# Assignment 1

## One

1. Please use the built-in `MATLAB` function (`dsolve`) to solve the $1^{st}$ order differential equation: $y'(x) = xy$.

   用eqn存储微分方程的表达式，直接用 **dsolve()** 函数得到：$y(x) = C1 \times e^{\frac{x^2}{2}}$

   exp：

```
syms y(x)
eqn = diff(y,x) == x*y;
res = dsolve(eqn)
```

## Two

2. Please solve the initial value problem, which is $y'(x) = xy$ with $y(1) = 1$.

   增加变量con，存储 $y(1) = 1$，代入 **dsove()** 函数，可以求得：$y(x) = e^{-\frac{1}{2}} \times e^{\frac{x^2}{2}}$

   即，$y(x) = e^{\frac{x^2-1}{2}}$

   exp：

```
syms y(x)
eqn = diff(y,x) == x*y;
con = y(1) == 1;
y = dsolve(eqn,con)
simy = simplify(y)
```

## Three

3. Please solve the $2^{nd}$ order differential equation: $y''(x) + 8y'(x) + 2y(x) = cos(x)$ with $y(0) = 0$ and $y'(0) = 1$.

   令 $y''(x) = Dy$，$y'(x) = D2y$，eqn存储二阶微分方程的表达式，

   con1和con2分别存储 $y(0) = 0$ 和 $y'(0) = 1$，代入 **dsove()** 函数，求得：

$$y(x) = \frac{1}{\sqrt{65}} sin(x + atan(\frac{1}{8})) - \frac{1}{1820}((14 - 53\sqrt{14})e^{-(\sqrt{14}+4)x} + (-14 + 53\sqrt{14})e^{(\sqrt{14}-4)x})$$

   exp：

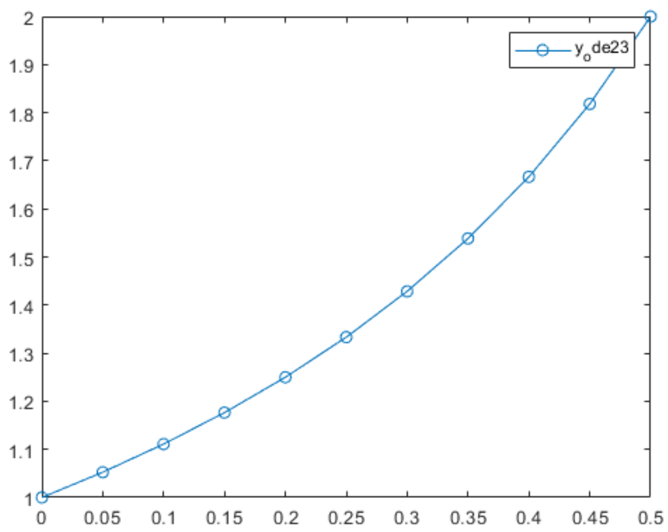
```
syms y(x)
Dy = diff(y,x);
D2y  = diff(Dy,x);
eqn = D2y + 8 * Dy + 2 * y == cos(x);
con1 = y(0) == 0;
con2 = Dy(0) == 1;
y = dsolve(eqn,con1,con2);
simy = simplify(y)
```

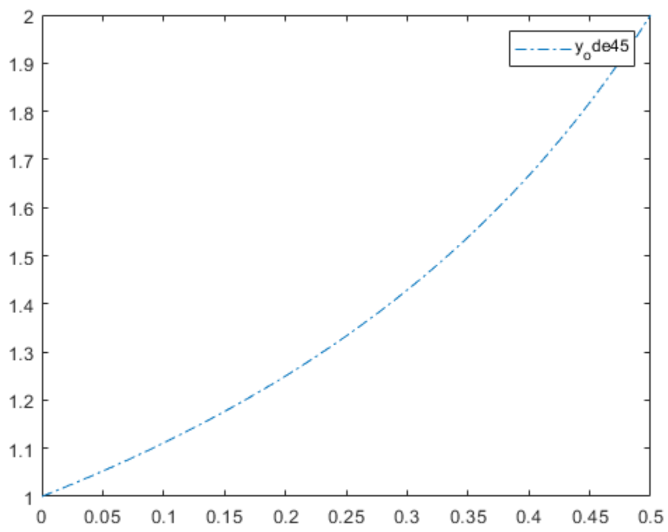
## Four

4. Please solve the numerical solution of the 1st ode: y′(x)=xy^2+y with y(0) = 1 and the x domain is [0, 0.5]. Try to use ode23 (https://www.mathworks.com/help/matlab/ref/ode23.html) and ode45 (https://www.mathworks.com/help/matlab/ref/ode45.html) respectively and compare the numerical results.

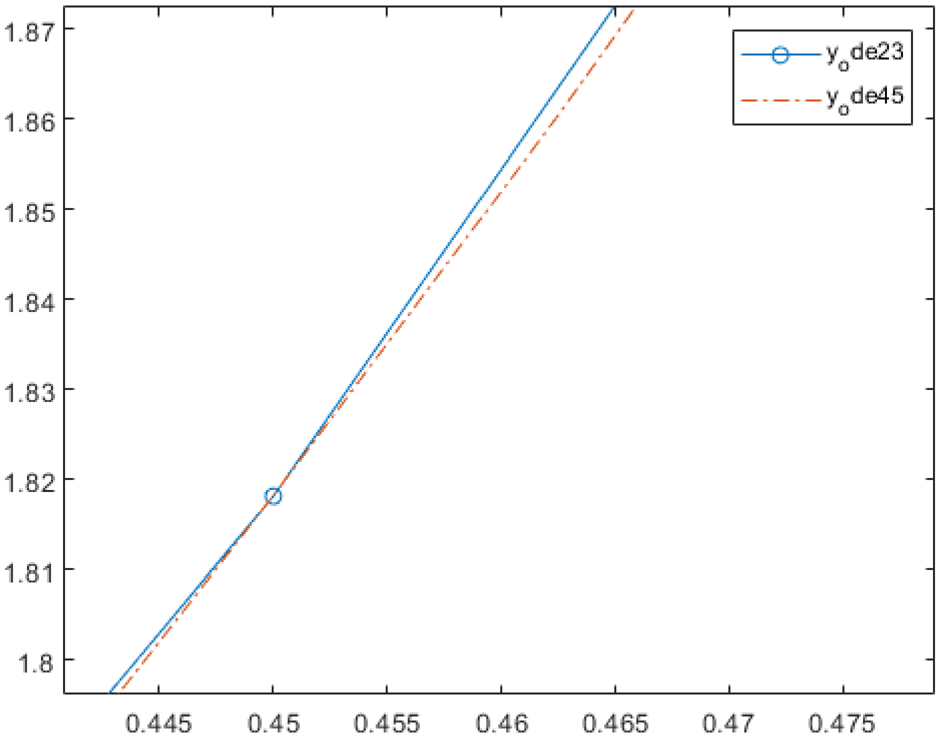
# ode23:

# ode45:




# Compare:




exp:

```
xdomain = [0 0.5];
y_0 = 1;
[x1,y1] = ode23(@(x1,y1) x1*y1^2+y1, xdomain, y_0);
[x2,y2] = ode45(@(x2,y2) x2*y2^2+y2, xdomain, y_0);
figure(1);
plot(x1,y1,'-o')
legend('y_ode23')
figure(2);
plot(x2,y2,'-.')
legend('y_ode45')
figure(3);
plot(x1,y1,'-o',x2,y2,'-.')
legend('y_ode23','y_ode45')
```

## Five

5. Solve the system of Lorenz equations (You may find this page very helpful with MATLAB/Python code: [https://en.wikipedia.org/wiki/Lorenz_system](https://en.wikipedia.org/wiki/Lorenz_system)). (1) Discuss the system behavior under the constant values: sigma, rho, and beta; (2) Comment the robustness of the dynamical system under different conditions.

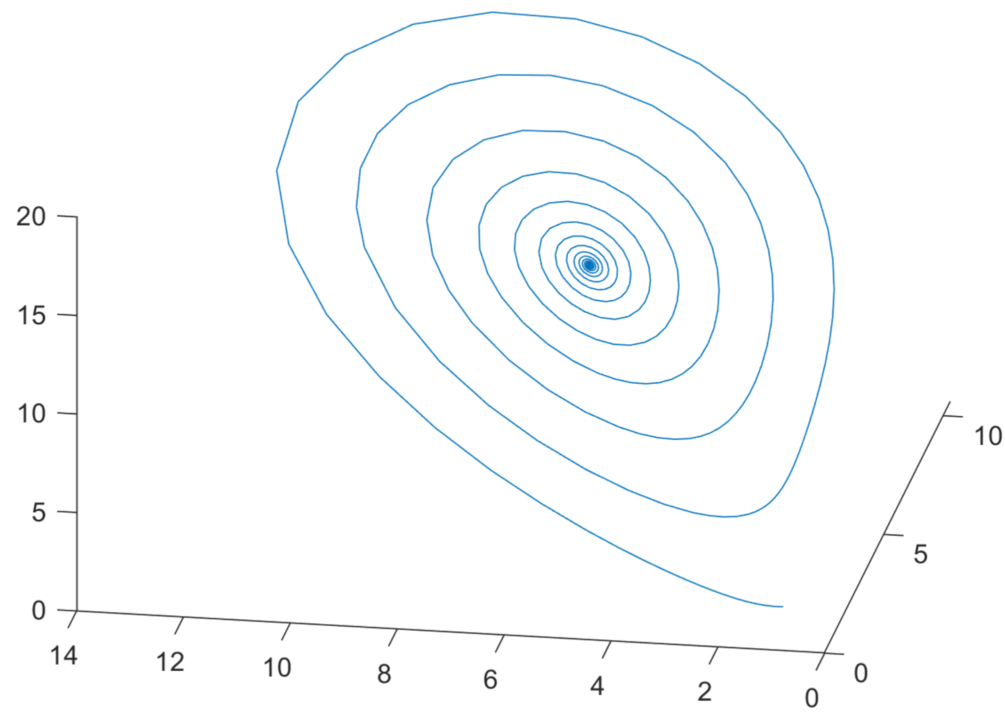
$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

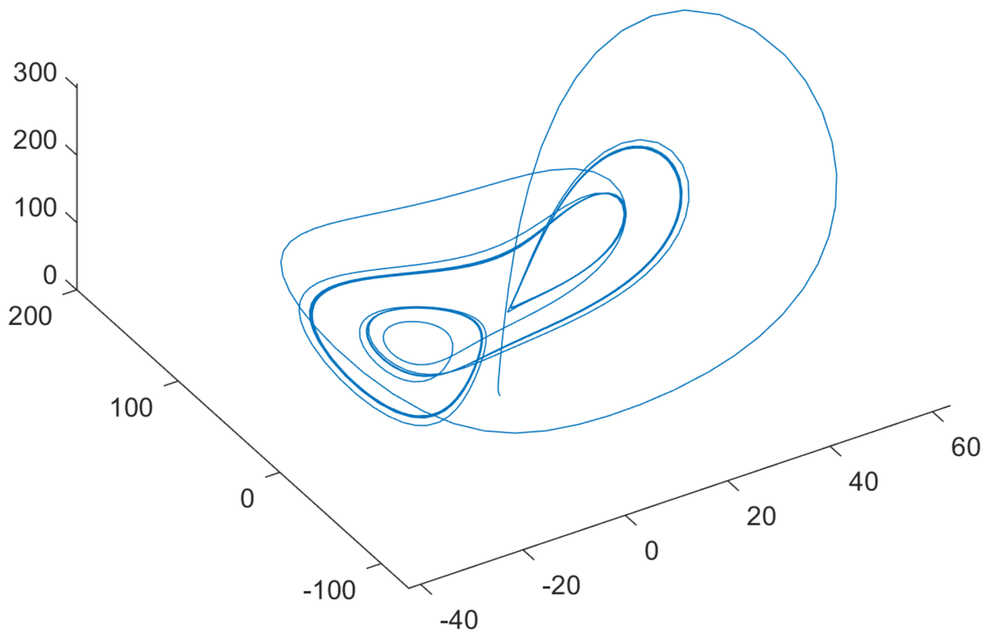
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

参数 σ 称为普兰特数，ρ 是规范化的瑞利数，β 和几何形状相关

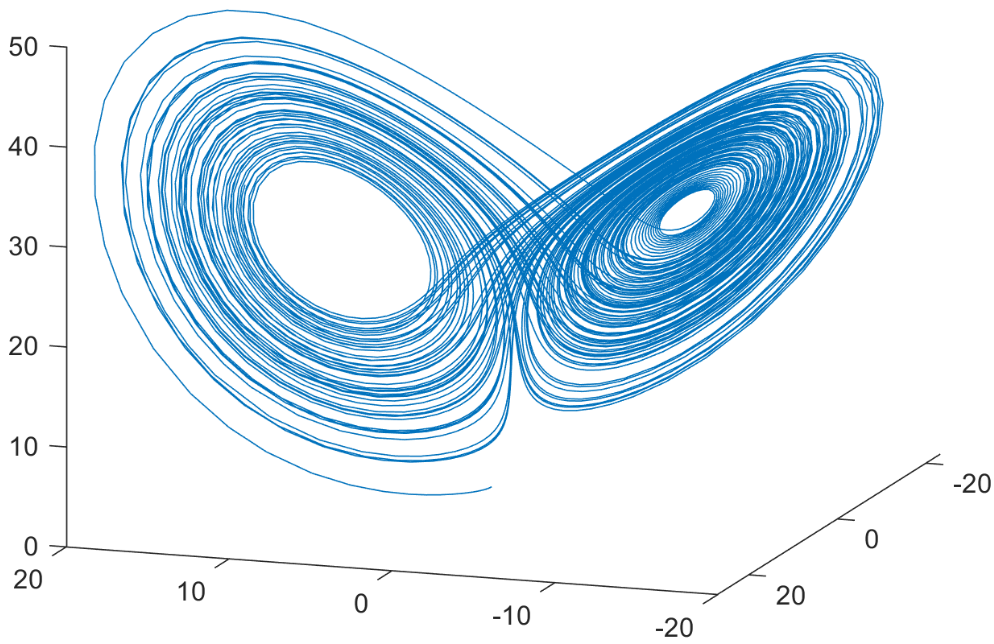
令 sigma = 10，beta = 8/3，rho = 13，获得图像：




令 sigma = 10，beta = 8/3，rho = 160，获得图像：




令 sigma = 10，beta = 8/3，rho = 28，获得图像：




exp:

```
sigma = 10;
beta = 8/3;
rho = 28;
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -
beta*a(3) + a(1)*a(2)];
[t,a] = ode45(f,[0 100],[1 1 1]);
plot3(a(:,1),a(:,2),a(:,3))
```

## Six

6. Consider Lotka-Volterra equations (https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations), which is known as predatory-prey equations: (1) Plot the phase portrait; (2) Compare the results using ode23 and ode45.

   Lotka-Volterra 方程是由两个一阶非线性 ODE 组成的方程组，用于描述生物系统中捕食者和猎物的种群。种群根据以下方程组随时间变化：

$$\begin{cases} \dfrac{dx}{dt} = \alpha x - \beta x y \\ \dfrac{dy}{dt} = \delta x y - \gamma y \end{cases}$$

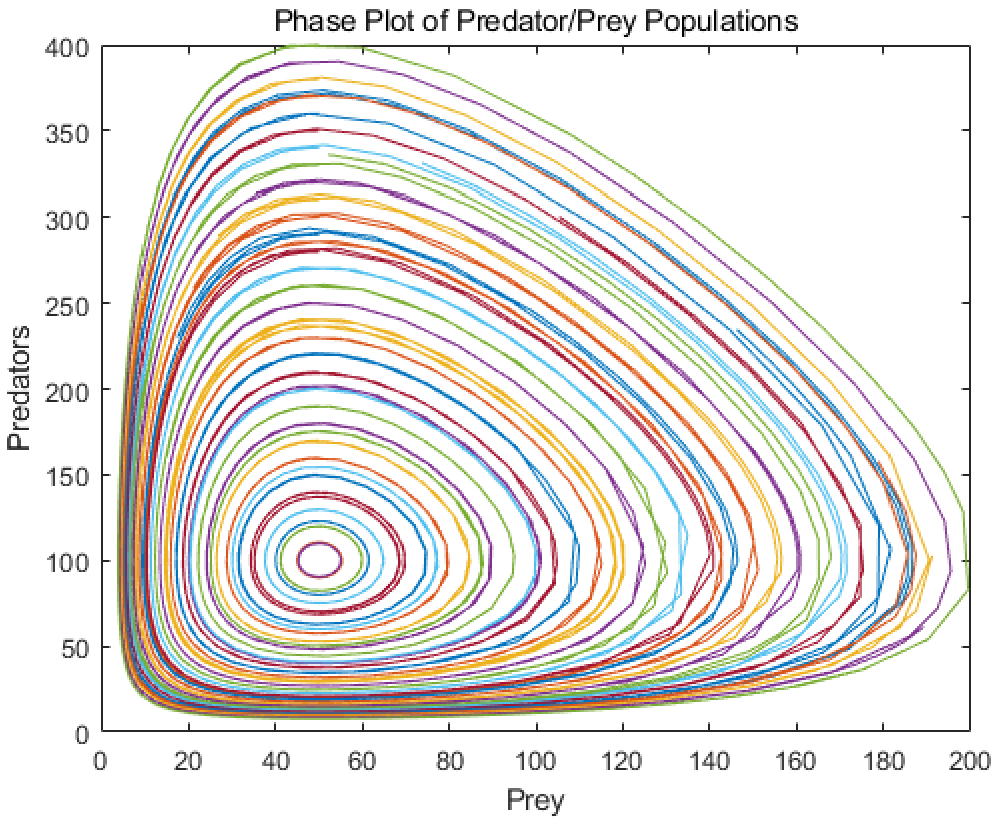
   其中，x 是猎物的种群大小，y 是捕食者的种群大小，t 是时间，α、β、δ 和 γ 是描述两个物种之间交互的常量参数，α 是自然增长率、β 是自然死亡率，γ 是猎物在单位时间内被猎食者捕获的比例

   令：$\alpha = \gamma = 1$、$\beta = 0.01$、$\delta = 0.02$，编写函数 Lotka.m 如下：

```
function dpdt = Lotka(t,p)

delta = 0.02;
beta = 0.01;
dpdt = [p(1) * (1 - beta*p(2));
        p(2) * (-1 + delta*p(1))];
end
```

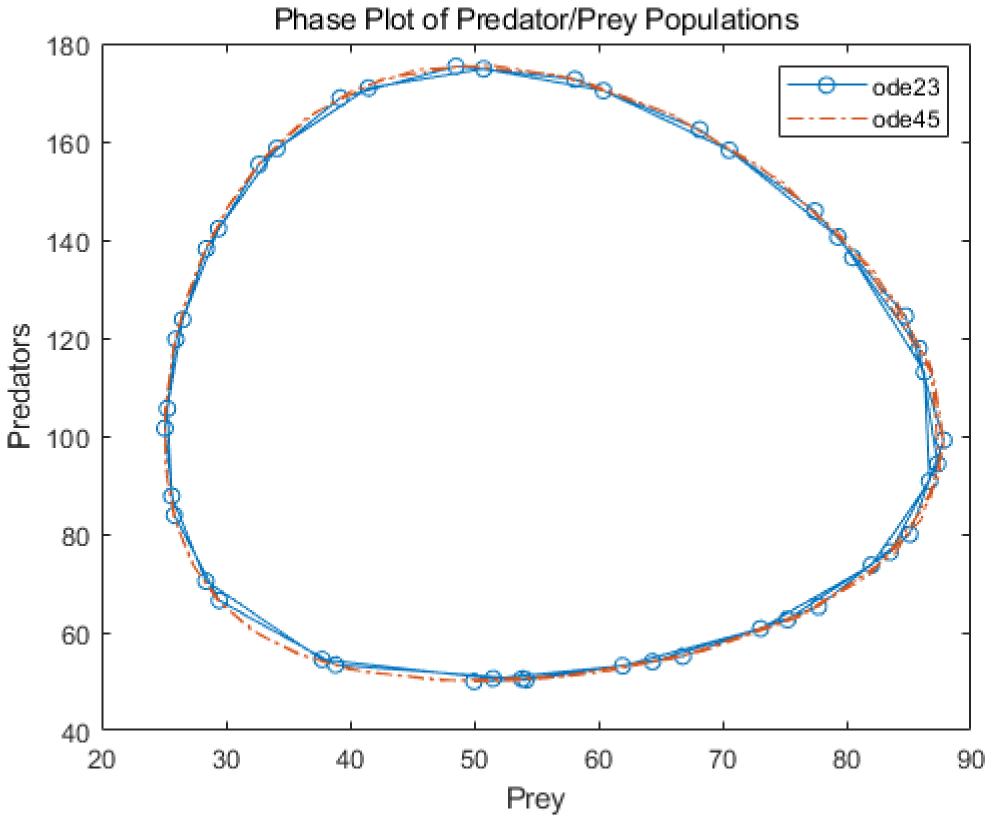
针对**不同**初始种群大小，假定猎物 x 的初始种群大小保持为50，改变捕食者 y 的初始种群大小，在区间 $[10:400]$ 变化，绘制相位图如下：



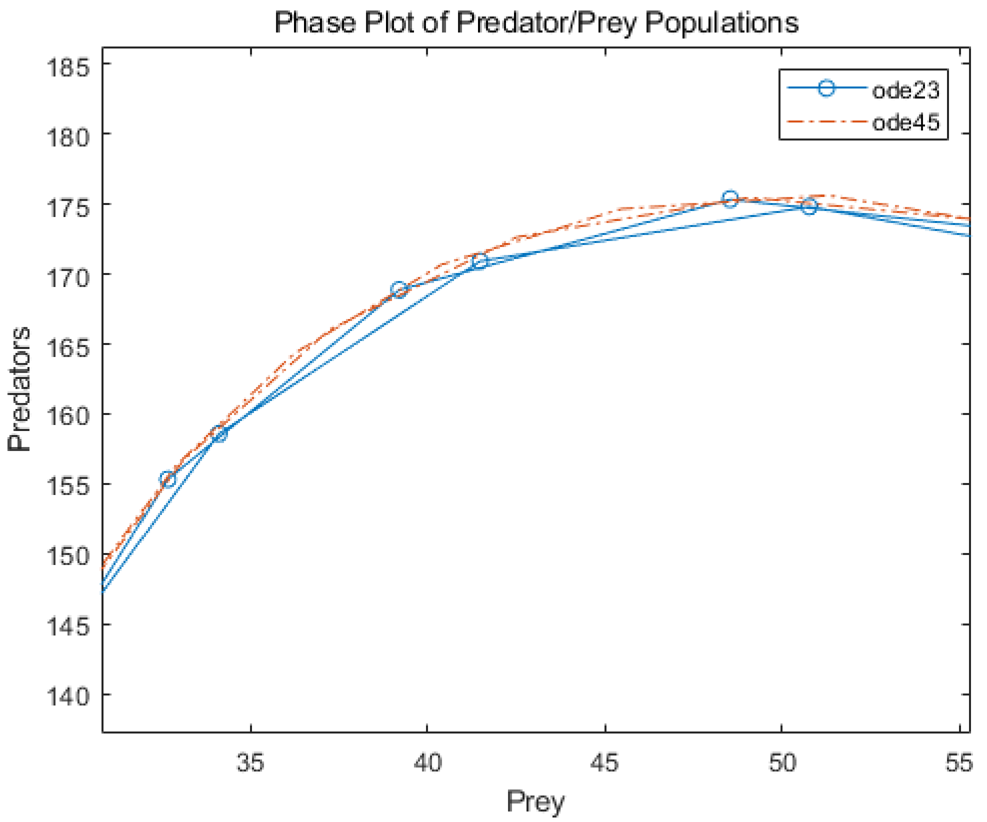

exp：

```
t0 = 0;
tfinal = 15;
y0 = 10:10:400;
for k = 1:length(y0)
    [t,p] = ode45(@Lotka,[t0 tfinal],[50 y0(k)]);
    plot(p(:,1),p(:,2))
    hold on
end
title('Phase Plot of Predator/Prey Populations')
xlabel('Prey')
ylabel('Predators')
hold off
```

取种群 x，y 的初值为 50，比较 ode23 和ode45 的结果，如下图：




放大查看，ode45的曲线更加圆滑




exp：

```
t0 = 0;
tfinal = 15;
p0 = [50; 50];
[t,p1] = ode23(@Lotka,[t0 tfinal],p0);
[t,p2] = ode45(@Lotka,[t0 tfinal],p0);
plot(p1(:,1),p1(:,2),'-o', p2(:,1),p2(:,2),'-.')
title('Phase Plot of Predator/Prey Populations')
legend('ode23','ode45')
xlabel('Prey')
ylabel('Predators')
```

## Seven

7. Consider Rossler attractor (https://en.wikipedia.org/wiki/R%C3%B6ssler_attractor) with the defining equations, use MATLAB to develop the code to solve the Rossler attractor problem.

   The defining equations of the Rössler system are（罗斯勒系统的定义方程是：）

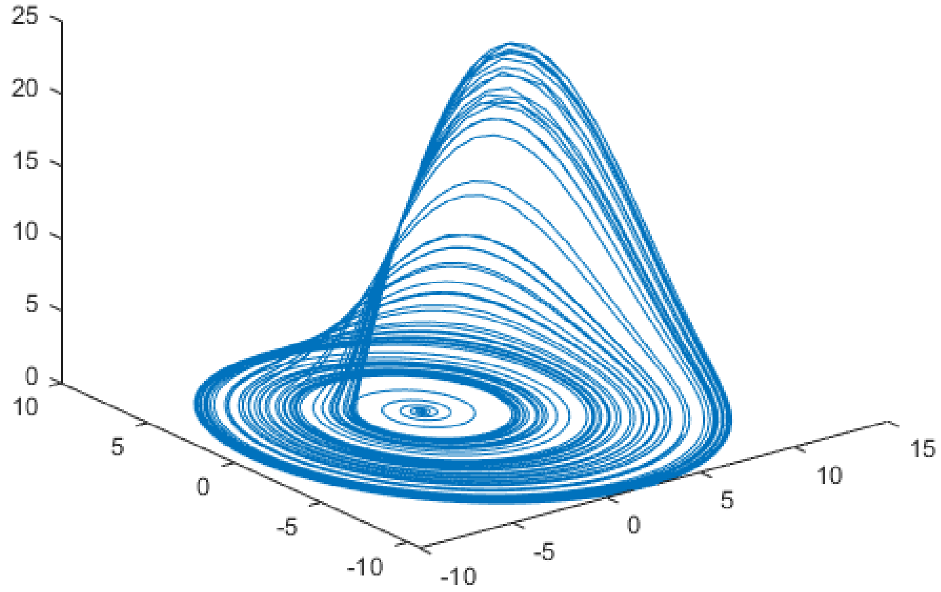
$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases}$$

   选择标准参数值，令 $a = 0.2, b = 0.2, c = 5.7$，编写函数 Rossler.m 如下：

```
function dpdt = Rossler(t,p)

a = 0.2;
b = 0.2;
c = 5.7;
dpdt = [-p(2) - p(3);
        p(1) + a * p(2);
        b + p(3) * (p(1) - c);];
end
```

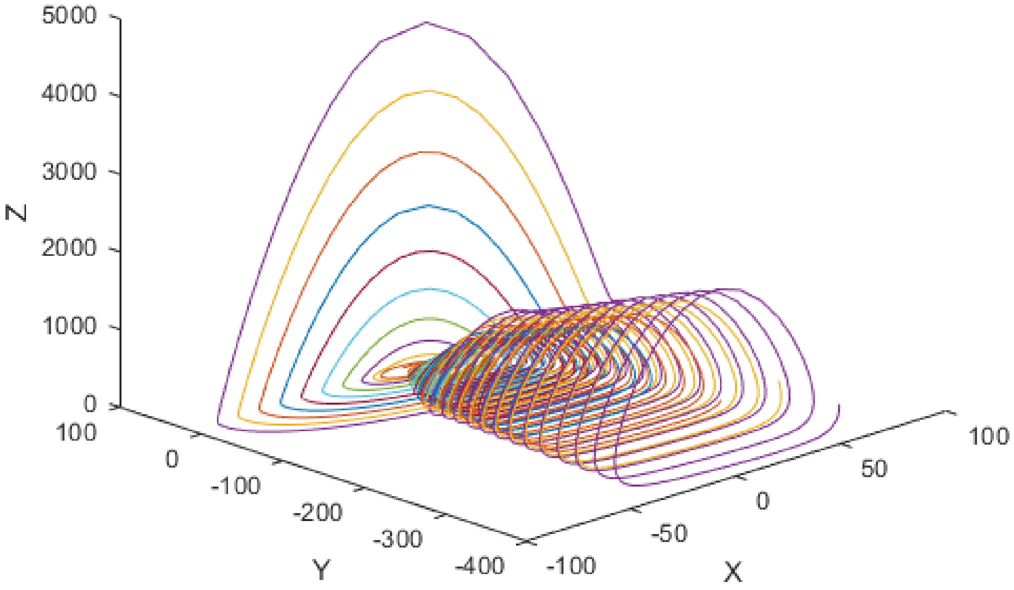
   - 令 x、y、z 三者的初值为 0，观察 t 在区间 $[0:400]$ 上的变化，得到图像如下：




exp:

```
t0 = 0;
p0 = [0 0 0];
tfinal = 400;
[t,p2] = ode45(@Rossler,[t0 tfinal],p0);
plot3(p2(:,1),p2(:,2),p2(:,3))
```

- 保持 y、z 的初值不变，t 则限定在区间 $[0:10]$ 上，变化 x 的初值，使其在区间 $[0:100]$ 上变化，每隔 10 记录下变化时得到的曲线，得到图像如下：

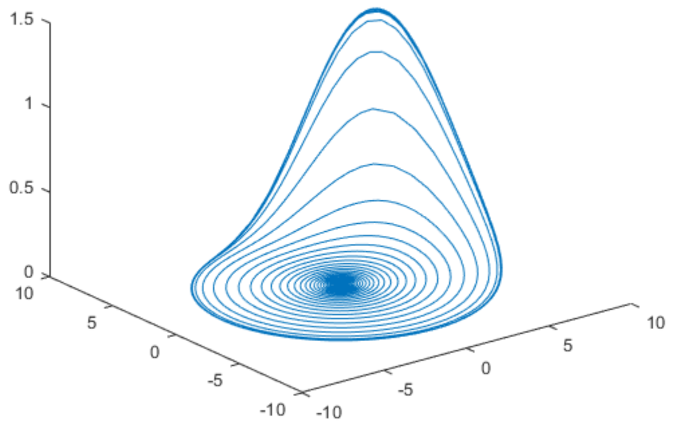


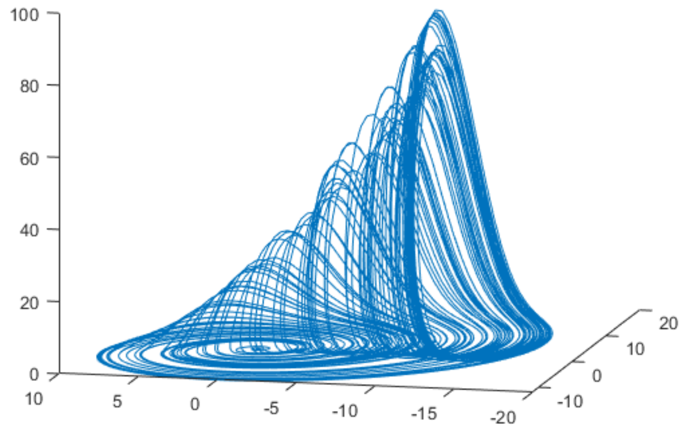

exp:

```
t0 = 0;
tfinal = 10;
x0 = 0:10:100;
figure(1);
for k = 1:length(x0)
    [t,p1] = ode45(@Rossler,[t0 tfinal],[x0(k) 0 0]);
    plot3(p1(:,1),p1(:,2),p1(:,3))
    hold on
end
xlabel('X')
ylabel('Y')
zlabel('Z')
hold off
```

- 保持常数 $b=0.2$、$c=5.7$，更改 a 的值为 3.8 和 0.05，观察到，$a=0.05$ 时，函数图像收敛到中心，$a=3.8$ 时，明显线条变得更加混乱：

  $a=0.05$:　　　　　　　　　　$a=3.8$:


1.5
1
0.5
0
10
5
0
-5
-10
-10
-5
0
5
10


100
80
60
40
20
0
10
5
0
-5
-10
-15
-20
-10
0
10
20