

实验序号: 实验 1

实验项目: MATLAB 基本操作

实验成绩:

教师签名:



# 西南大学人工智能学院 实验报告

学年学期	2023 年秋季学期
课程名称	数值分析
姓 名	洪浩钦
学 号	222021335210144
学 院	含弘学院
专 业	智能科学与技术
班 级	袁隆平班
任课教师	钟秀蓉

2023 年 11 月 8 日

## 一、实验目的

**1.知识目标:** 通过学习,使学生全面掌握 MATLAB 语言的基本语法,包括变量定义、操作符、控制流程和函数的使用等。学生将能够理解和编写结构良好的 MATLAB 代码,并掌握常用的编程技巧和最佳实践。

**2.能力目标:** 培养学生具备熟练的 MATLAB 编程能力和初步的数值分析与计算能力。学生将能够利用 MATLAB 进行数据处理和分析,包括数据导入、处理、可视化和统计分析。他们将熟悉 MATLAB 中的数值计算方法,并能够使用 MATLAB 解决实际问题,如数值积分、方程求解和优化等。

**3.素质目标:** 通过掌握现代算法设计工具,提高学生的职业素养水平。学生将了解和应用现代算法设计的基本原理,并能够选择和使用适当的算法解决实际问题。此外,他们将培养良好的编程风格和团队合作能力,能够有效地与其他团队成员合作开发和优化 MATLAB 代码。这将有助于提升学生的创新能力、问题解决能力和职业竞争力。

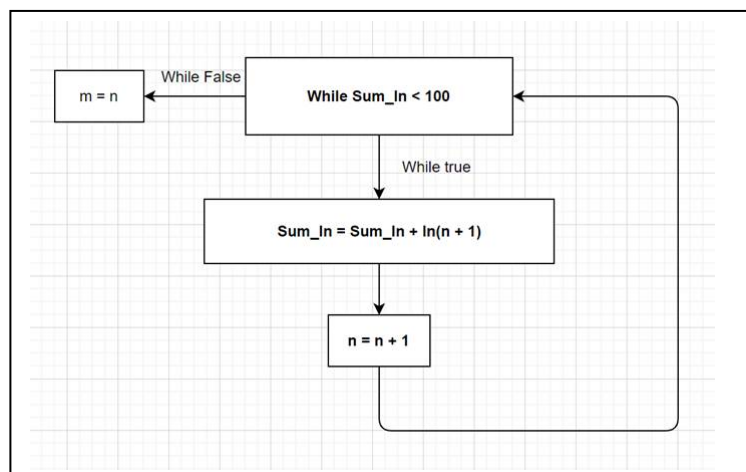
## 二、算法原理概述

MATLAB 是一个功能强大的数值分析工具,它提供了各种内置函数和工具箱,可以用于执行各种数值分析任务。

实验 3 求满足  $\sum_{n=0}^m \ln(1+n) > 100$  的最小整数  $m$

**解决思路:** 使用循环和 log 函数来求解问题。

首先初始化累加器  $\text{sum\_ln}$  和计数器  $m$ ,然后在循环中不断累加  $\log(1+m)$ ,直到  $\text{sum\_ln}$  大于 100。最后,输出结果  $m$ 。



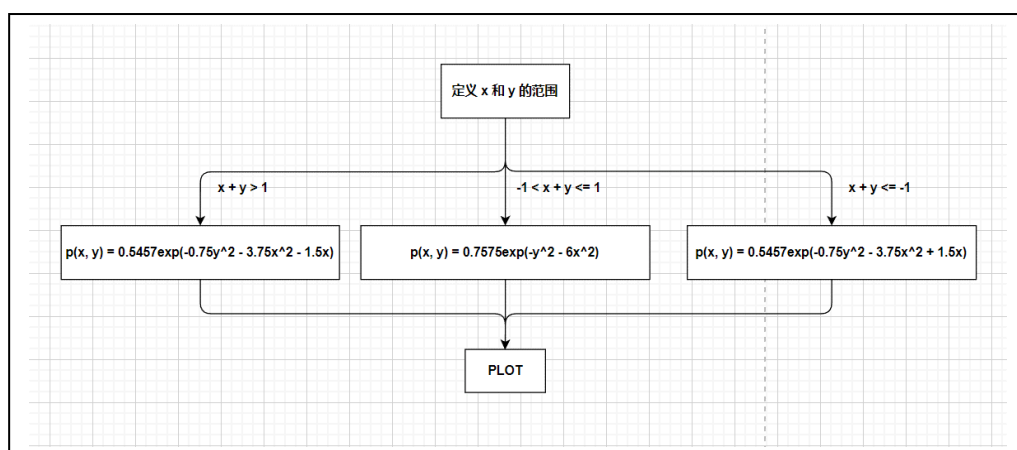
实验 8 用 MATLAB 函数表示下列函数, 并作图

$$p(x, y) = 0.5457 \exp(-0.75y^2 - 3.75x^2 - 1.5x), \quad x + y > 1$$

$$p(x, y) = 0.7575 \exp(-y^2 - 6x^2), \quad -1 < x + y \leq 1$$

$$p(x, y) = 0.5457 \exp(-0.75y^2 - 3.75x^2 + 1.5x), \quad x + y \leq -1$$

**解题思路:** 根据给定的函数表达式和条件, 计算每个 $(x, y)$ 点对应的函数值。这里使用了逻辑运算符来实现条件判断。



### 三、软件开发环境及工具

MATLAB R2021a

### 四、实验内容

- 1、问题提出
- 2、解决思路
- 3、算法步骤
- 4、结果分析
- 5、实验核心代码

#### 实验 1 执行命令并观察运算结果

##### 1. 结果分析:

(1)  $[1 \ 2; 3 \ 4] + 10 - 2i$

输出结果:

ans =

11.0000 - 2.0000i 12.0000 - 2.0000i

13.0000 - 2.0000i 14.0000 - 2.0000i

直接对矩阵进行加法运算。

(2)  $[1 \ 2; 3 \ 4].*[0.1 \ 0.2; 0.3 \ 0.4]$

输出结果：

ans =

0.1000 0.4000

0.9000 1.6000

对两个矩阵进行逐元素相乘。

(3)  $[1\ 2; 3\ 4] \setminus [20\ 10; 9\ 2]$

输出结果：

ans =

20.0000 5.0000

3.0000 0.5000

.操作符表示元素级运算，\操作符表示左除运算。所以， $A \setminus B$  表示对应元素的左除运算，即 B 矩阵中的每个元素除以 A 矩阵中的对应元素。

(4)  $[1\ 2; 3\ 4]^2$

输出结果：

ans =

1 4

9 16

$A.^2$  表示对矩阵 A 中的每个元素进行平方运算。

(5)  $\exp([1\ 2; 3\ 4])$

输出结果：

ans =

2.7183 7.3891

20.0855 54.5982

exp 函数直接对矩阵中元素进行指数运算。

(6)  $\log([1\ 10\ 100])$

输出结果：

ans =

0 2.3026 4.6052

log 函数用于计算自然对数（以 e 为底）。

(7) `prod([1 2; 3 4])`

输出结果:

`ans =`

3      8

`prod` 函数用于计算矩阵每一列元素的乘积。

(8) `[a, b] = min([10 20 ; 30 4])`

输出结果:

`a =`

10      4

`b =`

1    2

`min` 会返回一个向量 `a`，其中的元素是输入矩阵每一列的最小值，和一个向量 `b`，其中的元素是最小值在对应列中的行索引。

(9) `abs([1 2 ; 3 4] - pi)`

输出结果:

`ans =`

2.1416    1.1416

0.1416    0.8584

`abs([1 2 ; 3 4] - pi)` 会返回一个矩阵，其中的每个元素是输入矩阵中对应元素减去  $\pi$  的绝对值。

(10) `[10 20 ; 30 40] >= [40 30 ; 20 10]`

输出结果:

`ans =`

2×2 logical 数组

0    0

1    1

返回一个逻辑矩阵，其中的每个元素表示左边矩阵中的对应元素是否大于等于右边矩阵中的对应元素。

(11) `find([10 20 ; 30 40] >= [40 30 ; 20 10])`

输出结果:

ans =

2

4

返回一个向量，其中的元素是左边矩阵中大于等于右边矩阵中对应元素的元素的线性索引。

(12) [a, b] = find([10 20 ; 30 40] >= [40 30 ; 20 10])

输出结果：

a =

2

2

b =

1

2

a 中的元素是左边矩阵中大于等于右边矩阵中对应元素的元素的行索引，b 中的元素是对应的列索引。

(13) all([1 2 ; 3 4] > 1)

输出结果：

ans =

1×2 logical 数组

0 1

all 函数用于检查数组中的所有元素是否都为真（即非零）。当输入是一个逻辑数组时，all 函数会返回一个逻辑值，表示数组中的所有元素是否都为真。

(14) any([1 2 ; 3 4] > 1)

输出结果：

ans =

1×2 logical 数组

1 1

any 函数用于检查数组中是否有元素为真（即非零）。当输入是一个逻辑数组时，any 函数会返回一个逻辑值，表示数组中是否有元素为真。

(15) linspace(3, 4, 5)

输出结果：

```
ans =  
    3.0000    3.2500    3.5000    3.7500    4.0000
```

linspace 函数用于生成一个线性间隔的向量。linspace(a, b, n)生成一个长度为 n 的向量，其元素从 a 线性增长到 b。

(16) A = [1 2; 3 4]; A(:, 2)

输出结果：

```
ans =  
     2  
     4
```

:操作符用于索引数组。A(:, 2)表示取矩阵 A 的第二列。

2. 实验代码

```
clc;  
  
[1 2; 3 4] + 10 -2i                                % 直接对矩阵进行加法运算  
  
[1 2; 3 4].*[0.1 0.2 ; 0.3 0.4]                   % 对两个矩阵进行逐元素相乘  
  
[1 2; 3 4].\[20 10 ; 9 2]                           % A \ B = B(i, j) / A(i, j)  
  
[1 2; 3 4].^2  
  
exp([1 2; 3 4])  
  
log([1 10 100])  
  
prod([1 2; 3 4])                                     % 矩阵每一列元素的乘积  
  
[a, b] = min([10 20 ; 30 4])  
  
abs([1 2 ; 3 4] - pi)  
  
[10 20 ; 30 40] >= [40 30 ; 20 10]  
  
find([10 20 ; 30 40] >= [40 30 ; 20 10])  
  
[a, b] = find([10 20 ; 30 40] >= [40 30 ; 20 10])    % a为行号, b为列号  
  
all([1 2 ; 3 4] > 1)  
  
any([1 2 ; 3 4] > 1)  
  
linspace(3, 4, 5)                                     % linspace(a, b, n) 会生成一个包含n个元素的向量,  
其中第一个元素是a, 最后一个元素是b  
  
A = [1 2; 3 4]; A(:, 2)
```

装  
订  
线

实验 2 设  $x$  为一个维数为  $n$  的数组，编程求下列均值和标准差

$$\bar{x} = 1/n * (\sum_{i=1}^n x_i) \quad s = \sqrt{1/(n-1) \sum_{i=1}^n (x_i^2 - n * (\bar{x})^2)}$$

并就  $x = (81, 70, 65, 51, 76, 66, 90, 87, 61, 77)$  计算

### 1. 解决思路:

可以使用内置的 `mean` 函数来计算均值，使用 `std` 函数来计算标准差。

### 2. 实验结果:

>> exp2

Mean: 72.4

Standard deviation: 12.1124

编程计算均值为 72.4，标准差为 12.1124。

### 3. 实验代码:

```
% 设 x 为一个维数为 n 的数组，编程求下列均值和标准差
% \bar{x} = 1/n * (\sum_{i=1}^n x_i)   s = \sqrt{1/(n-1) \sum_{i=1}^n (x_i^2 - n * (\bar{x})^2)}
% 就 x = (81, 70, 65, 51, 76, 66, 90, 87, 61, 77) 计算

x = [81, 70, 65, 51, 76, 66, 90, 87, 61, 77];
n = length(x);

x_bar = sum(x) / n;
s = sqrt(1/(n-1) * (sum(x.^2) - n * x_bar^2));

disp(['Mean: ', num2str(x_bar)]);
disp(['Standard deviation: ', num2str(s)]);
```

实验 3 求满足  $\sum_{n=0}^m \ln(1+n) > 100$  的最小整数  $m$

### 1. 解决思路:

可以使用循环和 `log` 函数来求解这个问题。

首先初始化累加器 `sum_ln` 和计数器 `m`，然后在循环中不断累加 `log(1+m)`，直到 `sum_ln` 大于 100。最后，输出结果 `m`。

### 2. 实验结果:



>> exp3

The smallest integer m that satisfies the condition is: 37

编程计算得到满足条件的最小 m 为 37。

3. 实验代码:

```
% 求满足 \sum_{n=0}^m \ln(1 + n) > 100 的最小整数 m

sum_ln = 0;
m = 0;

% 迭代求解
while sum_ln <= 100
    sum_ln = sum_ln + log(1 + m);
    m = m + 1;
end

m = m - 1;
disp(['The smallest integer m that satisfies the condition is: ', num2str(m)]);
```

实验 4 用循环语句形成 Fibonacci 数列  $F_1 = F_2 = 1, F_k = F_{k-1} + F_{k-2} (k \geq 3)$

验证极限  $F_k / F_{k-1} \rightarrow (1 + \sqrt{5}) / 2$  (计算至两边误差小于精度  $1e-8$ )

1. 解决思路:

初始化: 创建一个数组 F 来存储 Fibonacci 数列, 初始值为[1, 1]。定义黄金比例 golden\_ratio 为  $(1 + \sqrt{5}) / 2$ , 初始化比值 ratio 为 0。

生成 Fibonacci 数列并计算比值: 使用 while 循环, 循环条件是比值 ratio 与黄金比例 golden\_ratio 的差的绝对值大于  $1e-8$ 。在每次循环中, 根据 Fibonacci 数列的定义, 计算下一个数, 并更新比值 ratio。

2. 实验结果:

>> exp4

Fibonacci sequence: 1        1        2        3        5        8        13        21        34        55  
89    144    233    377    610    987    1597    2584    4181    6765    10946  
Final ratio: 1.618

编程验证极限  $F_k / F_{k-1} \rightarrow (1 + \sqrt{5}) / 2$  成立。

3. 实验代码:

```
% 用循环语句形成 Fibonacci 数列  $F_1 = F_2 = 1, F_k = F_{k-1} + F_{k-2} (k \geq 3)$ 
```

装  
订  
线

```
% 验证极限  $F_k / F_{k-1} \rightarrow (1 + \sqrt{5}) / 2$  (计算至两边误差小于精度1e-8)
```

```
F = [1, 1];  
golden_ratio = (1 + sqrt(5)) / 2;  
ratio = 0;
```

```
% 生成Fibonacci数列, 并计算比值, 直到误差小于1e-8
```

```
k = 3;  
while abs(ratio - golden_ratio) > 1e-8  
    F(k) = F(k-1) + F(k-2);  
    ratio = F(k) / F(k-1);  
    k = k + 1;  
end
```

```
% 输出结果
```

```
disp(['Fibonacci sequence: ', num2str(F)]);  
disp(['Final ratio: ', num2str(ratio)]);
```

实验5 分别用 for 和 while 循环结构编写程序,

求出  $K = \sum_{i=1}^{10^6} \frac{\sqrt{3}}{i^2}$  的值, 并考虑一种避免循环语句的程序设计, 比较不同算法的运行时间。

### 1. 解决思路:

可以使用向量化操作来计算给定公式的值, 避免使用循环语句。

创建一个从 1 到  $10^6$  的向量 i。

计算  $\sqrt{3} / i^2$  的和, 结果存储在 K\_no\_loop 中。

这种方法的优点是运行速度快, 因为 MATLAB 对向量和矩阵操作进行了优化。

### 2. 实验结果:

```
>> exp5
```

历时 0.011286 秒。

使用 for 循环计算 k: 2.8491

历时 0.008795 秒。

使用 while 循环计算 k: 2.8491

历时 0.005197 秒。

避免使用循环语句计算 k: 2.8491

### 3. 实验代码:

```
% 分别用 for 和 while 循环结构编写程序, 求出  $K = \sum_{i=1}^{10^6} \frac{\sqrt{3}}{i^2}$  的值.
```

装订线

% 并考虑一种避免循环语句的程序设计, 比较不同算法的运行时间

```
% 使用for循环计算
tic
K_for = 0;
for i = 1:10^6
    K_for = K_for + sqrt(3) / i^2;
end
toc
disp(['使用for循环计算k: ', num2str(K_for)]);
```

```
% 使用while循环计算
tic
K_while = 0;
i = 1;
while i <= 10^6
    K_while = K_while + sqrt(3) / i^2;
    i = i + 1;
end
toc
disp(['使用while循环计算k: ', num2str(K_while)]);
```

```
% 避免使用循环语句
% 向量化计算
tic
i = 1:10^6;
K_no_loop = sum(sqrt(3) / i.^2);
toc
disp(['避免使用循环语句计算k: ', num2str(K_no_loop)]);
```

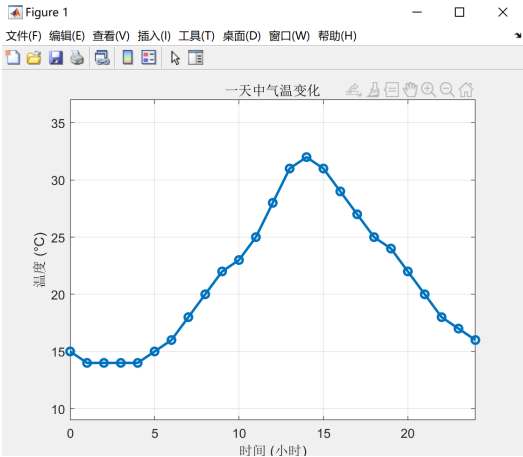
实验 6 假定某天的气温变化记录如表格, 试用 MATLAB 绘制气温变化曲线图

1. 解决思路:

将气温数据存储在一个数组中, 然后使用 MATLAB 的 plot 函数来绘制气温变化曲线图。

2. 实验结果

(1) 一天中的气温变化规律作图如下:



(2) 用 help 或 doc 查询 MATLAB 指令 dlmwrite 的使用方法:

装  
订  
线



(3) 导入数据:

1	2	3	4	5	6	7	8	9	10	11	12	13
VarName1	VarName2											
1	0	15										
2	1	14										
3	2	14										
4	3	14										
5	4	14										
6	5	15										
7	6	16										
8	7	18										
9	8	20										
10	9	22										
11	10	23										
12	11	25										
13	12	28										

3. 实验代码:

```
% 假定某天的气温变化记录如表格, 试用MATLAB绘制气温变化曲线图
% (1) 作图描述天气变化规律
time = 0:24;
temperature = [15, 14, 14, 14, 14, 15, 16, 18, 20, 22, 23, 25, 28, 31, 32, 31, 29, 27, 25, 24, 22, 20, 18, 17, 16];
figure;
plot(time, temperature, '-o', 'LineWidth', 2);
title('一天中气温变化');
xlabel('时间 (小时)');
ylabel('温度 (°C)');
grid on;
xlim([0 24]);
ylim([min(temperature) - 5, max(temperature) + 5]);
saveas(gcf, 'temperature_variation.png');

% (2) 用help或doc查询MATLAB指令dlmwrite的使用方法。
help dlmwrite
doc dlmwrite

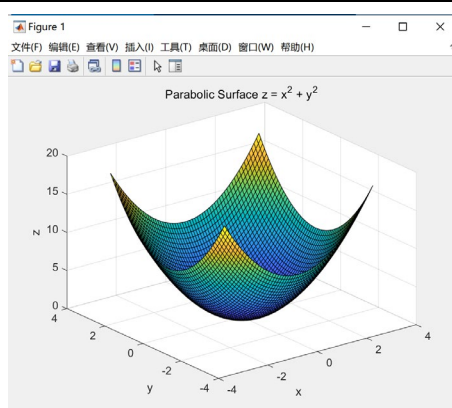
% (3) 用dlmwrite将(1)的数据输入到一个文本文件中
data = [time; temperature];
filename = 'temperature_data.txt';
dlmwrite(filename, data, 'delimiter', ',', 'precision', '%.2f');

% (4) 从工具条HOME中选择Import Data导入上述数据文件中的数据
```

实验 7 作出下列函数图像

(3) 抛物面  $z = x^2 + y^2, |x| < 3, |y| < 3$

1. 实验结果:



## 2. 实验代码:

```
% 作出下列函数图像
% (3) 抛物面  $z = x^2 + y^2$ ,  $|x| < 3$ ,  $|y| < 3$ 

% 定义x和y的范围
x = -3:0.1:3;
y = -3:0.1:3;

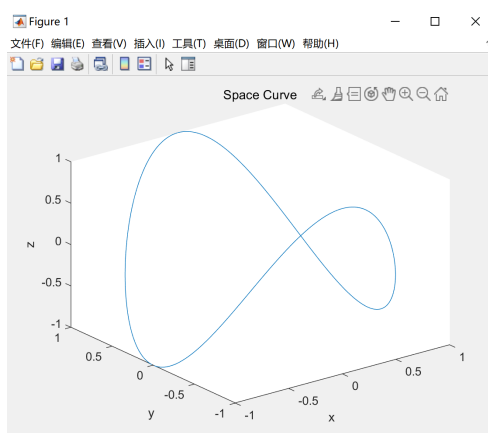
% 生成网格
[X, Y] = meshgrid(x, y);

% 计算z的值
Z = X.^2 + Y.^2;

% 绘制抛物面
surf(X, Y, Z);
title('Parabolic Surface  $z = x^2 + y^2$ ');
xlabel('x');
ylabel('y');
zlabel('z');
```

(5) 空间曲线  $x = \sin t$ ,  $y = \cos t$ ,  $z = \cos(2t)$ ,  $0 < t < 2\pi$

## 1. 实验结果:



## 2. 实验代码:

```
% 作出下列函数图像
% (5) 空间曲线  $x = \sin t, y = \cos t, z = \cos(2t), 0 < t < 2\pi$ 

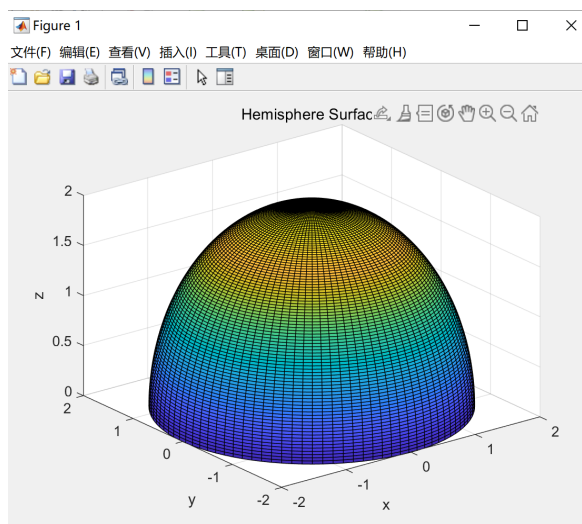
% 定义t的范围
t = 0:0.01:2*pi;

% 计算x, y, z的值
x = sin(t);
y = cos(t);
z = cos(2*t);

% 绘制空间曲线
plot3(x, y, z);
title('Space Curve');
xlabel('x');
ylabel('y');
zlabel('z');
```

(6) 半球面  $x = 2\sin(\phi)\cos(\theta), y = 2\sin(\phi)\sin(\theta), z = 2\cos(\phi),$   
 $0 \leq \theta \leq 2\pi, 0 \leq \phi \leq (1/2)\pi$

## 1. 实验结果:



## 2. 实验代码:

```
% 作出下列函数图像
% (6) 半球面  $x = 2\sin(\phi)\cos(\theta), y = 2\sin(\phi)\sin(\theta), z = 2\cos(\phi), 0 \leq \theta \leq 2\pi, 0 \leq \phi \leq (1/2)\pi$ 

% 定义\theta和\phi的范围
theta = linspace(0, 2*pi, 100);
phi = linspace(0, pi/2, 100);

% 生成网格
```

装  
订  
线

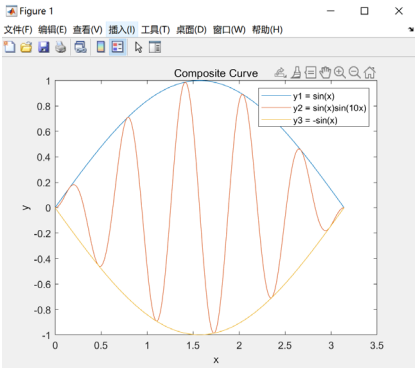
```
[Theta, Fi] = meshgrid(theta, fi);
```

```
% 计算x, y, z的值
X = 2 * sin(Fi) .* cos(Theta);
Y = 2 * sin(Fi) .* sin(Theta);
Z = 2 * cos(Fi);

% 绘制半球面
surf(X, Y, Z);
title('Hemisphere Surface');
xlabel('x');
ylabel('y');
zlabel('z');
```

(7) 三条曲线合成图  $y_1 = \sin x$ ,  $y_2 = \sin x \sin(10x)$ ,  $y_3 = -\sin x$ ,  $0 < x < \pi$

1. 实验结果:



2. 实验代码:

```
% 作出下列函数图像
% (7) 三条曲线合成图  $y_1 = \sin x$ ,  $y_2 = \sin x \sin(10x)$ ,  $y_3 = -\sin x$ ,  $0 < x < \pi$ 

% 定义x的范围
x = linspace(0, pi, 1000);

% 计算y的值
y1 = sin(x);
y2 = sin(x) .* sin(10*x);
y3 = -sin(x);

% 绘制曲线
plot(x, y1, x, y2, x, y3);
title('Composite Curve');
xlabel('x');
ylabel('y');
legend('y1 = sin(x)', 'y2 = sin(x)sin(10x)', 'y3 = -sin(x)');
```

## 实验8 用 MATLAB 函数表示下列函数, 并作图

$$p(x, y) = 0.5457 \exp(-0.75y^2 - 3.75x^2 - 1.5x), \quad x + y > 1$$

$$p(x, y) = 0.7575 \exp(-y^2 - 6x^2), \quad -1 < x + y \leq 1$$

$$p(x, y) = 0.5457 \exp(-0.75y^2 - 3.75x^2 + 1.5x), \quad x + y \leq -1$$

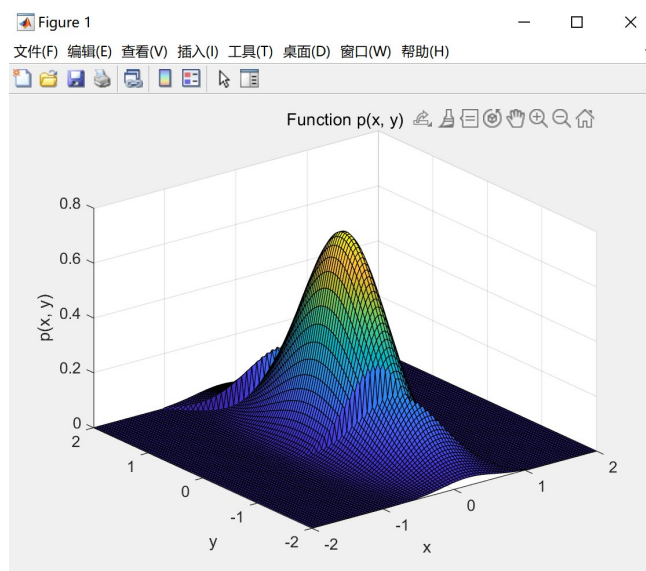
## 1. 解题思路:

初始化: 定义  $x$  和  $y$  的范围, 然后使用 `meshgrid` 函数生成网格。

计算函数值: 根据给定的函数表达式和条件, 计算每个  $(x, y)$  点对应的函数值。这里使用了逻辑运算符来实现条件判断。

绘制函数图像: 使用 `surf` 函数绘制函数图像, 并添加了标题和坐标轴标签。

## 2. 实验结果:



## 3. 实验代码:

```
% 用 MATLAB 函数表示下列函数, 并作图
% p(x, y) = 0.5457exp(-0.75y^2 - 3.75x^2 - 1.5x), x + y > 1
% p(x, y) = 0.7575exp(-y^2 - 6x^2), -1 < x + y <= 1
% p(x, y) = 0.5457exp(-0.75y^2 - 3.75x^2 + 1.5x), x + y <= -1

% 定义x和y的范围
x = linspace(-2, 2, 100);
y = linspace(-2, 2, 100);

% 生成网格
[X, Y] = meshgrid(x, y);

% 计算p的值
P1 = 0.5457 * exp(-0.75 * Y.^2 - 3.75 * X.^2 - 1.5 * X) .* (X + Y > 1);
P2 = 0.7575 * exp(-Y.^2 - 6 * X.^2) .* (-1 < X + Y & X + Y <= 1);
```



```
P3 = 0.5457 * exp(-0.75 * Y.^2 - 3.75 * X.^2 + 1.5 * X) .* (X + Y <= -1);
```

```
% 绘制函数
surf(X, Y, P1 + P2 + P3);
title('Function p(x, y)');
xlabel('x');
ylabel('y');
zlabel('p(x, y)');
```

### 五、实验总结

MATLAB 在数值分析方面具有强大的数学函数库、高效的向量和矩阵运算能力、丰富的绘图功能、专业的工具箱、良好的兼容性以及易于学习和使用的特点，使其在处理大量数据、进行复杂计算和专业分析时具有显著优势。

通过这些实验，我们学习了 MATLAB 的许多重要功能，包括如何定义变量，如何进行数学运算，如何绘制函数图像，以及如何使用条件语句。这些都是进行科学计算和数据分析的基础技能。

装  
订  
线