

重 庆 大 学

学 生 实 验 报 告

实验课程名称 数学模型

开课实验室 D1128

组员 1 姓 名 马梓恒 学 号 20233124

组员 2 姓 名 周宏仰 学 号 20232647

组员 3 姓 名 郑祺耀 学 号 20230692

开 课 时 间 2024 至 2025 学年第 一 学期

总 成 绩	
-------	--

数 统 学 院 制

课程名称	数学模型	实验项目名称	MATLAB 作图	实验项目类型				
				验证	演示	综合	设计	其他
指导教师	肖剑	成绩			√			

题目 1

1 编写函数 M-文件：用迭代法求 $x = \sqrt{a}$ 的值。求平方根的迭代公式为

$$x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n})$$

迭代的终止条件为前后两次求出的 x 的差的绝对值小于 10^{-5} 。

程序

```
% 初始化数据
a=2;
x=10000;
cha=1;
ci=0;

% 循环
while abs(cha)>=0.00001
    x1=1/2*(x+a/x);
    cha=x1-x;
    x=x1;
    ci=ci+1;
end

fprintf("迭代次数为%d,x 值为%.7f\n",ci,x)
```

结果

```
>> fun1
迭代次数为 17,x 值为 1.4142136
>>
```

分析

初始化：将要计算平方根的数 a 设置为 2，初始猜测值 x 设为 10000，cha 用于记录每次迭代前后 x 的差，ci 用于计数迭代次数。

迭代过程：使用 while 循环，不断根据迭代公式 $x1 = 1/2 * (x + a/x)$ 计算新的 x 值，直到前后两次迭代的差的绝对值 cha 小于 0.00001。

结果输出：当迭代结束后，程序输出迭代次数和最终的 x 值，结果为 17 次迭代后得出的平方根值为

1.4142136, 接近实际的 $\sqrt{2}$ 。

收敛性：该算法快速收敛，显示了迭代法在计算平方根时的高效性和准确性

题目 2

在同一个坐标下作出 $y_1 = e^x$, $y_2 = 1 + x$, $y_3 = 1 + x + \frac{1}{2}x^2$, $y_4 = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3$ 这四条曲线的图形，

要求在图上加各种标注，同时用 subplot 作出这四条曲线，为每幅图形加上标题。

程序

```
x = -2:0.01:2;

y1 = exp(x);
y2 = 1 + x;
y3 = 1 + x + (1/2) * (x.^2);
y4 = 1 + x + (1/2) * (x.^2) + (1/6) * (x.^3);

figure;
plot(x, y1, 'r', 'LineWidth', 1.5); hold on;
plot(x, y2, 'g', 'LineWidth', 1.5);
plot(x, y3, 'b', 'LineWidth', 1.5);
plot(x, y4, 'm', 'LineWidth', 1.5);
legend('y1 = e^x', 'y2 = 1 + x', 'y3 = 1 + x + (1/2) * x^2', 'y4 = 1 + x + (1/2) * x^2 + (1/6) * x^3');
xlabel('x'); ylabel('y');
title('Four Curves on the Same Plot');
grid on;

figure;

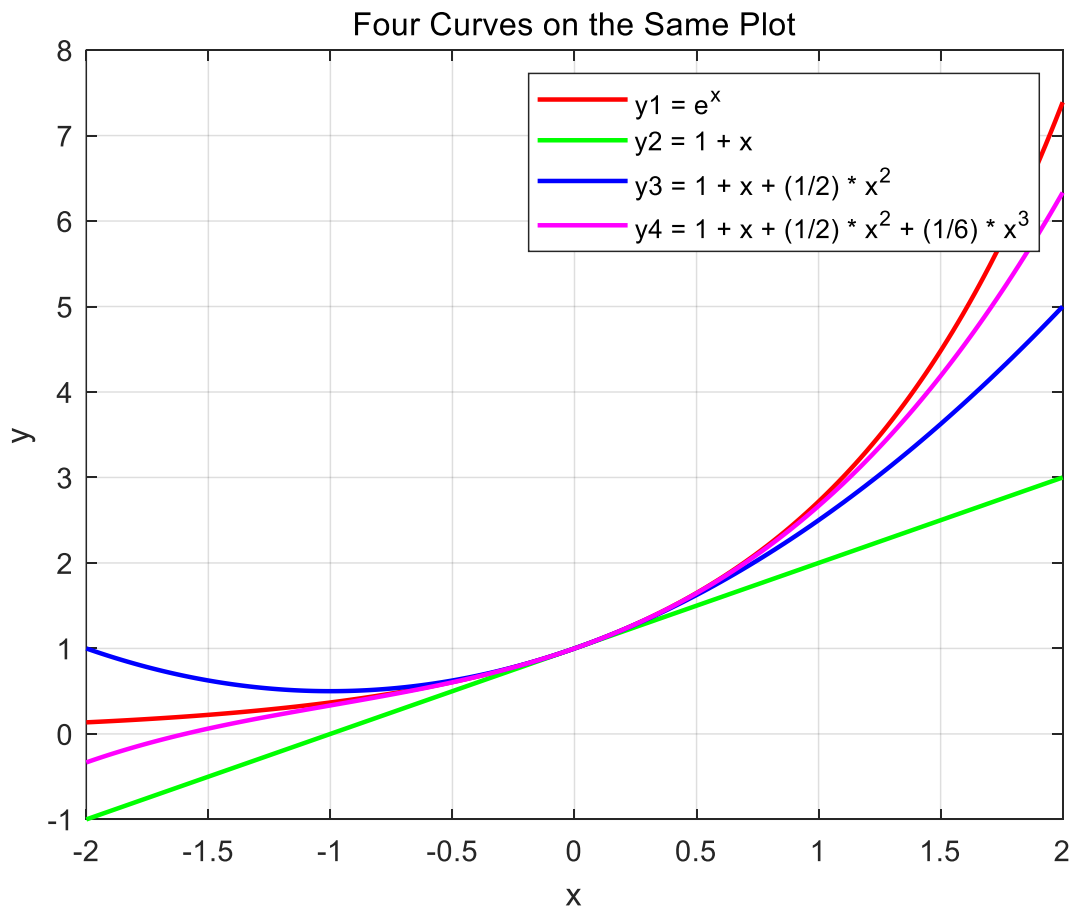
subplot(2, 2, 1);
plot(x, y1, 'r', 'LineWidth', 1.5);
title('y1 = e^x');
xlabel('x'); ylabel('y');
grid on;

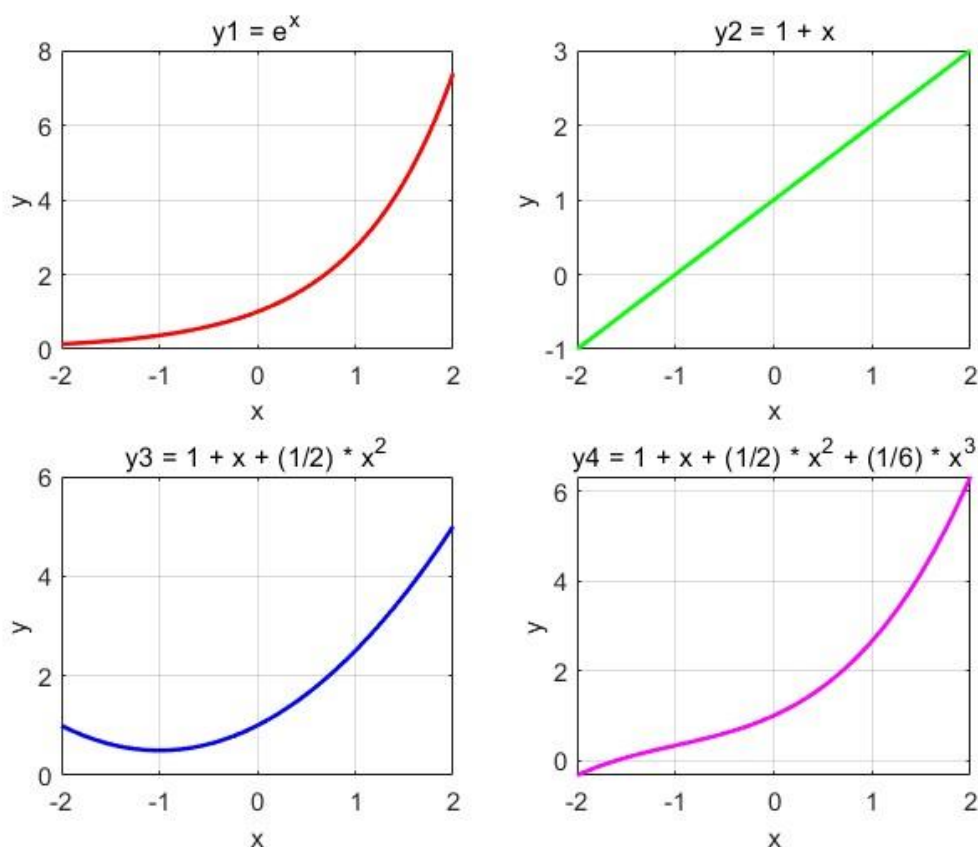
subplot(2, 2, 2);
plot(x, y2, 'g', 'LineWidth', 1.5);
title('y2 = 1 + x');
xlabel('x'); ylabel('y');
grid on;

subplot(2, 2, 3);
plot(x, y3, 'b', 'LineWidth', 1.5);
```

```
title('y3 = 1 + x + (1/2) * x^2');  
xlabel('x'); ylabel('y');  
grid on;  
  
subplot(2, 2, 4);  
plot(x, y4, 'm', 'LineWidth', 1.5);  
title('y4 = 1 + x + (1/2) * x^2 + (1/6) * x^3');  
xlabel('x'); ylabel('y');  
grid on;
```

结果





分析

首先，在 MATLAB 中定义变量范围为 $x = -2:0.01:2$ ，接着定义四个函数，其中 $y_1 = e^x$ ， $y_2 = 1 + x$ ， $y_3 = 1 + x + \frac{1}{2}x^2$ ， $z = y^2 - 3xy + x^2$ 。然后使用 `figure` 命令创建一个新的图形窗口，通过 `plot` 函数在同一坐标系中绘制四条曲线，并分别用红色、绿色、蓝色和洋红色表示这四个函数的变化情况，利用 `hold on` 命令确保后续绘图不会清除之前的曲线。接着为每条曲线添加图例，并设定 x 和 y 轴的标签，添加标题，并使用 `grid on` 命令启用网格，以使图形更加清晰。随后，再次使用 `figure` 创建新的图形窗口，并调用 `subplot(2, 2, n)` 来划分一个 2x2 的网格，分别在每个子图中绘制每个函数。通过这种方式，可以更详细地观察每条曲线的特性，设置合适的标题和标签，并为每个子图启用网格，以便于分析。这样就完成了四个函数在同一坐标系及各自子图的绘制，便于比较和观察它们的变化情况。

题目 3

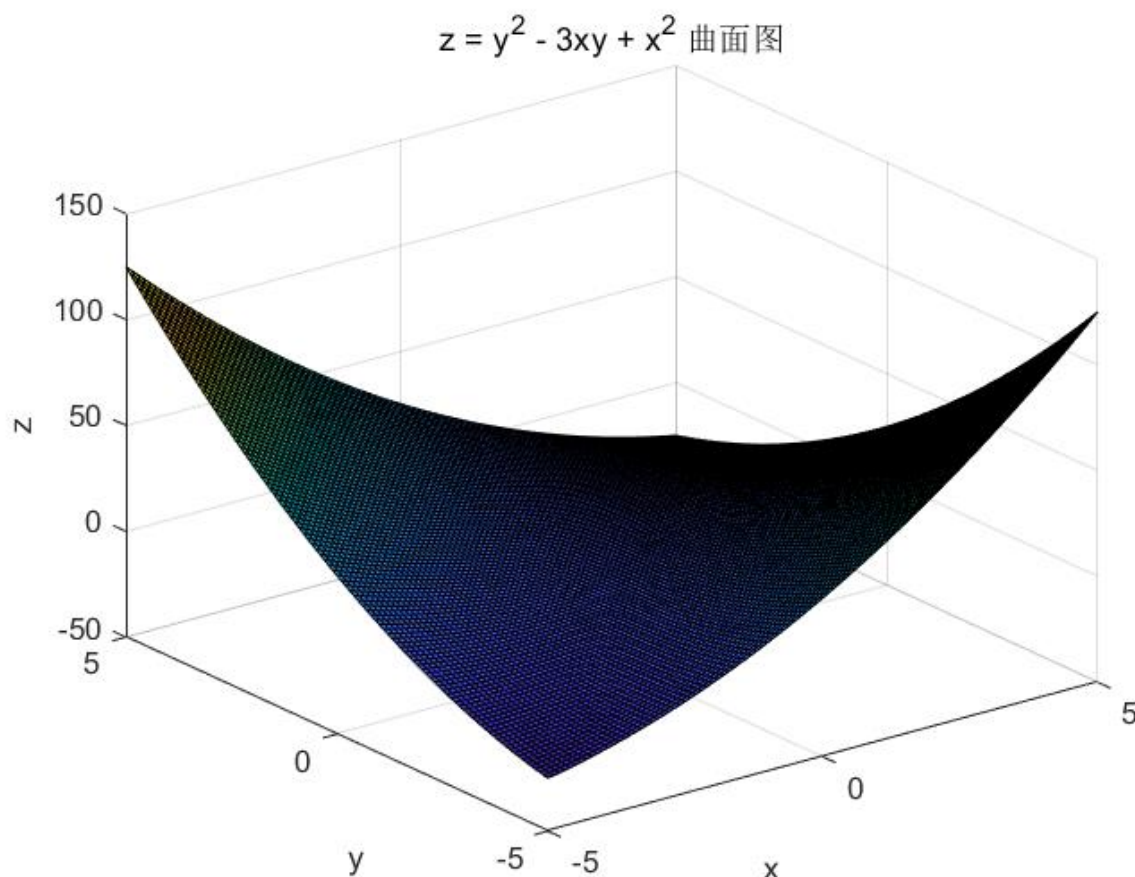
绘制如下函数 $z = y^2 - 3xy + x^2$ 曲面图， x, y 的取值范围可自行选择。

程序

```
[x, y] = meshgrid(-5:0.1:5, -5:0.1:5);

z = y.^2 - 3.*x.*y + x.^2;
```

```
figure;  
surf(x, y, z);  
xlabel('x');  
ylabel('y');  
zlabel('z');  
title('z = y^2 - 3xy + x^2 曲面图');
```



分析

这段代码首先通过 `meshgrid` 函数定义了 x 和 y 的取值范围，创建了一个从 -5 到 5 的网格，其间隔为 0.1，这样就生成了两个矩阵 x 和 y ，其中每个元素都对应于 x 和 y 的位置。接着，代码计算了 z 的值，公式为 $z = y^2 - 3xy + x^2$ ，生成了一个与网格相匹配的 z 矩阵，这显示了 z 在每个 (x, y) 点的值。随后，使用 `figure` 函数打开一个新的图形窗口，并调用 `surf` 函数绘制三维曲面图，基于 x 、 y 和 z 的矩阵数据来展示 z 随 x 和 y 的变化情况，使得图形具有立体感和层次感。最后，代码通过 `xlabel`、`ylabel` 和 `zlabel` 函数添加了相应的坐标轴标签，进一步通过 `title` 函数为图形添加了标题，以清晰地描述该曲面图所表示的数学关系。这样，整个代码段实现了从数据定义到图形呈现的完整流程，直观展示了该函数的形状和特性。

备注：

- 1、一门课程有多个实验项目的，应每一个实验项目一份，课程结束时将该课程所有实验项目

内页与封面合并成一个电子文档上交。