图和极坐标系线形图;根据绘图数据维度的不同,分为二维线形图和三维线形图。另外有一些特殊的表现方法,例如用球面表示点标记、给图添加背景颜色和标注等。

例 015 简单线形图

简单线形图用直线或曲线表现向量数据,如图 3-1 所示。图中在几个关键点添加了线标注和点标注。

[图表效果]

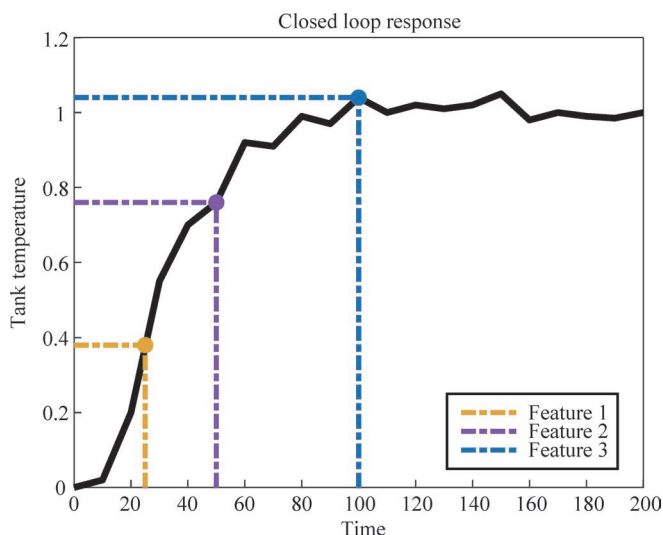


图 3-1 简单线形图

[代码实现]

```

1. % 简单线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x = 0:10:200; % 绘图数据
4. y = [0,0.02,0.2,0.55,0.7,0.76,0.92,0.91,0.99,0.97,1.04, ...
5.     1,1.02,1.01,1.02,1.05,0.98,1,0.99,0.985,1];
6. h = plot(x,y,'-'); % 绘制线形图
7. h.LineWidth = 2.5; % 修改属性
8. h.Color = 'k';
9.
10. % 标注
11. line([25,25],[0,0.38],'LineWidth',2,'LineStyle', ...
12.     '-.','Color',[0.93,0.69,0.13]) % 绘制第 1 组标注线,竖线
13. line([0,25],[0.38,0.38],'LineWidth',2,'LineStyle', ...
14.     '-.','Color',[0.93,0.69,0.13]) % 绘制第 1 组标注线,横线
15. hold on % 叠加绘图
16. plot(25,0.38,'o','MarkerFaceColor',[0.93,0.69,0.13], ...
17.     'MarkerEdgeColor',[0.93,0.69,0.13],'MarkerSize',6) % 绘制第 1 组标注线,圆点
18.
19. line([50,50],[0,0.76],'LineWidth',2,'LineStyle', ...
20.     '-.','Color',[0.72,0.27,1]) % 绘制第 2 组标注线,竖线
21. line([0,50],[0.76,0.76],'LineWidth',2,'LineStyle', ...
22.     '-.','Color',[0.72,0.27,1]) % 绘制第 2 组标注线,横线
23. plot(50,0.76,'o','MarkerFaceColor',[0.72,0.27,1], ...
24.     'MarkerEdgeColor',[0.72,0.27,1],'MarkerSize',6) % 绘制第 2 组标注线,圆点
25.
26. line([100,100],[0,1.04],'LineWidth',2,'LineStyle', ...
27.     '-.','Color',[0,0.45,0.74]) % 绘制第 3 组标注线,竖线
28. line([0,100],[1.04,1.04],'LineWidth',2,'LineStyle', ...
29.     '-.','Color',[0,0.45,0.74]) % 绘制第 3 组标注线,横线
30. plot(100,1.04,'o','MarkerFaceColor',[0,0.45,0.74], ...
31.     'MarkerEdgeColor',[0,0.45,0.74],'MarkerSize',6) % 绘制第 3 组标注线,圆点
32.
33. % 绘制图例,不用默认图例
34. line([136,156],[0.1,0.1],'LineWidth',2,'LineStyle', ...
35.     '-.','Color',[0,0.45,0.74])
36. line([136,156],[0.15,0.15],'LineWidth',2,'LineStyle', ...
37.     '-.','Color',[0.72,0.27,1])
38. line([136,156],[0.2,0.2],'LineWidth',2,'LineStyle', ...
39.     '-.','Color',[0.93,0.69,0.13])
40. text(161,0.1,'Feature 3','FontSize',7)
41. text(161,0.15,'Feature 2','FontSize',7)
42. text(161,0.2,'Feature 1','FontSize',7)
43. rectangle('Position',[131,0.05,60,0.2],'LineWidth',1)
44.
45. % 外框线
46. line([0,200],[1.2,1.2],'Color','k','LineWidth',0.25)
47. line([200,200],[0,1.2],'Color','k','LineWidth',0.25)
48.
49. xlim([0,200])
50. ylim([0,1.2])
51. title('Closed loop response','FontSize',8)

```

```

52. ylabel('Tank temperature','FontSize',7)
53. xlabel('Time','FontSize',7)
54. ax = gca;                                     % 设置坐标系为当前坐标系
55. ax.FontSize = 6;                             % 修改坐标系刻度标签的字体大小
56. ax.TickDir = "out";
57. hold off                                     % 取消叠加绘图
58. box off                                       % 不绘制外框
59.
60. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 400 300])
61. exportgraphics(gcf, 'sam03_01.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
62. exportgraphics(gcf, 'sam03_01.pdf', 'ContentType', 'vector')                 % 保存为 pdf 文件

```

[代码说明]

代码第 6~8 行绘制线形图并修改属性,第 10~31 行绘制 3 个点处的线标记和点标记,第 33~43 行绘制图例。

[技术要点]

用 plot 函数绘制线形图。标注和图例直接绘制即可。

给 MATLAB 二维图表添加外框时,顶部和右侧的外框线上会显示刻度线。如果希望去掉刻度线,可以不绘制外框,自己用直线段绘制顶部和右侧的外框线。

例 016 复合线形图

复合线形图用多条折线表现多组数据,如图 3-2 所示。如果在每个数据点显示点标记,常称为点线图。

[图表效果]

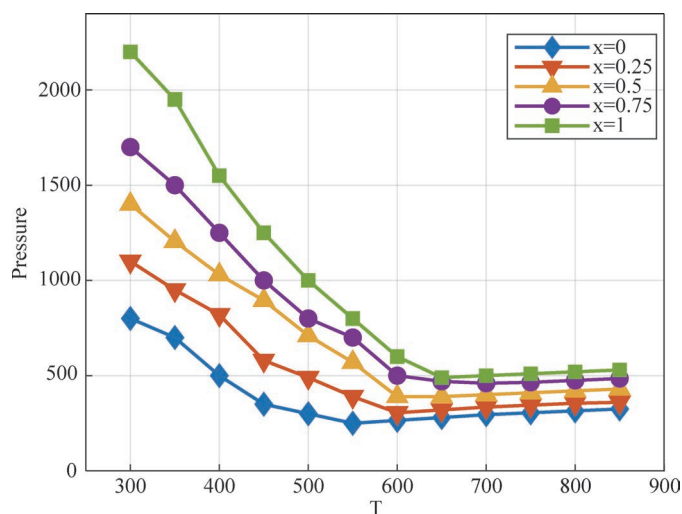


图 3-2 复合线形图

[代码实现]

```

1. % 复合线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x = 300:50:850; % 绘制图数据,5 组
4. y = [800,700,500,350,300,250,265,280,295,305,315,325;
5.      1100,950,820,580,490,390,305,320,335,345,355,360;
6.      1400,1205,1030,895,710,570,390,390,400,410,420,430;
7.      1700,1500,1250,1000,800,700,500,470,460,465,475,485;
8.      2200,1950,1550,1250,1000,800,600,490,500,510,520,530];
9. h = plot(x,y); % 绘制复合线形图
10. h(1).Marker = "diamond"; % 修改各序列的点标记类型
11. h(2).Marker = "v";
12. h(3).Marker = "^";
13. h(4).Marker = "o";
14. h(5).Marker = "square";
15. colororder gem % 指定颜色序列
16. for i = 1:length(h) % 修改各线形序列的线框、标记大小和标记填充色
17.     h(i).LineWidth = 1.25;
18.     h(i).MarkerSize = 6;
19.     h(i).MarkerFaceColor = h(i).Color;
20. end
21.
22. % 自己绘制外框线
23. xlim([250,900])
24. ylim([0,2400])
25. line([250,900],[2400,2400],'Color','k','LineWidth',0.25)
26. line([900,900],[0,2400],'Color','k','LineWidth',0.25)
27.
28. ylabel('Pressure','FontSize',7)
29. xlabel('T','FontSize',7)
30. ax = gca; % 设置坐标系为当前坐标系
31. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
32. labels = ["x = 0","x = 0.25","x = 0.5","x = 0.75","x = 1"]; % 刻度标签
33. legend(ax, labels) % 设置刻度标签
34. grid on % 添加网格
35. box off % 不绘制外框
36.
37. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
38. exportgraphics(gcf,'sam03_02.png','ContentType','image','Resolution',300) % 保存为 png 文件
39. exportgraphics(gcf,'sam03_02.pdf','ContentType','vector') % 保存为 pdf 文件

```

[代码说明]

代码第 3~8 行为绘图数据,第 9~20 行绘制复合线形图并设置属性,第 22~26 行绘制顶部和右侧的外框线。

[技术要点]

注意代码第 4~8 行给定的 y 数据是一个矩阵,矩阵每一行的数据对应复合线形图中的一条曲线,或者叫一个序列。

第 15 行用 colororder 函数指定颜色序列,这样可以自动给一系列对象设置颜色。MATLAB 提供了多个颜色序列,如表 3-1 所示,可以直接引用名称进行使用,如本例中使用

的是第 1 个颜色序列 gem。

表 3-1 MATLAB 提供的颜色序列

名 称	颜 色 序 列
gem	
gem12	
glow	
glow12	
sail	
reef	
meadow	
dye	
earth	

如果 MATLAB 提供的颜色序列不够用,还可以自定义颜色序列。自定义颜色序列的示例代码如下:

```
newcolors = [0 1 1;1 1 0;1 0 1];  
colororders(newcolors)
```

newcolors 变量定义颜色矩阵,然后用 colororders 函数使用该变量。如果对象个数大于颜色序列提供的颜色种数,MATLAB 会循环使用这些颜色。

例 017 复合线形图+线面标注

学术期刊上很多图表都不是简单的某种图表,而是经常添加背景色、标注等元素或进行图表组合等。图 3-3 中给复合线形图添加了线标注和面标注,可以突出数据特征,增加图表可读性。

[图表效果]

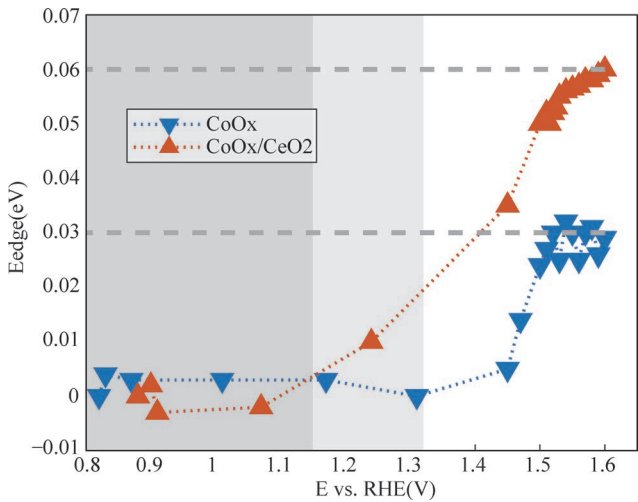


图 3-3 给复合线形图添加线标注和面标注

[代码实现]

```

1. % 复合线形图 + 线面标注
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x1 = [0.82,0.83,0.87,1.01,1.17,1.31,1.45,1.47, ...
4.      1.5,1.51,1.52,1.53,1.54,1.55,1.56,1.57, ...
5.      1.58,1.59,1.6]; % 绘图数据
6. y1 = [0.0,0.004,0.003,0.003,0.003,0.0,0.005,0.014, ...
7.      0.024,0.027,0.03,0.025,0.032,0.03,0.025,0.03, ...
8.      0.031,0.026,0.029];
9. x2 = [0.88,0.9,0.91,1.07,1.24,1.45,1.5,1.51, ...
10.     1.515,1.52,1.525,1.53,1.54,1.55,1.56,1.57, ...
11.     1.58,1.59,1.6]; % 绘图数据 2
12. y2 = [0.0,0.002,-0.003,-0.002,0.01,0.035,0.05,0.052, ...
13.     0.05,0.052,0.053,0.055,0.056,0.0565,0.057,0.058, ...
14.     0.058,0.059,0.06];
15. xlim([0.8 1.65]) % 设置 x 轴的取值范围
16. ylim([-0.01 0.07]) % 设置 y 轴的取值范围
17.
18. rectangle('Position',[0.8 -0.01 0.35 0.08],'FaceColor',[0.8 0.8 0.8],'EdgeColor','none')
    % 背景 1
19. rectangle('Position',[1.15 -0.01 0.17 0.08],'FaceColor',[0.9 0.9 0.9],'EdgeColor','none')
    % 背景 2
20.
21. h1 = line(x1,y1); % 绘制线形 1
22. h1.LineWidth = 1.25; % 修改属性
23. h1.LineStyle = ":";
24. h1.Marker = "v";
25. h1.MarkerSize = 7;
26. h1.MarkerFaceColor = '#F4CE91';
27. h1.MarkerFaceColor = h1.Color;
28.
29. h2 = line(x2,y2); % 绘制线形 2
30. h2.LineWidth = 1.25; % 修改属性
31. h2.LineStyle = ":";
32. h2.Marker = "^";
33. h2.MarkerSize = 7;
34. h2.MarkerFaceColor = '#0000FF';
35. h2.MarkerFaceColor = h2.Color;
36.
37. line([1.6 0.8],[0.03 0.03],'LineStyle','--','Color',[0.6 0.6 0.6],'LineWidth',2)
    % 横虚线 1
38. line([1.6 0.8],[0.06 0.06],'LineStyle','--','Color',[0.6 0.6 0.6],'LineWidth',2)
    % 横虚线 2
39.
40. rectangle('Position',[0.8 -0.01 0.85 0.08],'EdgeColor','k') % 绘制外框
41.
42. ylabel('Eedge(eV)','FontSize',7)
43. xlabel('E vs. RHE(V)','FontSize',7)
44. ax = gca; % 设置坐标系为当前坐标系

```

```

45. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
46. labels = ["CoOx", "CoOx/CeO2"];
47. lgd = legend(ax, labels); % 设置刻度标签
48. lgd.BackgroundAlpha = 0.5; % 图例区域半透明
49. lgd.Position = [0.25, 0.65, 0.1, 0.1]; % 设置图例的位置和大小
50.
51. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 400 300])
52. exportgraphics(gcf, 'sam03_03.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
53. exportgraphics(gcf, 'sam03_03.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

[代码说明]

代码第3~14行为绘图数据,第18、19行绘制面标注,第21~35行绘制曲线,第37、38行绘制线标注,第40行绘制外框。

[技术要点]

本例全部使用低级函数绘图。低级函数使用叠加方式绘图,所以不必使用 hold on 语句。需要注意的是绘图的顺序,先绘制面标注,再绘制其他。

本例中图例是自动生成的,第48行设置图例区域半透明,并调整了图例的位置和大小。

例 018 复合线形图+球面点

图3-4中用球面表示复合线形图中的数据点,形成一种更加生动形象的图表风格。

[图表效果]

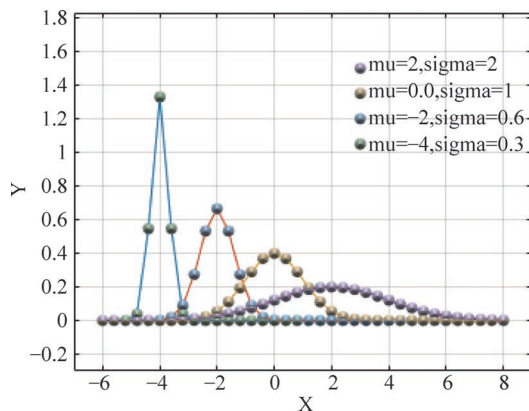


图 3-4 在复合线形图中用球面表示数据点

[代码实现]

```

1. % 复合线形图+球面点
2. clear; close all; % 清空工作空间的变量,关闭所有打开的对话框

```

```

3.   x = -6:.4:8;                                % 绘图数据
4.   y1 = normpdf(x, -4, 0.3);                    % 正态概率密度函数曲线, 注意参数不同
5.   y2 = normpdf(x, -2, 0.6);
6.   y3 = normpdf(x, 0, 1);
7.   y4 = normpdf(x, 2, 2);
8.
9.   plot(x, y1)                                  % 绘制第 1 条曲线
10.  hold on                                       % 叠加绘图
11.  plot(x, y2)                                  % 绘制第 2 条曲线
12.  plot(x, y3)                                  % 绘制第 3 条曲线
13.  plot(x, y4)                                  % 绘制第 4 条曲线
14.
15.  minY = min(min(y2), min(y1)) - 0.3;          % 计算两个方向上的取值范围
16.  maxY = max(max(y2), max(y1)) + 0.5;
17.  maxX = max(x) + 1;
18.  minX = min(x) - 1;
19.  xlim([minX maxX])
20.  ylim([minY maxY])
21.  scale = (maxY - minY) / (maxX - minX);        % 计算纵横比
22.  ax = gca;                                     % 设置坐标系为当前坐标系
23.  v = ax.PlotBoxAspectRatio;
24.  rt = v(2)/v(1);                               % 各方向长度比例
25.  for i = 1:length(x)                           % 用球面绘制第 1 条线上的数据点
26.      [X, Y, Z] = sphere;                       % 单位球面上网格节点的坐标
27.      X = 0.2 * X + x(i);                       % 缩放、平移
28.      Y = 0.2 * Y * scale/rt + y1(i);            % 根据纵横比和各方向显示比例调整
29.      surface(X, Y, Z, 'EdgeColor', 'none', 'FaceColor', uint8([194 220 191]))
30.  end
31.  for i = 1:length(x)                           % 用球面绘制第 2 条线上的数据点
32.      [X2, Y2, Z2] = sphere;
33.      X2 = 0.2 * X2 + x(i);
34.      Y2 = 0.2 * Y2 * scale/rt + y2(i);
35.      surface(X2, Y2, Z2, 'EdgeColor', 'none', 'FaceColor', uint8([186 216 251]))
36.  end
37.  for i = 1:length(x)                           % 用球面绘制第 3 条线上的数据点
38.      [X3, Y3, Z3] = sphere;
39.      X3 = 0.2 * X3 + x(i);
40.      Y3 = 0.2 * Y3 * scale/rt + y3(i);
41.      surface(X3, Y3, Z3, 'EdgeColor', 'none', 'FaceColor', uint8([251 229 184]))
42.  end
43.  for i = 1:length(x)                           % 用球面绘制第 4 条线上的数据点
44.      [X4, Y4, Z4] = sphere;
45.      X4 = 0.2 * X4 + x(i);
46.      Y4 = 0.2 * Y4 * scale/rt + y4(i);
47.      surface(X4, Y4, Z4, 'EdgeColor', 'none', 'FaceColor', uint8([231 216 250]))
48.  end
49.
50.  % 绘制图例, 不使用默认图例
51.  xx = [3, 3, 3, 3];
52.  yy = [1.05, 1.2, 1.35, 1.5];
53.  colors = [[194 220 191]; [186 216 251]; ...

```



```

54.     [251 229 184];[231 216 250]]./255;
55.     for i = 1:4                                % 绘制各点
56.         [X5,Y5,Z5] = sphere;
57.         X5 = 0.2 * X5 + xx(i);
58.         Y5 = 0.2 * Y5 * scale/rt + yy(i);
59.         surface(X5,Y5,Z5,'EdgeColor','none','FaceColor' colors(i,1:3))
60.     end
61.     text(3.5,yy(1),"mu = -4,sigma = 0.3","FontSize",7)    % 绘制各文本
62.     text(3.5,yy(2),"mu = -2,sigma = 0.6","FontSize",7)
63.     text(3.5,yy(3),"mu = 0.0,sigma = 1","FontSize",7)
64.     text(3.5,yy(4),"mu = 2,sigma = 2","FontSize",7)
65.
66.     ylabel('Y','FontSize',7)
67.     xlabel('X','FontSize',7)
68.     ax.FontSize = 6;                                % 修改坐标系刻度标签的字体大小
69.     view(2)                                          % 二维视图
70.     grid on                                          % 添加网格
71.     box on                                           % 显示外框
72.     camlight left                                   % 添加左侧光照
73.     camlight right                                 % 添加右侧光照
74.
75.     set(gcf,'PaperUnits','points','Position',[0 0 400 300])
76.     exportgraphics(gcf,'sam03_04.png','ContentType','image','Resolution',300) % 保存为 png 文件
77.     exportgraphics(gcf,'sam03_04.pdf','ContentType','vector')           % 保存为 pdf 文件

```

[代码说明]

代码第 3~7 行为绘图数据,使用 `normpdf` 函数生成服从指定参数正态分布的概率密度函数数据。第 9~13 行绘制 4 组数据的曲线,第 15~48 行绘制曲线上的球面点,第 30~64 行绘制图例。

[技术要点]

例 010 绘制了正弦曲线和余弦曲线的点图,为了保证球面始终是圆的,用了坐标轴取值范围的纵横比在曲面进行几何变换时进行调节,但是要求坐标系是方形的。

本例在纵横比的基础上增加一个调节参数,即代码第 23、24 行用坐标系对象的 `PlotBoxAspectRatio` 属性值计算坐标系绘图区两个方向上的显示长度比率。注意,前面讲的纵横比是真实长度的比率,这里是显示长度的比率。用 `axis square`、`axis equal` 等语句绘图时,绘图区长宽方向上的显示长度比率是不一样的。代码第 28 行、第 34 行、第 40 行和第 46 行进行几何变换时用纵横比和显示长度比率进行了调节。

例 019 复合线形图+球面点+背景色

图 3-5 所示为给复合线形图添加渐变色填充的背景。

[图表效果]

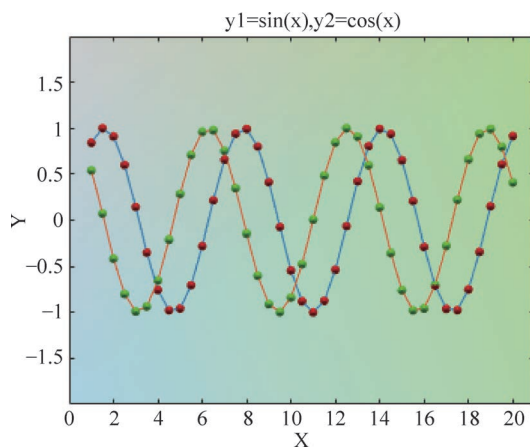


图 3-5 给复合线形图添加背景色

[代码实现]

```

1. % 复合线形图 + 球面点 + 背景色
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x = 1:0.5:20; % 绘图数据
4. y1 = sin(x);
5. y2 = cos(x);
6.
7. plot(x,y1) % 绘制第 1 条曲线
8. hold on % 叠加绘图
9. plot(x,y2) % 绘制第 2 条曲线
10.
11. minY = min(min(y2),min(y1)) - 1; % 计算两个方向上的取值范围
12. maxY = max(max(y2),max(y1)) + 1;
13. maxX = max(x) + 1;
14. minX = min(x) - 1;
15. xlim([minX maxX])
16. ylim([minY maxY])
17. scale = (maxY - minY)/(maxX - minX); % 计算纵横比
18. ax = gca; % 设置坐标系为当前坐标系
19. v = ax.PlotBoxAspectRatio;
20. rt = v(2)/v(1); % 各方向长度比例
21.
22. for i = 1:length(x) % 绘制第 1 条线上的点
23. [X,Y,Z] = sphere;
24. X = 0.2 * X + x(i);
25. Y = 0.2 * Y * scale/rt + y1(i);
26. surface(X,Y,Z,'EdgeColor','none','FaceColor','r')
27. end
28. for i = 1:length(x) % 绘制第 2 条线上的点
29. [X2,Y2,Z2] = sphere;
30. X2 = 0.2 * X2 + x(i);
31. Y2 = 0.2 * Y2 * scale/rt + y2(i);

```

```

32.     surface(X2,Y2,Z2,'EdgeColor','none','FaceColor','g')
33. end
34.
35. color = uint8([173 175 177 200;           % 绘图区 4 个顶点的颜色,R、G、B 和 Alpha
36.               216 217 218 200;
37.               236 154 132 200;
38.               228 228 85 200]);
39. fc = ax.Backdrop.Face;                    % 绘图区渐变色填充
40. fc.ColorBinding = 'interpolated';
41. fc.ColorData = color;
42.
43. title('y1 = sin(x),y2 = cos(x)','FontSize',8) % 标题
44. ylabel('Y','FontSize',7)
45. xlabel('X','FontSize',7)
46. ax.FontSize = 6;                         % 修改坐标系刻度标签的字体大小
47. view(2)                                  % 二维视图
48. box on                                    % 显示外框
49. camlight left                             % 添加左侧光照
50. camlight right                           % 添加右侧光照
51. hold off                                  % 取消叠加绘图
52.
53. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
54. exportgraphics(gcf,'sam03_05.png','ContentType','image','Resolution',300) % 保存为 png 文件
55. exportgraphics(gcf,'sam03_05.pdf','ContentType','vector')                % 保存为 pdf 文件

```

[代码说明]

代码第 3~5 行为绘图数据,第 7~9 行绘制复合线形图,第 11~33 行绘制曲线上的球面点,第 35~41 行绘制绘图区背景。

[技术要点]

例 017 介绍用 `rectangle` 函数直接绘制矩形构造绘图区背景,本例介绍另外一种更加直接的方法,这种方法适合对整个绘图区进行颜色填充。

代码第 35~41 行直接通过坐标系对象的属性设置实现绘图区的渐变色填充,代码如下:

```

color = uint8([173 175 177 200;           % 绘图区 4 个顶点的颜色,R、G、B 和 Alpha
               216 217 218 200;
               236 154 132 200;
               228 228 85 200]);
fc = ax.Backdrop.Face;                    % 绘图区渐变色填充
fc.ColorBinding = 'interpolated';
fc.ColorData = color;

```

`color` 变量定义绘图区 4 个顶点的颜色组成的颜色矩阵,注意需要指定透明度值,每行 4 个数分别表示颜色的红色、绿色、蓝色分量和透明度值,可以直接用 0~1 的小数定义,也可以用 0~255 的整数定义,然后用 `uint8` 等函数转换。注意按逆时针方向指定 4 个顶点的颜色。

通过连续引用获取 `ax.Backdrop.Face` 属性的值,它是一个表示绘图区区域的面对象。设置该对象的 `ColorBinding` 属性值为 `'interpolated'`、`ColorData` 属性的值为 `color` 变量,实现

渐变色插值。该对象还有另外一些属性,如设置图片纹理等,但测试后发现暂不可用。

例 020 平滑线形图

通过三次样条插值可以对折线数据进行平滑,用平滑数据画出来的图形都经过数据点,并且是平滑曲线,如图 3-6 所示。

[图表效果]

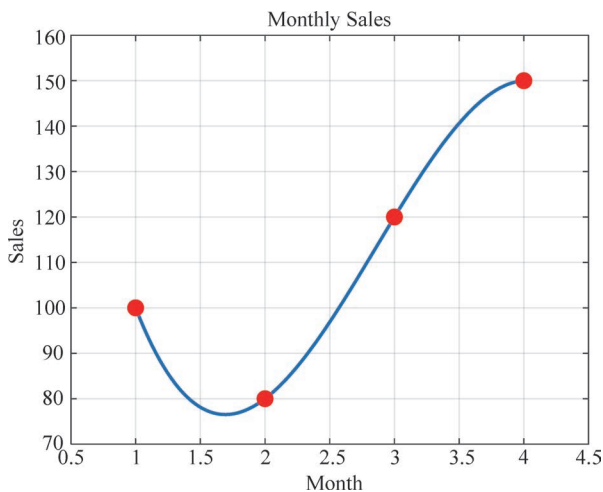


图 3-6 平滑线形图

[代码实现]

```

1. % 平滑线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x = 1:4; % 绘图数据
4. y = [100,80,120,150];
5.
6. pp = spline(x,y); % 根据原始数据得到合适的样条插值模型
7. xx = linspace(min(x),max(x),100); % x 向细分
8. yy = ppval(pp,xx); % 插值计算各细分点处的 y 值
9.
10. h = plot(xx,yy,'-'); % 绘制平滑曲线
11. h.LineWidth = 1.25; % 修改线宽
12. hold on % 叠加绘图
13. h = plot(x,y,'o'); % 绘制原始数据点
14. h.MarkerSize = 7; % 修改属性
15. h.MarkerFaceColor = '#FF0000';
16. h.MarkerEdgeColor = '#FF0000';
17. xlim([0.5 4.5]) % 设置 x 轴的取值范围
18. ylim([70 160]) % 设置 y 轴的取值范围
19.
20. title('Monthly Sales','FontSize',8) % 标题
21. ylabel('Sales','FontSize',7)

```

```
22. xlabel('Month','FontSize',7)
23. ax = gca; % 设置坐标系为当前坐标系
24. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
25. grid on % 添加网格
26. hold off % 取消叠加绘图
27.
28. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
29. exportgraphics(gcf,'sam03_06.png','ContentType','image','Resolution',300) % 保存为 png 文件
30. exportgraphics(gcf,'sam03_06.pdf','ContentType','vector') % 保存为 pdf 文件
```

[代码说明]

代码第 3、4 行为原始数据,第 5~8 行通过样条插值得到平滑后的数据 `xx` 和 `yy`,第 10~16 行用平滑数据绘制曲线,用原始数据绘制数据点。

[技术要点]

数据平滑的代码如下:

```
pp = spline(x,y);
xx = linspace(min(x),max(x),100);
yy = ppval(pp,xx);
```

显然,第 1 行代码用 `spline` 函数根据原始数据得到合适的样条插值模型,第 2 行代码用 `linspace` 函数在 `x` 的最小值和最大值之间等间隔取 100 个数,第 3 行代码用 `ppval` 函数结合前面得到的样条插值模型计算各 `xx` 值处的 `yy` 值,最终得到平滑后的曲线数据。

例 021 线形图+特殊字符标注

线形图常常用来表现数学公式的图形,数学公式中常常有一些特殊字符,本节介绍特殊字符的标注。图 3-7 中添加了两个数学公式,用箭头标注了 $x=3.14$ 处的 y 值 $\sin(\pi)$ 。

[图表效果]

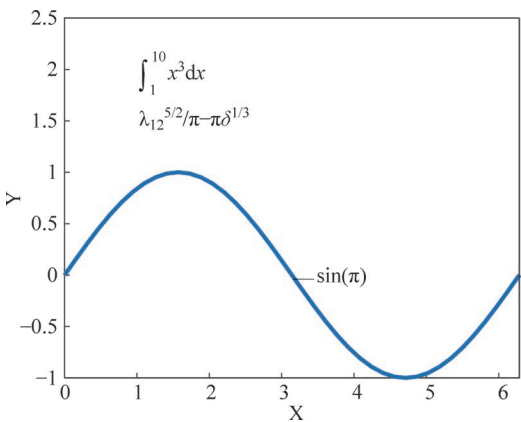


图 3-7 特殊字符标注

[代码实现]

```

1. % 线形图 + 特殊字符标注
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. ax = gca; % 设置坐标系为当前坐标系
4. ax.XLim = [0 2 * pi]; % 设置 x 轴的取值范围
5. ax.YLim = [-1 2.5]; % 设置 y 轴的取值范围
6. x = 0:pi/20:2 * pi; % 绘图数据
7. y = sin(x);
8. line(x,y, 'LineWidth',1.5) % 绘制线形图
9. text(pi,0, '\leftarrow sin(\pi)') % 用 Tex 标记添加文本标注
10. % 用 Latex 标记添加文本标注
11. text(1,2, '$ \int_1^{10} x^3 dx $ ', 'Interpreter', 'latex');
12. % 用 texlabel 函数添加文本标注
13. txt = texlabel('lambda 12^{(5/2)}/pi - pi * delta^{(1/3)}');
14. text(1,1.5,txt)
15. ylabel('Y', 'FontSize',7)
16. xlabel('X', 'FontSize',7)
17. ax.FontSize = 6;
18. box off
19. line([0,2 * pi],[2.5,2.5], 'Color', 'k', 'LineWidth',0.25)
20. line([2 * pi,2 * pi],[-1,2.5], 'Color', 'k', 'LineWidth',0.25)
21.
22. set(gcf, 'PaperUnits', 'points', 'Position',[0 0 400 300])
23. exportgraphics(gcf, 'sam03_07.png', 'ContentType', 'image', 'Resolution',300) % 保存为 png 文件
24. exportgraphics(gcf, 'sam03_07.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

[代码说明]

代码第 6、7 行为绘图数据,第 8 行绘制正弦曲线图,第 9~14 行添加标注。

[技术要点]

对于特殊字符, MATLAB 中可以使用 Tex 标记和 LaTeX 标记。用 text 函数添加文本标注时,将 Interpreter 参数的值设置为'tex'或'latex',可以使用 Tex 标记和 LaTeX 标记。默认时,即不声明时可直接使用 Tex 标记。关于 MATLAB 可用的 Tex 标记和 LaTeX 标记,请参见官方文档。

例 022 复合平滑曲线

将复合线形图进行平滑后得到复合平滑曲线,如图 3-8 所示。

[图表效果]

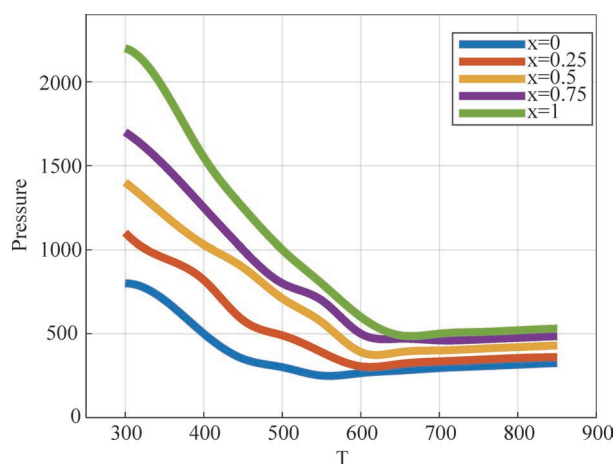


图 3-8 复合平滑曲线

[代码实现]

```

1. % 复合平滑曲线
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. x = 300:50:850; % 绘图数据,5 组
4. y = [800,700,500,350,300,250,265,280,295,305,315,325;
5.      1100,950,820,580,490,390,305,320,335,345,355,360;
6.      1400,1205,1030,895,710,570,390,390,400,410,420,430;
7.      1700,1500,1250,1000,800,700,500,470,460,465,475,485;
8.      2200,1950,1550,1250,1000,800,600,490,500,510,520,530];
9.
10. colororder gem % 颜色序列
11. for i = 1:5 % 平滑绘图数据并绘图
12.     pp = spline(x,y(i,:));
13.     xx = linspace(min(x),max(x),100); % x 轴细化
14.     yy = ppval(pp,xx); % 通过样条插值计算各细化 x 处的 y 值
15.     h = plot(xx,yy); % 绘制各线形图
16.     h.LineWidth = 3; % 设置线宽
17.     hold on % 叠加绘图
18. end
19. h.LineWidth = 3; % 第 5 根线的线宽
20.
21. % 绘制外框线,不使用默认的外框线
22. xlim([250,900])
23. ylim([0,2400])
24. line([250,900],[2400,2400],'Color','k','LineWidth',0.25)
25. line([900,900],[0,2400],'Color','k','LineWidth',0.25)
26.
27. ylabel('Pressure','FontSize',7)
28. xlabel('T','FontSize',7)
29. ax = gca; % 设置坐标系为当前坐标系
30. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小

```

```

31. labels = ["x = 0", "x = 0.25", "x = 0.5", "x = 0.75", "x = 1"];
32. legend(ax, labels)
33. grid on % 添加网格
34. box off % 不绘制外框
35. hold off % 取消叠加绘图
36.
37. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 400 300])
38. exportgraphics(gcf, 'sam03_08.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
39. exportgraphics(gcf, 'sam03_08.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

「代码说明」

代码第 3~8 行为绘图数据,第 10~19 行平滑数据并绘制曲线,第 21~25 行绘制外框。

〔技术要点〕

本例的技术要点在于数据平滑,请参见例 020 的介绍。

例 023 颜色填充复合线形图

将复合线形图中的曲线与其基线之间的区域进行颜色填充,得到颜色填充复合线形图,如图 3-9 所示。正弦和余弦曲线与 0 基线之间的区域进行了渐变色填充。

〔图表效果〕

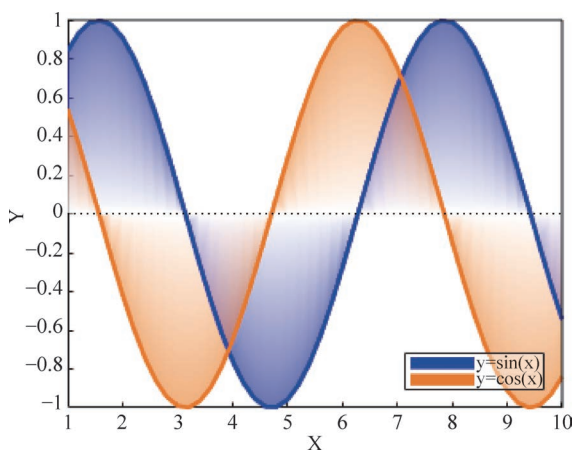


图 3-9 颜色填充复合线形图

「代码实现」

1.	% 颜色填充复合线形图	
2.	clear;close all;	% 清空工作空间的变量,关闭所有打开的对话框
3.	x1 = 1:0.2:10;	% 绘图数据,分两组
4.	y1 = sin(x1);	


```

5.  x2 = 1:0.2:10;
6.  y2 = cos(x2);
7.  h1 = drawPoly(x1,y1,[0 0 1]);           % 绘制第1个填充区域
8.  h2 = drawPoly(x2,y2,[1 0.5 0]);         % 绘制第2个填充区域
9.  hold on                                 % 叠加绘图
10. plot(x1,y1,'LineWidth',2,'Color','b')   % 绘制第1个线形
11. plot(x2,y2,'LineWidth',2,'Color',[1 0.5 0]) % 绘制第2个线形
12. line([1,10],[0,0],'Color','k','LineStyle',':') % 绘制0线
13.
14. % 自己绘制外框线
15. xlim([1,10])
16. ylim([-1,1])
17. line([1,10],[1,1],'Color','k','LineWidth',0.25)
18. line([10,10],[-1,1],'Color','k','LineWidth',0.25)
19.
20. ylabel('Y','FontSize',7)                % y轴标题
21. xlabel('X','FontSize',7)
22. ax = gca;                               % 设置坐标系为当前坐标系
23. ax.FontSize = 6;                        % 修改坐标系刻度标签的字体大小
24. hold off                                % 取消叠加绘图
25. lgd = legend(["y = sin(x)","y = cos(x)"],"Location','southeast');
26. lgd.BackgroundAlpha = 0.5;              % 图例区域半透明
27.
28. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
29. print(gcf,['sam03_09'],'.png','-r300','-dpng')
30. print(gcf,['sam03_09'],'.pdf','-bestfit','-dpdf')
31.
32. function p = drawPoly(x,y,cr)
33.     % 根据曲线和0线绘制填充区域
34.     len = length(x);
35.     maxY = max(y);
36.     for i = 1:len
37.         x(i+len) = x(i);
38.         y(i+len) = 0.0;
39.     end
40.     % 用面片实现曲线上各点及其在横轴上投影点构成的区域的渐变色填充
41.     for i = 1:len * 2
42.         vert(i,1) = x(i);
43.         vert(i,2) = y(i);
44.         va(i,1) = 0.7;
45.     end
46.     for i = 1:len
47.         vc(i,:) = cr;                    % 曲线上点的颜色为指定颜色
48.         vc(i+len,:) = [1 1 1];          % 投影点的颜色为白色
49.     end
50.     for i = 1:len - 1
51.         face(i,1) = i;
52.         face(i,2) = i + len;
53.         face(i,3) = i + len + 1;
54.         face(i,4) = i + 1;
55.     end

```

```

56.     p = patch('Vertices',vert,'Faces',face,'FaceVertexCData',vc,...
57.               'FaceColor','interp','EdgeColor','none','FaceVertexAlpha',va,'FaceAlpha','flat');
58.     slpha(0.5)
59. end

```

[代码说明]

代码第 3~6 行为绘图数据,第 7~12 行填充曲线与基线之间的区域并绘制曲线和基线,第 14~18 行绘制外框,第 20~26 行设置图表样式,第 28~30 行设置窗口并输出图表,第 32~59 行用一个函数实现曲线与基线之间区域的渐变色填充。

[技术要点]

本例的技术要点在于 drawPoly 函数的实现。该函数用面片绘制曲线和基线之间的面。MATLAB 常用面片表示不规则面。本书大量用到面片,所以有必要介绍面片的创建和设置。在 MATLAB 命令窗口输入以下代码:

```

>> vert = [0 0 .5;1 0 1;1 .5 1;.5 1 1;0 1 1];
>> fac = [1 3 2;1 4 3;1 5 4];
>> tcolor = [.7 .1 .4;1 1 0;1 0 0;0 1 0;0 0 1];
>> patch('Faces',fac,'Vertices',vert,'FaceVertexCData',tcolor,'FaceColor','interp');
>> axis([0 1 0 1 0 2]);
>> view(3)

```

生成图 3-10。图中面片由 3 个三角形面组合而成,这些三角形面称为面片的小面。

分析以上创建面片的代码,vert 变量是一个 5 行 3 列的矩阵,记录面片各顶点的三维坐标,每行数据表示 1 个点,如果是三维点,每行有 3 个值;如果是二维点,就只有两个值。各顶点从前往后从 1 开始依序编号。fac 变量也是一个矩阵,每行整数用顶点编号表示一个小面。例如第一行的 3 个数 1、3 和 2 表示第 1 个三角形小面的 3

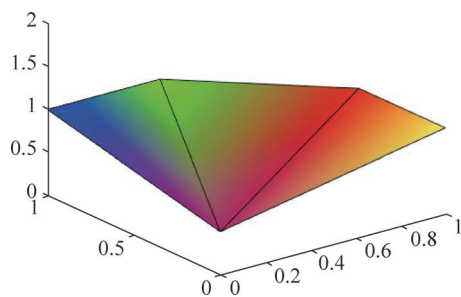


图 3-10 创建面片

个顶点的编号,分别为 1、3 和 2。注意点的编号按逆时针方向排列。tcolors 变量定义各顶点的颜色,用 5 行 3 列的矩阵表示,每行数据表示一个点的颜色。然后用 patch 函数绘制面片。面片的着色方式有 3 种,可以是单色、刻面着色和插值着色,这里使用插值着色,得到的是渐变色填充的效果。关于面片更详细的介绍,请参见本书姊妹篇《从理论到实践》。

明白 MATLAB 中面片的创建后,理解 drawPoly 函数就比较容易了,重点在于该函数怎么组织表示顶点、面和颜色的变量。

如图 3-11 所示,将正弦曲线与 0 基线之间的所有面看作一个面片,面片中有很多灰色的竖线。曲线、基线和相邻竖线围成的梯形可以看作一个小面,所以整个面片由很多四边形的小面组合而成。基线上灰色线的端点是曲线上数据点在基线上的投影点,该点处的颜色为白色。这样,就不难得到需要的表示顶点、小面和顶点颜色的变量。

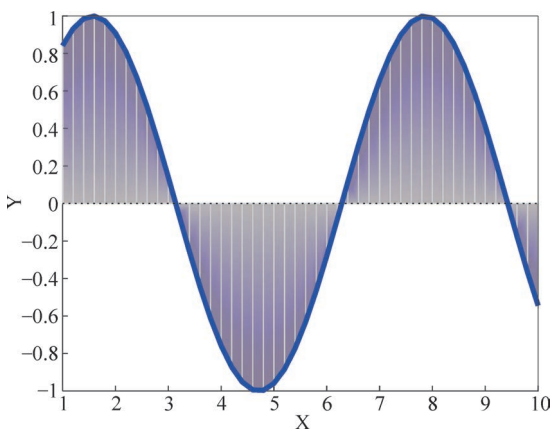


图 3-11 用面片绘制曲线与基线之间的区域

例 024 分面线形图

采集到的原始数据可以根据分类变量进行分类,各分类的数据可以用点标记、颜色、标记大小进行区分,也可以用分面图分别显示。所谓分面图,就是将各分类的数据分别在不同的坐标系中绘图,如图 3-12 所示。

[图表效果]

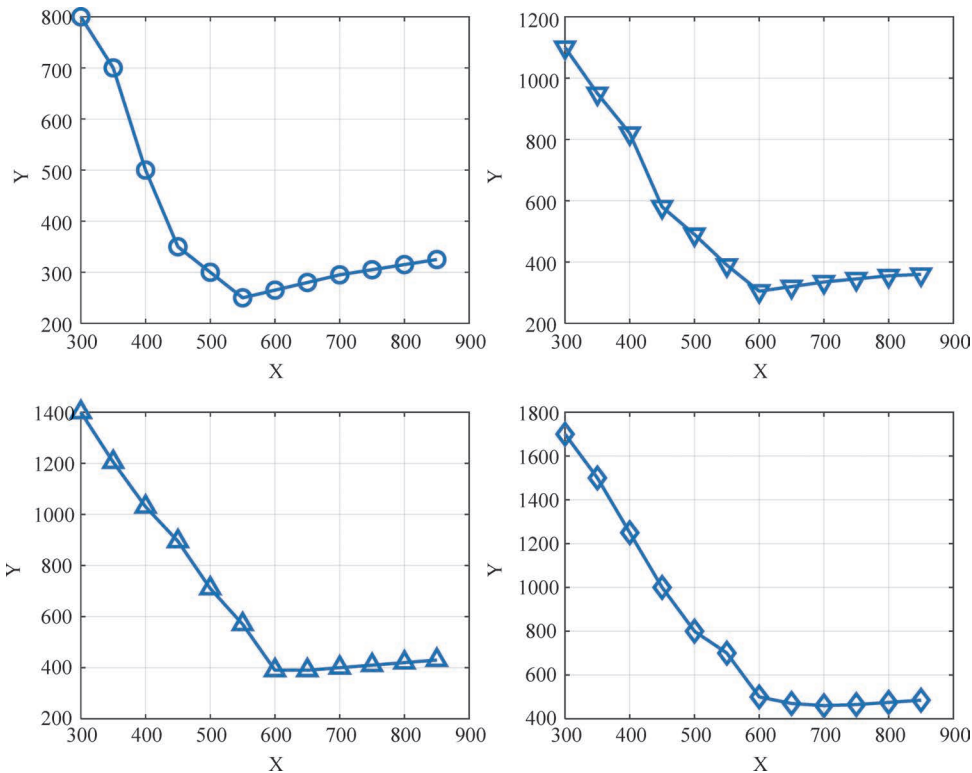


图 3-12 分面线形图

[代码实现]

```

1. % 分面线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. tiledlayout(2,2,"TileSpacing","compact"); % 多图
4. x = 300:50:850; % 绘图数据,5 组
5. y = [800,700,500,350,300,250,265,280,295,305,315,325;
6.      1100,950,820,580,490,390,305,320,335,345,355,360;
7.      1400,1205,1030,895,710,570,390,390,400,410,420,430;
8.      1700,1500,1250,1000,800,700,500,470,460,465,475,485;
9.      2200,1950,1550,1250,1000,800,600,490,500,510,520,530];
10.
11. ax1 = nexttile;
12. h = plot(x,y(1,:), '- o'); % 绘制第 1 组数据的线形图
13. h.LineWidth = 1.25; % 修改属性
14. h.MarkerSize = 6;
15. ylabel('Y','FontSize',7)
16. xlabel('X','FontSize',7)
17. ax1.FontSize = 6; % 刻度标签的字体大小
18. grid on % 添加网格
19.
20. ax2 = nexttile;
21. h = plot(x,y(2,:), '- v'); % 绘制第 2 组数据的线形图
22. h.LineWidth = 1.25; % 修改属性
23. h.MarkerSize = 6;
24. ylabel('Y','FontSize',7)
25. xlabel('X','FontSize',7)
26. ax2.FontSize = 6; % 刻度标签的字体大小
27. grid on % 添加网格
28.
29. ax3 = nexttile;
30. h = plot(x,y(3,:), '- ^'); % 绘制第 3 组数据的线形图
31. h.LineWidth = 1.25; % 修改属性
32. h.MarkerSize = 6;
33. ylabel('Y','FontSize',7)
34. xlabel('X','FontSize',7)
35. ax3.FontSize = 6; % 刻度标签的字体大小
36. grid on % 添加网格
37.
38. ax4 = nexttile;
39. h = plot(x,y(4,:), '- d'); % 绘制第 4 组数据的线形图
40. h.LineWidth = 1.25; % 修改属性
41. h.MarkerSize = 6;
42. ylabel('Y','FontSize',7)
43. xlabel('X','FontSize',7)
44. ax4.FontSize = 6; % 刻度标签的字体大小
45. grid on % 添加网格
46.
47. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 600 500])
48. exportgraphics(gcf, 'sam03_10.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
49. exportgraphics(gcf, 'sam03_10.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

[代码说明]

代码第4~9行为绘图数据,第11~45行在4个坐标系中分别绘制分面线形图。

[技术要点]

本例的技术要点是在同一个绘图窗口中绘制多个图表。MATLAB中实现多图绘制可以使用传统的 `subplot` 函数,或者使用本例用到的 `tiledlayout`...`nexttile` 结构。第3行代码用 `tiledlayout` 函数创建一个2行2列的网格布局,第11行、第20行、第29行和第38行代码用 `nexttile` 语句指定下一个坐标系进行绘图。

例 025 三维线形图

MATLAB可以在三维坐标系中绘制线形图,如图3-13所示。

[图表效果]

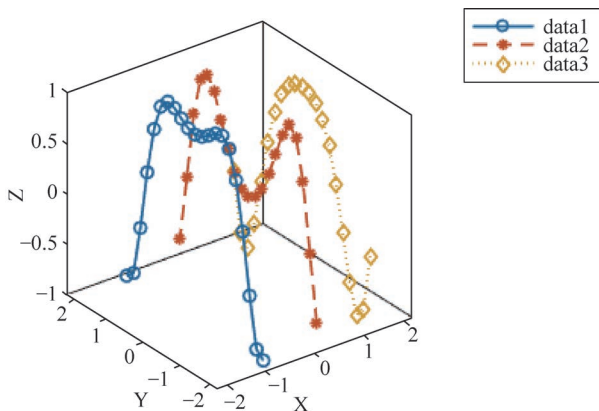


图 3-13 三维线形图

[代码实现]

```
1. % 三维线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. [X,Y] = meshgrid(-2:.2:2,-2:.2:2); % 数据
4. Z = sin(X.^2 + Y.^2);
5. h = plot3(X(:,6),Y(:,6),Z(:,6),'-o',...
6.         X(:,12),Y(:,12),Z(:,12),'-- * ',...
7.         X(:,18),Y(:,18),Z(:,18),' :d'); % 绘制三维线形图
8. for i = 1:length(h) % 设置线宽
9.     h(i).LineWidth = 1.25;
10. end
11. xlim([-2.2 2.2]) % 设置 x 轴的取值范围
12. ylim([-2.2 2.2]) % 设置 y 轴的取值范围
```

```
13. xlabel('X','FontSize',7)
14. ylabel('Y','FontSize',7)
15. zlabel('Z','FontSize',7)
16. ax = gca;                                % 设置坐标系为当前坐标系
17. ax.FontSize = 6;                          % 修改坐标系刻度标签的字体大小
18. legend                                    % 显示图例
19. box on                                    % 显示外框
20.
21. set(gcf, 'PaperUnits','points','Position',[0 0 400 300])
22. exportgraphics(gcf, 'sam03_11.png', 'ContentType','image', 'Resolution',300) % 保存为 png 文件
23. exportgraphics(gcf, 'sam03_11.pdf', 'ContentType','vector') % 保存为 pdf 文件
```

[代码说明]

代码第 3、4 行为绘图数据,第 5~10 行绘制三维线形图并修改属性。

[技术要点]

用 plot3 函数绘制三维线形图。与二维线形图不同的是,三维线形图中点的坐标使用三维坐标。

例 026 极坐标线形图

在极坐标系中绘制的线形图称为极坐标线形图,如图 3-14 所示。极坐标系中,用数据点相对于原点的方位角和二者之间的距离定义数据点在坐标系中的位置。

[图表效果]

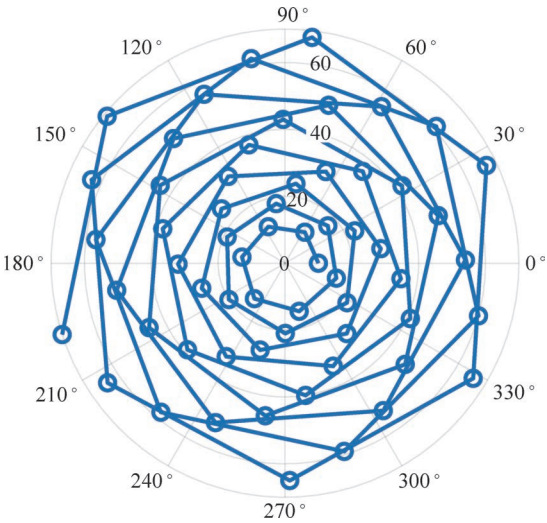


图 3-14 极坐标线形图

[代码实现]

```

1. % 极坐标线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. theta = 0:1:60; % 数据
4. rho = 10:1:70;
5. polarplot(theta,rho,'-o','LineWidth',1.5) % 绘制极坐标线形图
6. ax = gca; % 设置坐标系为当前坐标系
7. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
8.
9. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
10. exportgraphics(gcf,'sam03_12.png','ContentType','image','Resolution',300) % 保存为 png 文件
11. exportgraphics(gcf,'sam03_12.pdf','ContentType','vector') % 保存为 pdf 文件

```

[代码说明]

代码第 3、4 行为绘图数据,第 5~7 行绘制极坐标线形图并修改属性。

[技术要点]

用 `polarplot` 函数绘制极坐标线形图。绘图数据需要定义每个点的方位角和半径长度。

例 027 线形图区间填充

图 3-15 和图 3-16 中对两根折线之间的区域进行渐变色填充。前者利用 y 轴数据进行渐变色填充,后者利用 x 轴数据进行渐变色填充。

[图表效果]

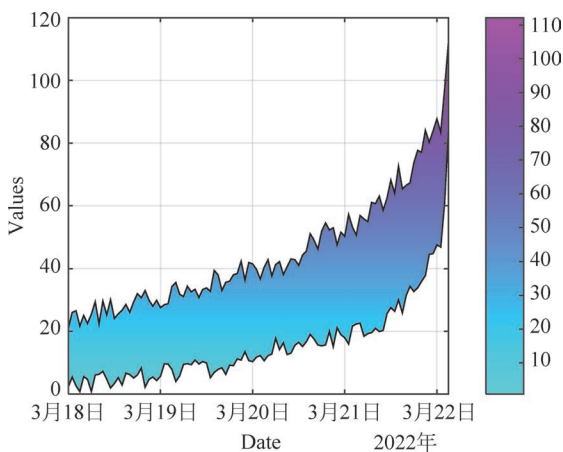


图 3-15 用 y 轴数据渲染

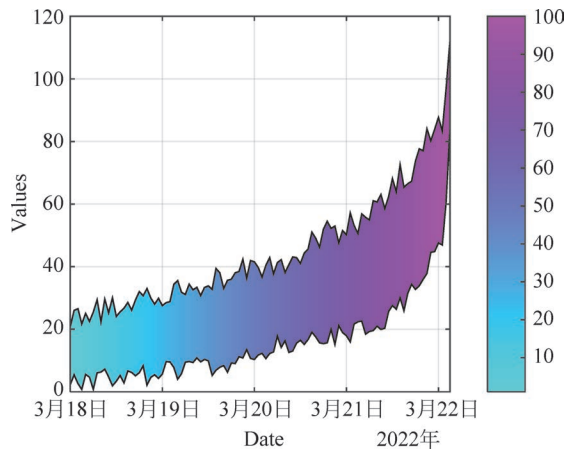


图 3-16 用 x 轴数据渲染

[代码实现]

```

1. % 线形图区间填充
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. tbl = readtimetable('时间序列 2.xlsx'); % 数据
4. xconf = [tbl.time;tbl.time(end:-1:1)]; % 多边形顶点 x 坐标,逆时针方向
5. yconf = [tbl.data1;tbl.data2(end:-1:1)]; % 多边形顶点 y 坐标
6. f = fill(xconf,yconf,'r'); % 面填充
7. f.CData = f.YData; % 根据 y 轴数据着色
8. f.FaceColor = 'interp'; % 渐变着色
9. xlim([min(tbl.time) max(tbl.time)]) % 设置 x 轴的取值范围
10. xlabel('Date','FontSize',7)
11. ylabel('Values','FontSize',7)
12. ax = gca; % 设置坐标系为当前坐标系
13. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
14. box on % 显示外框
15. grid on % 添加网格
16. colormap cool % 修改颜色查找表
17. colorbar % 显示色条
18.
19. set(gcf,'PaperUnits','points','Position',[0 0 300 220])
20. exportgraphics(gcf,'sam03_13.png','ContentType','image','Resolution',300) % 保存为 png 文件
21. exportgraphics(gcf,'sam03_13.pdf','ContentType','vector') % 保存为 pdf 文件
22.
23. figure
24. f2 = fill(xconf,yconf,'r');
25. f2.CData = [1:length(tbl.time) length(tbl.time):-1:1]'; % 用 x 轴数据着色
26. f2.FaceColor = 'interp';
27. xlim([min(tbl.time) max(tbl.time)])
28. xlabel('Date','FontSize',7)
29. ylabel('Values','FontSize',7)
30. ax = gca; % 设置坐标系为当前坐标系
31. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
32. box on % 显示外框
33. grid on % 添加网格
34. colormap cool % 修改颜色查找表
35. colorbar % 显示色条
36.

```



```
37. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 300 220])
38. exportgraphics(gcf, 'sam03_12_2.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
39. exportgraphics(gcf, 'sam03_12_2.pdf', 'ContentType', 'vector') % 保存为 pdf 文件
```

[代码说明]

代码第 3~21 行绘制图 3-15,第 22~38 行绘制图 3-16。

[技术要点]

例 023 介绍了用面片实现曲线和基线之间区域的渐变色填充的方法。本例介绍另外一种方法,即使用 fill 函数进行渐变色填充。该方法使用起来更简便。

使用 fill 函数进行多边形的颜色填充,先要定义一个多边形。定义多边形,用两个变量分别保存多边形各顶点的 x 坐标和 y 坐标,注意各顶点按逆时针方向排列。定义好后,将 x 坐标和 y 坐标作为参数指定给 fill 函数即可。该函数返回一个 fill 对象 f ,指定该对象 CData 属性的值为 f . YData 时,用 y 轴数据映射当前颜色查找表进行渐变色填充;指定为 x 轴数据组成的数组时,用 x 轴数据实现渐变色填充。如本例代码第 8 行和第 25 行所示。

例 028 纵向线形图

纵向线形图中, y 轴是分类轴, x 轴是数值轴,如图 3-17 所示。纵向线形图实际上是将同一序列的滑珠图用直线段连接相邻点。

[图表效果]

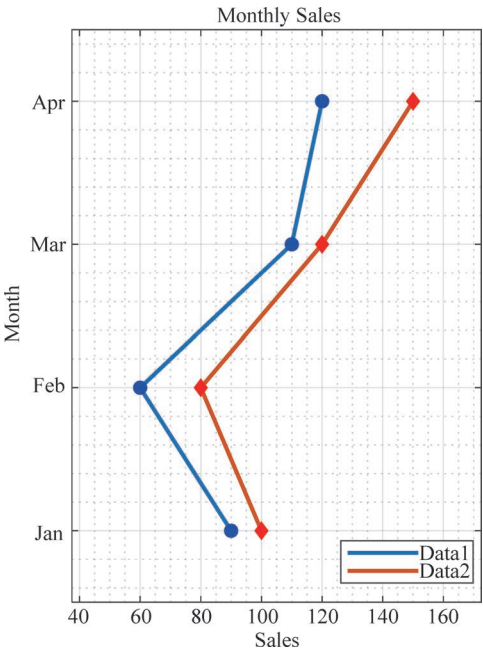


图 3-17 纵向线形图

[代码实现]

```

1.  % 纵向线形图
2.  clear;close all;                                % 清空工作空间的变量,关闭所有打开的对话框
3.  y = [1,2,3,4]';                                % 数据
4.  x1 = [90,60,110,120]';
5.  x2 = [100,80,120,150]';
6.  plot([x1 x2],y)                                % 绘制线形图
7.  hold on                                         % 叠加绘图
8.  h2 = scatter(x1,y,'b','filled');                % 绘制蓝色数据点
9.  h2.MarkerFaceAlpha = 1;                         % 修改属性
10. h3 = scatter(x2,y,'r','filled');                % 绘制红色数据点
11. h3.Marker = "diamond";
12. h3.MarkerFaceAlpha = 1;
13. xMin = min(min(x1),min(x2));                    % 计算并设置 x 轴的取值范围
14. xMax = max(max(x1),max(x2));
15. xRag = xMax - xMin;
16. xlim([xMin - xRag/4,xMax + xRag/4])
17. title('Monthly Sales','FontSize',8)            % 标题
18. xlabel('Sales','FontSize',7)
19. ylabel('Month','FontSize',7)
20. ylim([0.5 4.5])                                % 设置 y 轴的取值范围
21. yticks(1:4)                                     % 设置 y 轴的刻度位置
22. labels = ["Jan","Feb","Mar","Apr"];
23. yticklabels(labels)                             % 设置 y 轴的刻度标签
24. ax = gca;                                        % 设置坐标系为当前坐标系
25. ax.FontSize = 6;                                % 修改坐标系刻度标签的字体大小
26. grid on                                         % 添加网格
27. ax.XMinorGrid = "on";
28. ax.YMinorGrid = "on";
29. legend(["Data1","Data2"],'Location','southeast')
30. hold off                                         % 取消叠加绘图
31.
32. set(gcf,'PaperUnits','points','Position',[0 0 400 300])
33. exportgraphics(gcf,'sam03_14.png','ContentType','image','Resolution',300) % 保存为 png 文件
34. exportgraphics(gcf,'sam03_14.pdf','ContentType','vector')           % 保存为 pdf 文件

```

[代码说明]

代码第 3~5 行为绘图数据,第 6~11 行绘制纵向线形图并修改属性。

例 029 时间序列数据线形图

MATLAB 中,时间序列数据用时间表表示,它其实是以索引列为日期时间类型数据的表。时间序列数据常用线形图和面积图表示。与其他类型数据绘制成的线形图不同的是,时间序列数据线形图的横轴为时间轴,刻度标签是日期时间数据,如图 3-18 所示。

[图表效果]

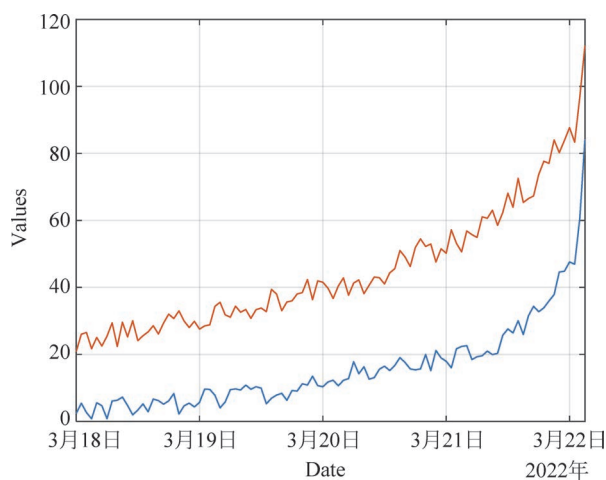


图 3-18 时间序列数据线形图

[代码实现]

```

1. % 时间序列数据线形图
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. tbl = readtimetable('时间序列 2.xlsx'); % 读取数据
4. plot(tbl.time,tbl.data1) % 绘制曲线 1
5. hold on % 叠加绘图
6. plot(tbl.time,tbl.data2) % 绘制曲线 2
7. xlim([min(tbl.time) max(tbl.time)]) % 设置 x 轴的取值范围
8. xlabel('Date','FontSize',7)
9. ylabel('Values','FontSize',7)
10. ax = gca; % 设置坐标系为当前坐标系
11. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
12. box on % 显示外框
13. grid on % 添加网格
14. hold off % 取消叠加绘图
15.
16. % set(gcf,'PaperUnits','points','Position',[0 0 300 220])
17. exportgraphics(gcf,'sam03_15.png','ContentType','image','Resolution',300) % 保存为 png 文件
18. exportgraphics(gcf,'sam03_15.pdf','ContentType','vector') % 保存为 pdf 文件

```

[代码说明]

代码第 3 行导入时间序列数据,第 4~6 行绘制两条表示时间序列数据的折线。

[技术要点]

用 plot 函数绘制时间序列数据线形图。

例 030 局部放大

为了查看图表局部的细节,常在图表中叠加局部放大的图形,如图 3-19 所示。放大的图形显示在另外一个坐标系中,是图中文的效果。

[图表效果]

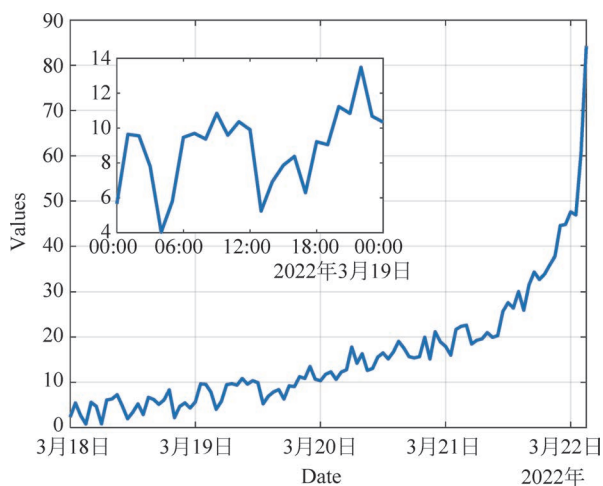


图 3-19 曲线局部放大

[代码实现]

```

1. % 局部放大
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3.
4. tbl = readtimetable('时间序列 2.xlsx'); % 读取数据
5. plot(tbl.time,tbl.data1,'LineWidth',1.25) % 绘制曲线
6. xlim([min(tbl.time) max(tbl.time)]) % 设置 x 轴的取值范围
7. xlabel('Date','FontSize',7)
8. ylabel('Values','FontSize',7)
9. ax = gca; % 设置坐标系为当前坐标系
10. ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
11. box on % 显示外框
12. grid on % 添加网格
13.
14. ax2 = axes(gcf); % 添加第 2 个坐标系
15. ax2.Position = [0.2,0.5,0.4,0.35]; % 第 2 个坐标系的位置
16. d1 = tbl.data1; % 绘图数据
17. t1 = tbl.time;
18. x2 = t1(25:49);
19. y2 = d1(25:49);
20. plot(ax2,x2,y2,'LineWidth',1.25) % 绘制第 2 条曲线
21. box(ax2,"on") % 添加外框

```

```

22. ax2.FontSize = 6; % 刻度标签的字体大小
23.
24. set(gcf, 'PaperUnits', 'points', 'Position', [0 0 400 300])
25. exportgraphics(gcf, 'sam03_16.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
26. exportgraphics(gcf, 'sam03_16.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

[代码说明]

代码第 4~12 行绘制原始线形图,第 14~22 行在另外一个坐标系中绘制放大的线形图。

[技术要点]

所谓的局部放大,实际上是把要放大显示的局部数据单独拿出来,在另外一个坐标系中画出来。

例 031 三维线形图+面板

在三维线形图的基础上叠加绘制半透明的矩形区域,以增强表现力,如图 3-20 所示。

[图表效果]

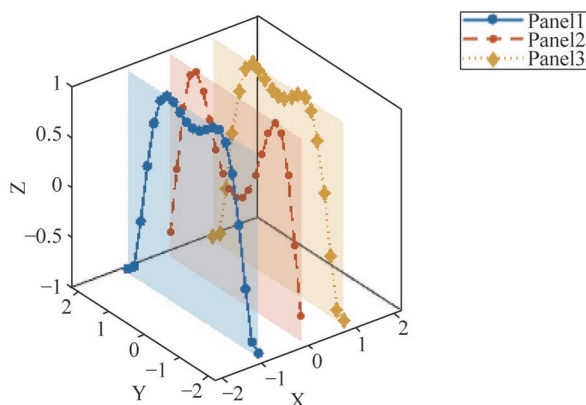


图 3-20 三维线形图叠加半透明面板

[代码实现]

```

1. % 三维线形图+面板
2. clear;close all; % 清空工作空间的变量,关闭所有打开的对话框
3. [X,Y] = meshgrid(-2:.2:2,-2:0.2:2); % 数据
4. Z = sin(X.^2 + Y.^2);
5. h = plot3(X(:,6),Y(:,6),Z(:,6),'-o',...
6. X(:,11),Y(:,11),Z(:,11),'--*',...
7. X(:,16),Y(:,16),Z(:,16),'d'); % 绘制三维曲线图
8. minZ = min(min(Z)); % 最小 Z 值

```

```

9.     maxZ = max(max(Z)); % 最大 Z 值
10.    for i = 1:length(h) % 修改每条线的线宽和标记大小
11.        h(i).LineWidth = 1.25;
12.        h(i).MarkerSize = 4;
13.    end
14.    h(1).MarkerFaceColor = h(1).Color;
15.    h(3).MarkerFaceColor = h(3).Color;
16.
17.    hold on % 叠加绘图
18.    for i = -1:1 % 用面片绘制面板
19.        vert = [i - 2 minZ; i 2 minZ; i 2 maxZ; i - 2 maxZ]; % 面片顶点
20.        face = [1 2 3 4]; % 定义面
21.        % vc = [0 0 1; 0 0 1; 0 0 1; 0 0 1];
22.        vc = [h(i+2).Color; h(i+2).Color; h(i+2).Color; h(i+2).Color]; % 顶点颜色
23.        va = [0.95; 0.95; 0.95; 0.95]; % 顶点透明度
24.        p = patch('Faces', face, 'Vertices', vert, ...
25.            'FaceVertexCData', vc, 'FaceColor', 'interp', ...
26.            'FaceVertexAlphaData', va, 'FaceAlpha', 'flat'); % 绘制面片
27.        p.EdgeColor = 'none'; % 不显示面片的边线
28.        alpha 0.15 % 面的透明度
29.    end
30.
31.    xlim([-2.2 2.2]) % 设置 x 轴的取值范围
32.    ylim([-2.2 2.2]) % 设置 y 轴的取值范围
33.    xlabel('X', 'FontSize', 7)
34.    ylabel('Y', 'FontSize', 7)
35.    zlabel('Z', 'FontSize', 7)
36.    ax = gca; % 设置坐标系为当前坐标系
37.    ax.FontSize = 6; % 修改坐标系刻度标签的字体大小
38.    legend(["Panel 1", "Panel 2", "Panel 3"])
39.    box on % 显示外框
40.    view(3) % 三维视图
41.    hold off % 取消叠加绘图
42.
43.    set(gcf, 'PaperUnits', 'points', 'Position', [0 0 400 300])
44.    exportgraphics(gcf, 'sam03_17.png', 'ContentType', 'image', 'Resolution', 300) % 保存为 png 文件
45.    exportgraphics(gcf, 'sam03_17.pdf', 'ContentType', 'vector') % 保存为 pdf 文件

```

[代码说明]

代码第 3~15 行绘制三维线形图,第 17~29 行绘制在曲线上叠加的半透明矩形区域。

[技术要点]

MATLAB 中使用 `rectangle` 函数只能绘制二维矩形区域,这里使用 `patch` 函数用面板绘制矩形区域。关于面板的创建,请参见例 023。这里每个矩形区域用对应曲线的颜色进行绘制。