组　　号	Z08
实验题目	数据结果的可视化实验
队员姓名	1. 宋昭琰 2. 徐意 3. 李宗琳 4. 向飞宇

目　录

1 问题一：同一坐标系作多个函数图像 **2**

2 问题二：绘制变上限积分图像 **2**

2.1 根据定积分定义求解........ 2

2.2 使用 MatLab 中积分函数 int 求解 3

3 问题三：绘制二元函数的表面图，等高线图，伪彩图和着色表面图 **3**

4 问题四：绘制三元函数的各种切片图 **4**

5 问题五：绘制随时间变化的三元函数动态图 **6**

6 问题六：绘制常微分方程的线素场 **6**

7 问题七：绘制多个迭代点的图形 **8**

7.1 使用 MatLab 做出迭代图形 8

7.2 对于该图像的进一步解释........ 8

8 问题八：绘制隐函数的图形 **9**

9 问题九：根据数据绘制散点图 **9**

参考文献 **11**

附录 A 关于动态图的表示 **11**

附录 B 实验代码 **12**

B.1 问题一的代码 12

B.2 问题二的代码 12

B.3 问题三的代码 13

B.4 问题四的代码 14

B.5 问题五的代码 15

B.6 问题六的代码 16

B.7 问题七的代码 17

B.8 问题八的代码 17

B.9 问题九的代码 18

1 问题一：同一坐标系作多个函数图像

问题描述： 在同一坐标系中画出 $y = x^2, y = \sin x$ 的图象.

本题要求在同一坐标系下作出两个函数的图像，这说明在画图的时候需要加入 hold on 参数使得一张图片上显示两个不同函数的内容，或者利用 plot 函数对多条曲线同时作图. 同时需要对不同的函数加入相应的图例，由此可以得到作图的代码. 有多种方式作出函数图像，既可以自己根据函数构造数据点，使用 MatLab 中的 plot 函数对这些点进行插值. 构造的数据点越多，图像越精确. 也可以利用 MatLab 中 fplot 或 ezplot 函数，直接将函数作出. 以下图像使用**多条曲线同时作图 + plot 函数**的方法作出.

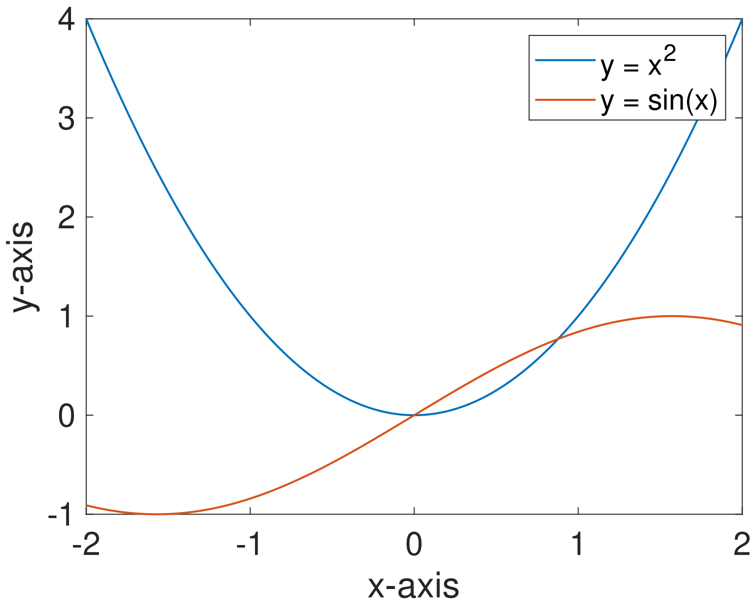


图 1　同一坐标系作 $y = x^2, y = \sin x$ 图像

2 问题二：绘制变上限积分图像

问题描述： 利用定积分定义，绘制一元函数 $f(x) = \int_0^{\sin x} (1 - t^2)^{3/2}\,\mathrm{d}t$ 在 $[0, 4\pi]$ 的图形.

2.1 根据定积分定义求解

定积分的计算方式一般可以分为：**分割，近似，求和，取极限**四个步骤，MatLab 中不提供取极限的功能，所以需要通过分割区间的加细完成对定积分的计算. 为方便计算，本题中使用**等长分割，用矩形面积近似曲边梯形面积**的思路，对每一个给定的 x，将积分区间 $[0, \sin x]$ 分为 1000 个小区间，在这些小区间上分别计算函数 $g(t) = (1 - t^2)^{3/2}$ 的

值，然后乘以区间长度 $\frac{\sin x}{1000}$ 作为对这个积分的近似. 具体而言，计算公式为：

$$f(x) = \sum_{i=0}^{999} (1 - \frac{ix}{1000}^{2})^{3/2} \frac{\sin x}{1000} \tag{1}$$

根据以上公式编写程序得到实验结果，如下图所示：

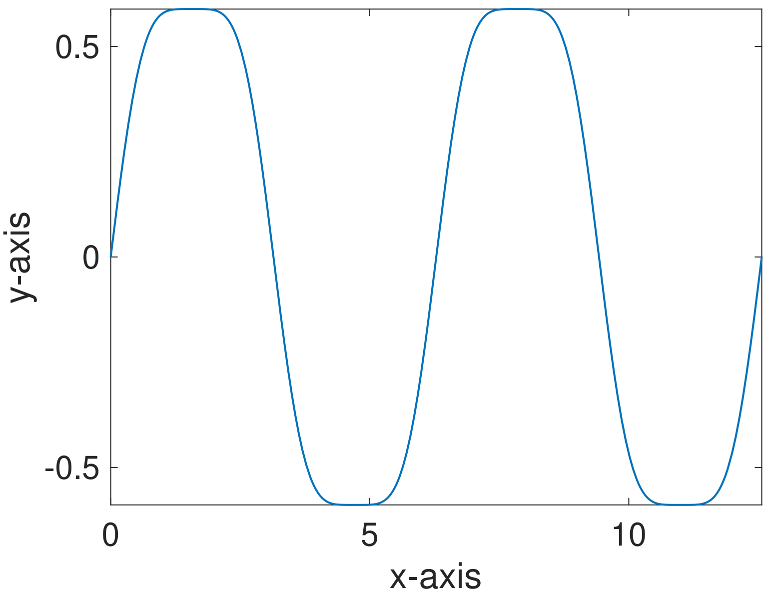


图 2 $f(x) = \int_0^{\sin x} (1 - t^2)^{3/2}\, \mathrm{d}t$ **在** $[0, 4\pi]$ **的图形**

2.2 使用 MatLab 中积分函数 int 求解

本题要求作出定积分形式的一元函数（含参数的定积分）的图形，考虑到使用定积分的定义，需要写出多个匿名函数，较为不方便，所以我们使用到 MatLab 内置的 int 函数求数值积分，同时使用 fplot 函数作出数值函数曲线，根据以上方式式编写程序得到实验结果，得到的图像与图（2）类似，故不作出.

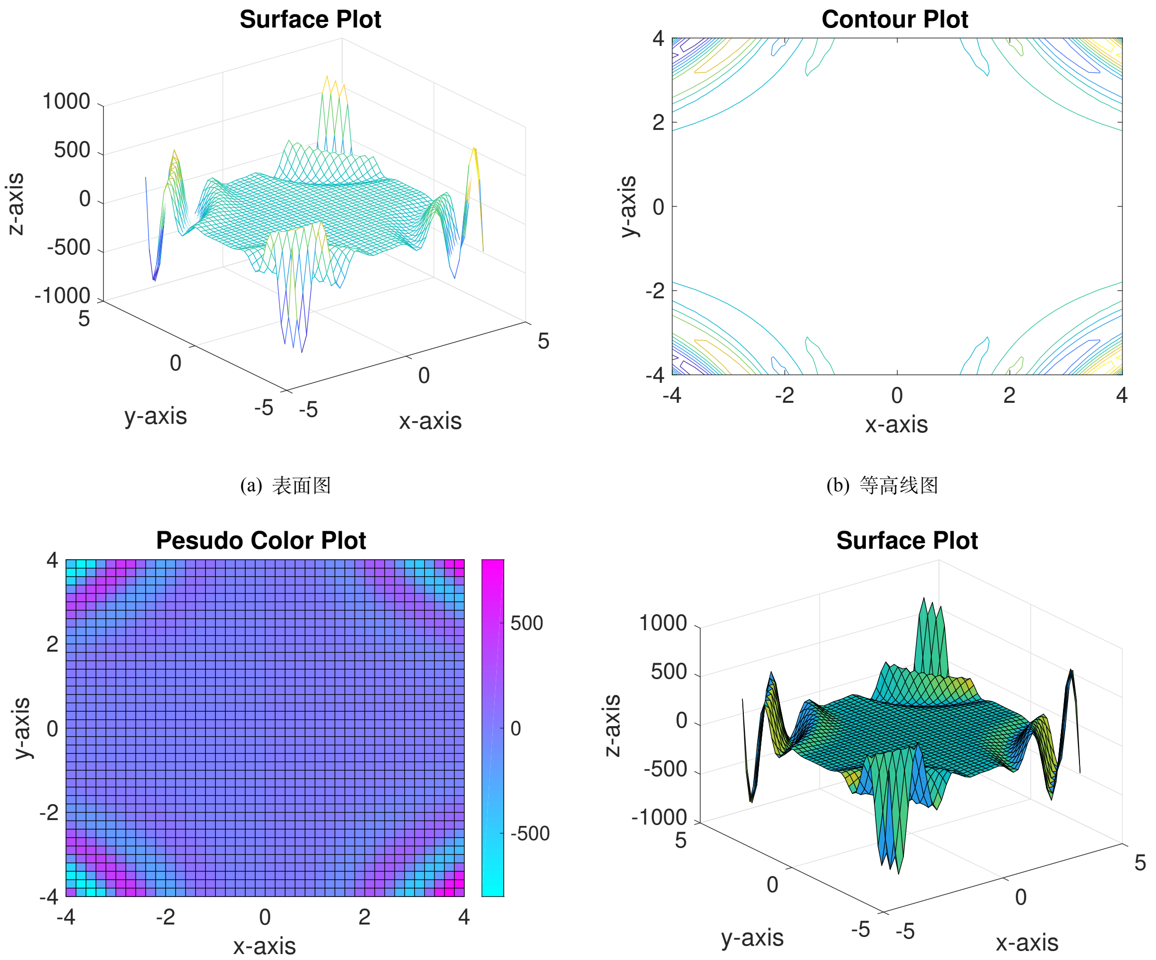
这种方式的代码量较之前的程序少了很多，这说明**适当使用内置函数是减少代码的一个选择**.

3 问题三：绘制二元函数的表面图，等高线图，伪彩图和着色表面图

问题描述：绘制二元函数 $f(x,y) = \sin(xy)x^2y^3$ 在 $[-4,4] \times [-4,4]$ 上的表面图，等高线图，伪彩图以及着色表面图.

为了方便对元素进行相乘（不采用矩阵乘法的方式），需要使用 Matlab 中的 meshgrid 函数用于生成网格采样点，这种处理方式在使用 Matlab 进行三维图形绘制方面有着广泛的应用. 然后，我们利用 mesh 函数绘制由线条框构成的曲面，从而得到表面图，接着

分别利用 contour，pcolor，surf 函数得到轮廓图，伪彩图和着色表面图，实验结果如下所示：



(a) 表面图

(b) 等高线图

(c) 伪彩图

(d) 着色表面图

图 3　表面图，等高线图，伪彩图和着色表面图的绘制

4 问题四：绘制三元函数的各种切片图

问题描述： 绘制三元函数 $f(x,y,z)=x^2y^3z^4$ 在 $[-1,1]\times[-2,2]\times[-3,3]$ 上的各种切片图.

由于三元函数不能直接画在一个三维平面中，需要固定一个 x（或 y,z）的值，并画出在那个值处函数的投影，从而就画出了切片图. 在 MatLab 中，切片图使用 slice 命令画出. 以下分别是该函数在各个方向上投影所构成的切片图.

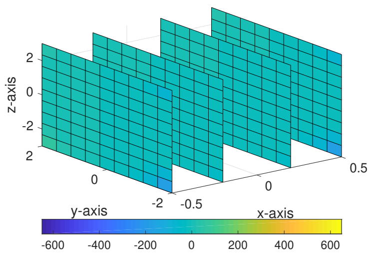
slice 函数不仅可以对正常的 x,y,z 轴进行切片处理，还可以对沿曲面的三维体数据进行切片，此时需要给出由曲面的方程所生成的网格点才能得到切片图. 这些切片图适用于查看三维函数较为“奇异”的区域，我们作出了函数在双曲抛物面 $z=x^2-y^2$ 上

的切片图.

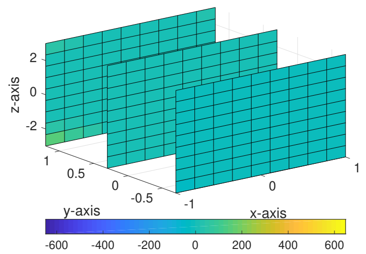
slice 函数运用了插值的方法，可以选择的插值方式有：

- linear: 在每个维度的相邻网格点进行线性插值；
- cubic: 在每个维度的相邻网格点进行三次插值；
- nearest: 最近的网格点值；

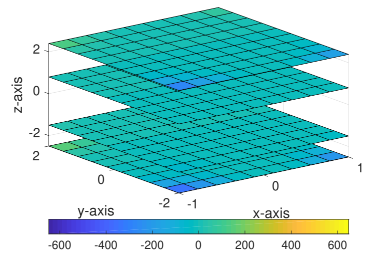
一般情况下，进行三次插值能够获得最佳的作图精度，而取最近的网格点值能够获得最快的计算速度，本题中的函数较为简单，所以不需要选择特殊的插值方法，使用普通的**线性插值**就足够了. 以下实验所得的各种切片图结果：



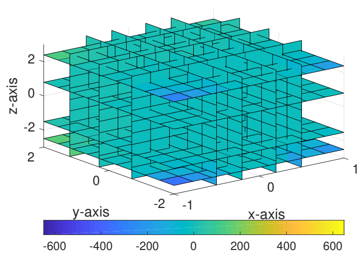
(a) 在 x 轴的切片图



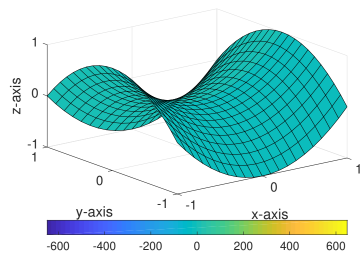
(b) 在 y 轴的切片图



(c) 在 z 轴的切片图



(d) 在 x, y, z 轴同时切片



(e) 以 $z = x^2 - y^2$ 平面作切片图

图 4　各种切片图的绘制

由这一些切片图可以看出，这个函数在 $[-1,1] \times [-1,1] \times [-1,1]$ 上变化很慢，而在其余的区间上变化就很快，这可以从该函数对 x, y, z 的偏导数上看出：

$$\frac{\partial f}{\partial x} = 2xy^3z^4, \quad \frac{\partial f}{\partial y} = 3x^2y^2z^4, \quad \frac{\partial f}{\partial z} = 4x^2y^3z^3$$

当 $|x|, |y|, |z| \geqslant 1$ 时，这些偏导数的值均大于 1 且不断增大，从而原函数的增速也会越来越快，这就是在切片图中角落的方格颜色与中间的方格颜色不同的原因.

5 问题五：绘制随时间变化的三元函数动态图

问题描述： 绘制三元函数 $f(x,y,t)=10\,\mathrm{e}^{1-t}\sin x\cos y$ 在 $x\times y=[-2\pi,2\pi]\times[-2\pi,2\pi]$ 上随 $t\in[0,10]$ 变化的动态图形.

本题要求作出三元函数的动态图形，我们对时间取大量的点，利用循环将每一个时间点的图形画出来，然后使用 drawnow 进行实时画图. 由于函数值的变化比较大，我们需要利用 axis 来固定 z 轴才能看出这个三元函数值的变化；若不固定坐标轴，z 轴将随着函数值改变刻度，视觉上的动态变化与实际的动态变化有所出入，会出现**视觉上觉得函数值在波动，实际上函数值在不断减小的问题.** 同时利用 surf 函数作出三维着色表面图能更好地观测到图形的动态变化，由此可以得到作图的代码. 我们根据函数图像与时间的变化规律，选取了 $t=0,3,10$ 三个时刻 $f(x,y,t)$ 的着色表面图如下：

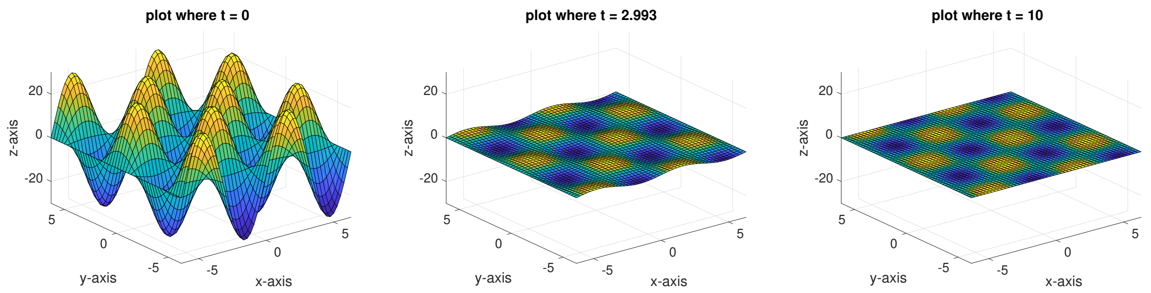


图 5　$f(x,y,t)$ 在 $x\times y=[-2\pi,2\pi]\times[-2\pi,2\pi]$ 上当 $t=0,3,10$ 时的图形

由所绘制的图像可以看出，这个函数的图像与 t 值密切相关，随着 t 的增大，函数值迅速减小，在 $t=5$ 时，函数值几乎为 0，所以从 $t=5$ 至 $t=10$，函数基本上就是在 xOy 上的一个“平面”（这也是没有取 $t=5$ 之后的图像作为示例的原因）. 对于函数中的 x,y 项，由于 $\sin x,\cos y$ 均是周期函数，在 t 较大时，函数的图像就很像平面上的“光斑”，而 x,y 的值则决定了“光斑”的方位，这就是上图中出现周期性“光斑”的原因.

6 问题六：绘制常微分方程的线素场

问题描述： 已知微分方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=\sin(xy)$，画出其线素场.

为了完成题目要求，我们需要给出线素场与等斜线的定义[1]：

定义 6.1 (线素场) 考虑一阶微分方程：

$$\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y) \tag{2}$$

其中 $f(x,y)$ 是平面区域 G 内的连续函数. 在 G 内每一点 $P(x,y)$，可以作一个以 $f(P)$ 为斜率的直线段 $l(P)$，以标明积分曲线在该点的切线方向. 称 $l(P)$ 为微分方程 (2) 在 P 点的线素，称区域 G 连通全体线素为微分方程的线素场.

定义 6.2 (等斜线) 由关系式 $f(x,y)=k$ 确定的曲线 L_k，称它为线素场的等斜线.

由以上的定义可以看出，等斜线上的每一个点的线素（方向）平行. 以下是作出

本题要求作出微分方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=\sin(xy)$ 的线素场，在这里先用 meshgrid 函数生成网格采样点，然后使用 contour 函数能够绘制出该函数的等值线. 最后用 quiver 函数绘制二维矢量，quiver 函数的用法如下：

quiver(x, y, u, v)

其中 x,y,u,v 是大小相等的矩阵，在 (x,y) 点处绘制 x 方向长为 u，y 方向长为 v 的箭头，基于本题中对线素场的定义，需要设定 $u=1,v=\sin(xy)$，通过固定 x 方向的长度来统一斜率的计算.

除此以外，在绘制过程中，使用 hold on 函数使轮廓图与所生成的箭头生成的内容在一张图上.

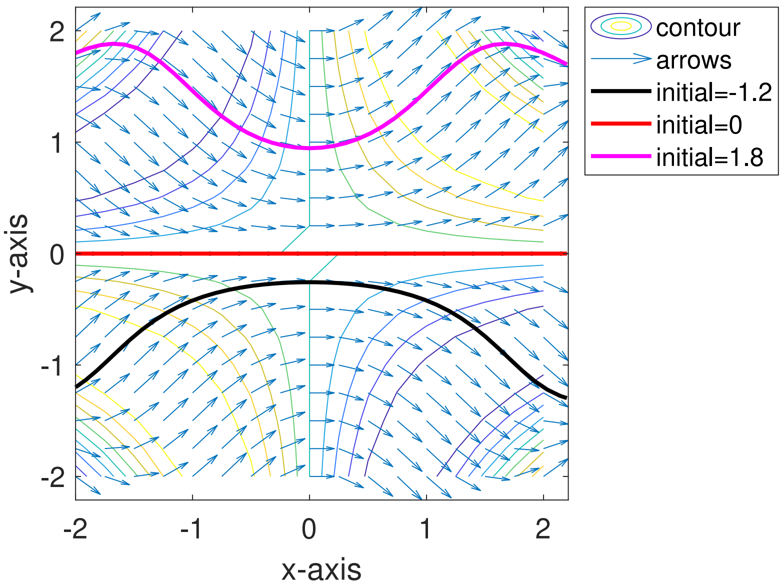


图 6　微分方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=\sin(xy)$ 的线素场，等值线与积分曲线图

由图像可以看出，这个图形是中心对称的，而且满足等斜线上的每一个点的线素（方向）平行，从图中看，方程有一个零解 $y=0$，与实际情况相符. 为了进一步验证试验结果，我们分别以 $(x,y)=(-2,-1.2)$，$(-2,0)$ 与 $(-2,1.8)$ 为初值，使用 MatLab 中的 ode45 作出这个微分方程数值上的积分曲线（解），发现积分曲线上的点的线素与积分曲线相切.

7 问题七：绘制多个迭代点的图形

问题描述： 以 (x_0, y_0) 为初始点，以

$$\begin{cases} x_n = ax_{n-1} + by_{n-1} + e \\ y_n = cx_{n-1} + dy_{n-1} + f \end{cases}$$

为迭代格式，其中系数 $[a, b, c, d, e, f]$ 以 $p = [0.3333, 0.3333, 0.3334]$ 为概率分布分别选取如下系数:

$$M = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0.25 & 0.5 \end{bmatrix}$$

绘出所有迭代点所形成的图形.

7.1 使用 MatLab 做出迭代图形

系数的大小由矩阵 M 决定，且每一行出现的概率是等可能的. 采用 MatLab 中生成随机整数的 randi 函数可以保证随机选取三种可能的情况. 接下来，我们采取一个循环结构，保证后一步的位置与走法和前一步的位置相关. 我们可以将迭代次数取到一个很大的值，由**大数定律**，每种系数出现的频率可以无限趋近于概率，所有迭代点采取的迭代格式中的参数出现的频率将会基本与题目要求的概率相同. 本题中取**总迭代次数**为 1000，并给出经过 500 次迭代后的图形与 1000 次迭代后的图形如下：

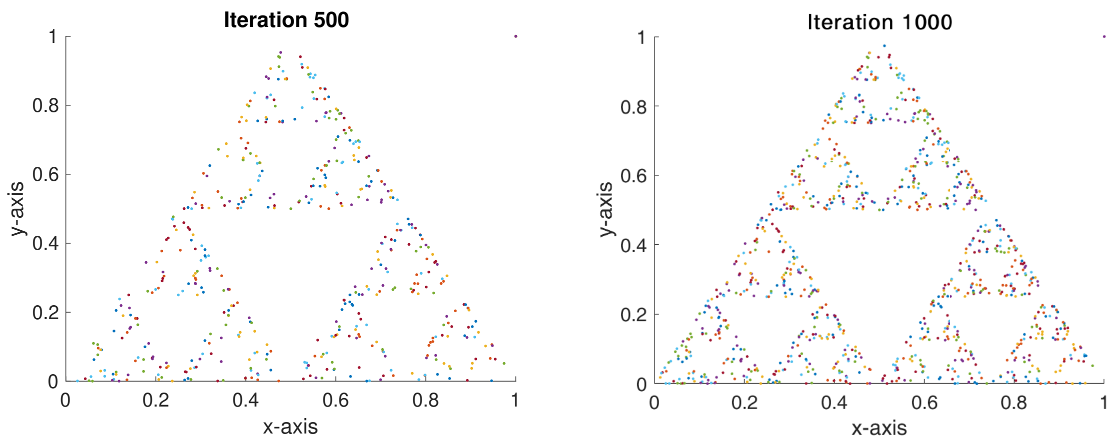


图 7　迭代 500, 1000 **次后形成的图形**

7.2 对于该图像的进一步解释

事实上，以上的图形是**谢尔平斯基三角形**（Sierpinski triangle）的数值生成结果[2]，对于这个三角形有一个几何上的生成方法：

1. 取一个实心的等边三角形.
2. 沿三边中点的连线，将它分成四个小三角形.
3. 去掉中间的那一个小三角形.
4. 对其余三个小三角形重复步骤 1.

这个"三角形"是自相似的，也是分形的一个经典示例，下面采用 TikZ 中的 lindenmayersystems 做出一个较为具体的图（迭代 6 步）：

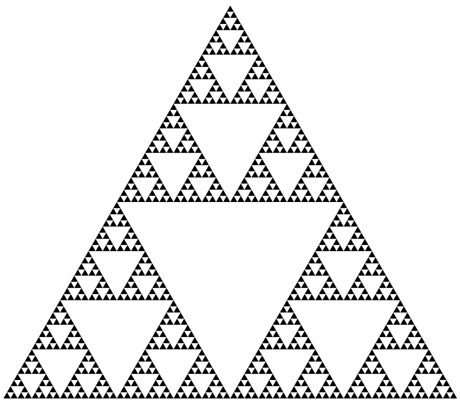


图 8　使用 TikZ 作谢尔平斯基三角形

8 问题八：绘制隐函数的图形

问题描述： 绘制隐函数 $\sin(xy) + x + y = 0$ 的图形.

首先我们发现，通过 MatLab 中的 ezplot 函数，可以作出任意符号函数在给定区间上的图像，但是 ezplot 已经被 MatLab 列入不推荐使用名单. 通过查找资料，MatLab 提供了一个绘制隐函数的函数：fimplicit，它的使用方法是:

fimplicit(function_handle, interval).

这样，将需要绘制的函数写成**二元匿名函数**的形式：$@(x, y)\,\text{sin}(x. * y) + x + y$，并设置绘制区间为 $[-5, 5] \times [-5, 5]$ 即可作出图像如图（9）所示.

9 问题九：根据数据绘制散点图

问题描述： data.mat 数据文件是某一工程中测量的位置数据（其中 NaN 标识漏测的数据），请在平面直角坐标系下，绘制给定数据的散点图（除 NaN）.

虽然 MatLab 在绘制散点图时会自动忽略 NaN 的数据，但为了之后处理数据的方便，首先要找出 NaN 数据并删除之. 这里有两种等价的方法：

1. 对于二维数组我们这里直接用 find(isnan(x(1, :))) 与 find(isnan(x(1, :))) 函数来查找出所有 NaN 数据所在的对应行，然后根据行的坐标将含有 NaN 数据的行置空；

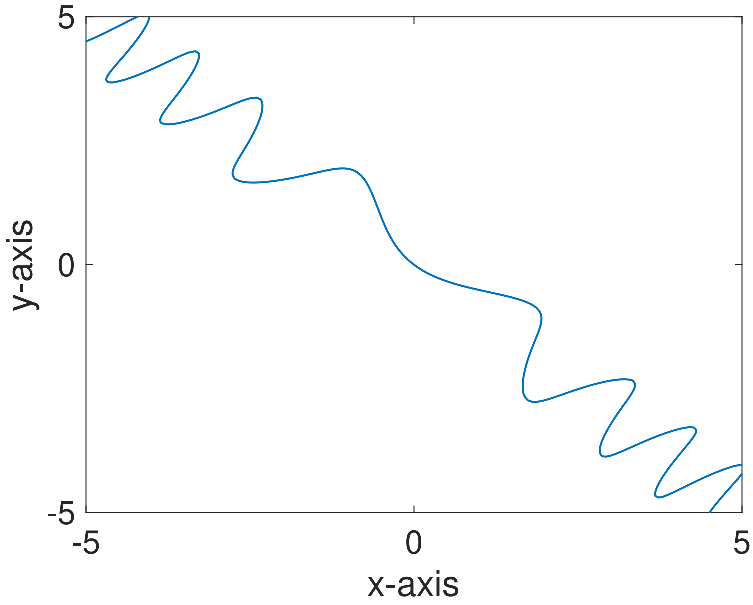


图 9　隐函数图像的绘制

2. 直接操作数组的索引，使用 x(isnan(x(1, :))) = [] 与 x(isnan(x(2, :))) = [] 来同时查找和删除数据.

根据 MatLab 的文档，第二种方式效率更高一些，而且它的代码量较少，故本题中使用第二种方式来查找和删除 NaN 数据.

接着，分别将两组数据命名为 x 与 y 行，并用 scatter(x, y) 函数画出散点图即可，scatter 函数可以指定绘图点的大小，本题中指定为 20 单位.

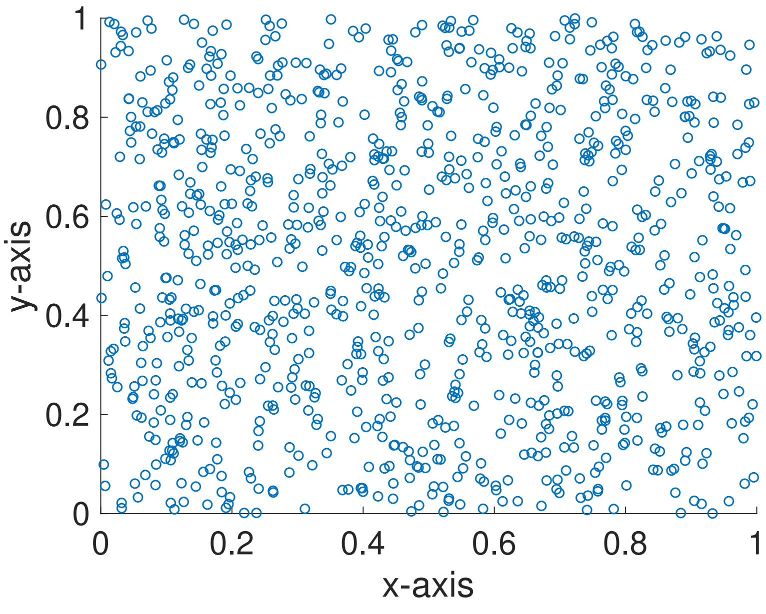


图 10　根据数据绘制散点图

参考文献

[1] 丁同仁，李承治. 常微分方程教程 (第 2 版)[M]. 高等教育出版社, 2004.

[2] Wikipedia contributors. (2019, March 10). Sierpinski triangle. In Wikipedia, The Free Encyclopedia. Retrieved 09:11, March 14, 2019, from https://en.wikipedia.org/w/index.php?title=Sierpinski_triangle&oldid=887051985

附录 A 关于动态图的表示

虽然动态图在 MatLab 中展现出来比较容易，但是将图片导出将会更有利于传播. 考虑到 GIF 格式能动态显示图片，我们将需要操作的多张图片一次保存为 GIF 格式. 保存时与 movie 函数非常类似：先使用 gcf 获得当前 Figure 对象的句柄值，然后将图像句柄转换为索引形式，最后使用 imwrite 的 append 模式，将每幅生成的图形依次至于上一张图片的末尾，具体实现的代码如下：

```
frame = getframe(gcf); % get frame
im = frame2im(frame); % convert frame to image handler
[imind, cm] = rgb2ind(im, 256); % convert image to index
% Write to the GIF File
if i == 1
    imwrite(imind, cm, 'test.gif', 'gif', 'Loopcount', inf,
     ↪  'DelayTime', 0.08);
else
    imwrite(imind, cm, 'test.gif', 'gif', 'WriteMode', 'append',
     ↪  'DelayTime', 0.08);
end
```

我们对基本的发布代码进行了定制，通过判断迭代次数决定是否对图片进行采样，通过修改时间间隔实现非线性采样，从而减少图片所占空间.

由于 GIF 格式的动态图无法在此文档中呈现，我们将其放在我们的网站上以供参考.

第五题图片：

https://t0nyx1ang.github.io/Mathematical-Experiment/exp1/experiment1_12.gif

第七题图片：

https://t0nyx1ang.github.io/Mathematical-Experiment/exp1/experiment1_14.gif

附录 B 实验代码

B.1 问题一的代码

```
%% Exercise 1
clear, clc;
x = linspace(-2, 2, 1000);
y1 = x.^2;
y2 = sin(x);
plot(x, y1, x, y2, 'linewidth', 1);
legend('y = x^2', 'y = sin(x)');
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);
```

B.2 问题二的代码

```
%% Exercise 2
% method 1
clear, clc;
interval = 1000;
func1 = @(x) sin(x);
func = @(t) (1 - t .^ 2) .^ (3 / 2);
x = linspace(0, 4 * pi, 100);
for i = 1:100
    t = linspace(0, func1(x(i)), interval + 1);
    y(i) = sum(func(t) * func1(x(i)) / interval);
end
figure(1);
plot(x, y, 'linewidth', 1);
xlabel('x-axis');
ylabel('y-axis');
xlim([0 4 * pi]);
set(gca, 'fontsize', 16);

% method 2
clear, clc;
```

```
syms x t;
y = (1 - t ^ 2) ^ (3 / 2);
z = int(y, t, 0, sin(x));
figure(2);
fplot(z, [0 4 * pi], 'linewidth', 1);
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);
```

B.3 问题三的代码

```
%% Exercise 3
clear; clc;
x=-4: 0.2: 4;
y=-4: 0.2: 4;

figure(1); %表面图
[xx, yy] = meshgrid(x, y);
zz = sin(xx .* yy) .* xx .^ 2 .* yy .^ 3;
mesh(xx, yy, zz);
title('Surface Plot');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 16);

figure(2); %轮廓图
contour(xx, yy, zz, 10);
title('Contour Plot')
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);

figure(3);  %伪彩图
pcolor(xx, yy, zz);
title('Pesudo Color Plot')
```

```
colorbar('vert');
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);

figure(4); %着色表面图
surfl(xx, yy, zz);
title('Surface Plot');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 16);
```

B.4 问题四的代码

```
%% Exercise 4
clear, clc;
x = linspace(-1, 1, 10);
y = linspace(-2, 2, 10);
z = linspace(-3, 3, 10);
[xx, yy, zz] = meshgrid(x, y, z);
v = (xx.^2) .* (yy.^3) .* (zz.^4);
xi = [-0.5, -0.2, 0.2, 0.5];
yi = [-0.7 0.3 1.2];
zi = [-2.5, -1.5, 0.8, 2.4];

figure(1); % projection on X-axis
slice(xx, yy, zz, v, xi, [], []);
colorbar('horiz');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 14);

figure(2); % projection on Y-axis
slice(xx, yy, zz, v, [], yi, []);
```

```
colorbar('horiz');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 14);

figure(3); % projection on Z-axis
slice(xx, yy, zz, v, [], [], zi);
colorbar('horiz');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 14);

figure(4); % full projection
slice(xx, yy, zz, v, xi, yi, zi);
colorbar('horiz');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 14);

figure(5); % slice on z = x^2 - y^2
[xsurf, ysurf] = meshgrid(-1: 0.1: 1, -1: 0.1: 1);
zsurf = xsurf .^ 2 - ysurf .^ 2;
slice(xx, yy, zz, v, xsurf, ysurf, zsurf);
colorbar('horiz');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
set(gca, 'fontsize', 14);
```

B.5 问题五的代码

```
%% Exercise 5
clear, clc;
```

```
x = linspace(-2 * pi, 2 * pi, 50);
y = linspace(-2 * pi, 2 * pi, 50);
t = linspace(0, 10, 1000);
[xx, yy] = meshgrid(x, y);
for i = 1: size(t, 2)
    zz = 10 .* exp(1 - t(i)).* sin(xx) .* cos(yy);
    surf(xx, yy, zz);
    axis([-2 * pi, 2 * pi, -2 * pi, 2 * pi, -30, 30]);
    xlabel('x-axis');
    ylabel('y-axis');
    title("plot where t = " + num2str(t(i)));
    set(gca, 'fontsize', 14);
    drawnow;
end
```

B.6 问题六的代码

```
%% Exercise 6
[x, y] = meshgrid(-2: .25 :2, -2: .25: 2);
Dy = sin(x .* y);
contour(x, y, Dy);
hold on;
quiver(x, y, ones(size(Dy)), Dy);
[xreal, yreal] = ode45(@(x, y) sin(x .* y), [-2, 2.2], -1.2); % test
↪   solution 1
plot(xreal, yreal, 'black', 'linewidth', 2);
[xreal, yreal] = ode45(@(x, y) sin(x .* y), [-2, 2.2], 0); % test
↪   solution 2
plot(xreal, yreal, 'red', 'linewidth', 2);
[xreal, yreal] = ode45(@(x, y) sin(x .* y), [-2, 2.2], 1.8); % test
↪   solution 3
plot(xreal, yreal, 'magenta', 'linewidth', 2);
legend('contour', 'arrows', 'initial=-1.2', 'initial=0',
↪   'initial=1.8', 'Location','NorthEastOutside')
hold off;
axis square;
```

```
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 14);
```

B.7 问题七的代码

```
%% Exercise 7
clear, clc;
x0 = 1;
y0 = 1;
m = [0.5 0 0 0.5   0    0;
     0.5 0 0 0.5 0.5    0;
     0.5 0 0 0.5 0.25 0.5];
p = [0.3333 0.3333 0.3334];
plot(x0, y0, '.');
hold on;

for i = 1:1000
    prob = m(randi(3), :);
    plot(x0, y0, '.');
    hold on;
    xlabel('x-axis');
    ylabel('y-axis');
    title("Iteration " + num2str(i));
    set(gca, 'fontsize', 14);
    drawnow;
    x=dot([x0 y0], prob(1: 2)) + prob(5);
    y=dot([x0 y0], prob(3: 4)) + prob(6);
    x0 = x;
    y0 = y;
end
hold off;
```

B.8 问题八的代码

```
%% Exercise 8
clear, clc;
```

```
fimplicit(@(x, y) sin(x .* y) + x + y, [-5 5 -5 5], 'linewidth', 1);
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);
```

B.9 问题九的代码

```
%% Exercise 9
clear, clc;
load('data.mat')
xx(isnan(xx(1, :)), :)=[]; % remove x-axis nan
xx(isnan(xx(2, :)), :)=[]; % remove y-axis nan
x=xx(:, 1);
y=xx(:, 2);
scatter(x, y, 20);
xlabel('x-axis');
ylabel('y-axis');
set(gca, 'fontsize', 16);
```