人口曲线拟合

此文档所在的github repo根目录下文件 uspop.txt 是美国从1790年至1990年的人口数据。请回答如下问题：

1. 请用 numpy.loadtxt 函数读入数据，将年份和对应的人口数分别存入 x 和 y 两个变量中。
2. 请用线性拟合函数 scipy.stats.linregress 拟合人口数随年份变化的曲线，拟合函数为 $y = kx + b$，其中为拟合参数。并将拟合结果及原始数据用 matplotlib.pyplot.plot 函数画在同一张图中。
3. 请用 scipy.optimize.curve_fit 函数拟合人口数随年份变化的曲线，拟合函数为 $y = ae^{cx}$，其中a, c为拟合参数。并将拟合结果及原始数据用 matplotlib.pyplot.plot 函数画在同一张图中。
4. 将上一小题中的曲线拟合问题通过公式变换转为线性拟合问题重新拟合。试比较这两种拟合方法的结果是否一致。

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

1. 得到x, y

In [ ]:
```
x, y = np.loadtxt("uspop.txt", dtype=int, unpack=True)
x, y
```

Out[ ]:
```
(array([1990, 1980, 1970, 1960, 1950, 1940, 1930, 1920, 1910, 1900, 1890,
        1880, 1870, 1860, 1850, 1840, 1830, 1820, 1810, 1800, 1790]),
 array([248709873, 226545805, 203211926, 179323175, 150697361, 131669275,
        122775046, 105710620,  91972266,  75994575,  62947714,  50155783,
         38558371,  31443321,  23191876,  17063353,  12860702,   9638453,
          7239881,   5308483,   3929214]))
```

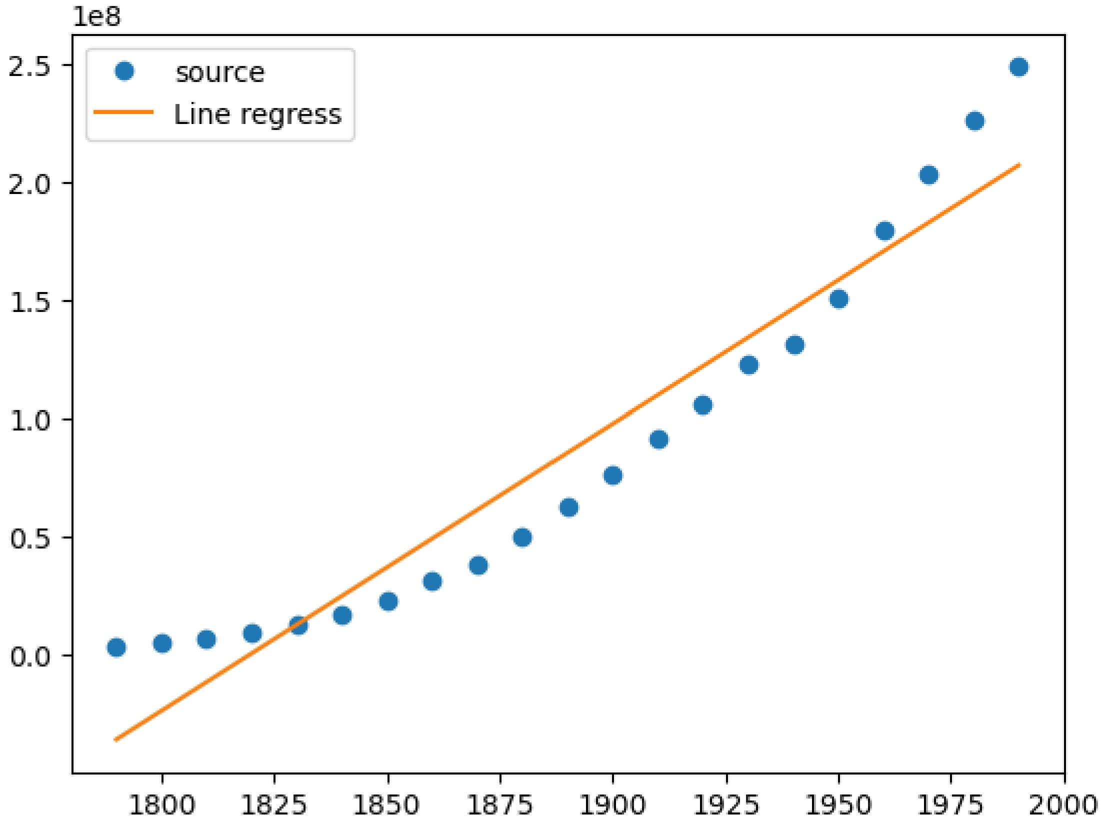
2. 线性拟合

In [ ]:
```
from scipy.stats import linregress

result = linregress(x, y)
result
```

Out[ ]:
```
LinregressResult(slope=1214073.0681818181, intercept=-2208933952.5303025, rvalu
e=0.9598400475216902, pvalue=6.095831503077399e-12, stderr=81409.68376049123, i
ntercept_stderr=153943250.88191068)
```

In [ ]:
```
x_lin = np.linspace(1790, 1990, 200)
plt.plot(x, y, 'o', label = 'source')
plt.plot(x_lin, result.intercept + result.slope * x_lin, '-', label = 'Line regr
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x25a48678750>
```



3. ae^{cx}拟合

考虑到e^x增长过快, 先对x进行线性变换, 防止overflow等错误

In [ ]:
```
def opt(x):
    return (x - 1790) / 50
x_temp = opt(x)
x_temp
```

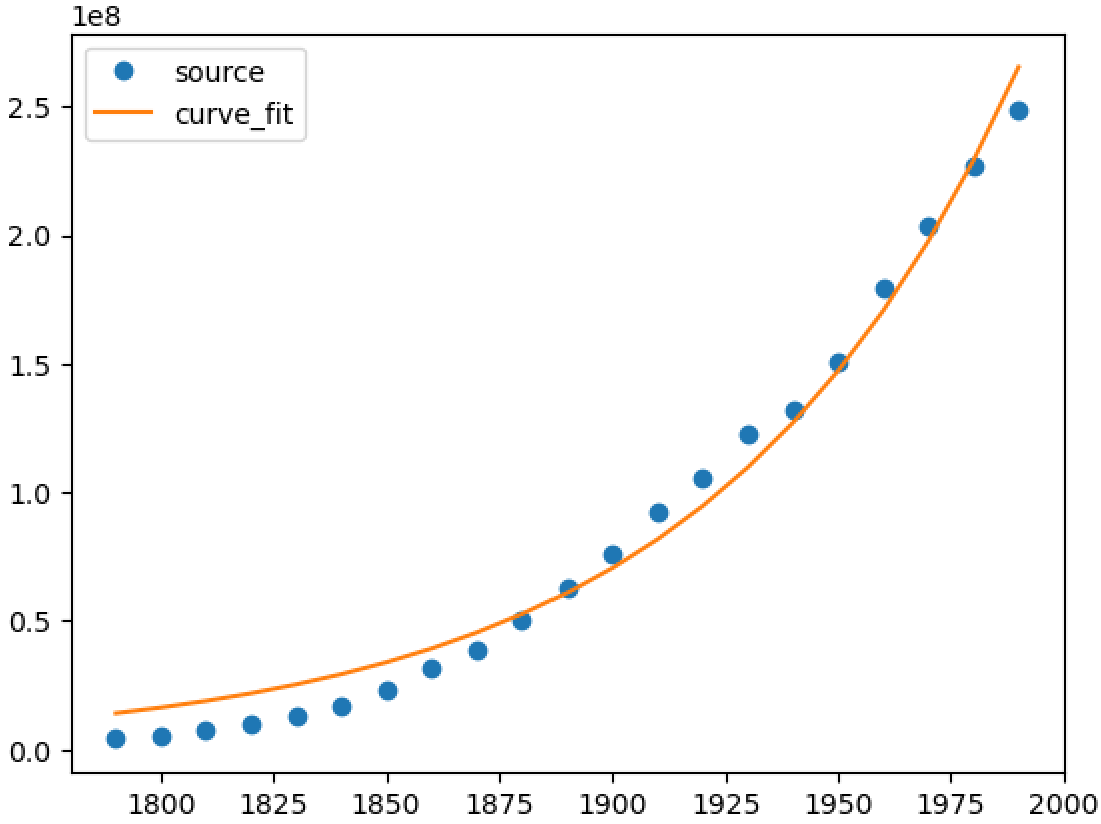
Out[ ]:
```
array([4. , 3.8, 3.6, 3.4, 3.2, 3. , 2.8, 2.6, 2.4, 2.2, 2. , 1.8, 1.6,
       1.4, 1.2, 1. , 0.8, 0.6, 0.4, 0.2, 0. ])
```

In [ ]:
```
from scipy.optimize import curve_fit

def f(x, a, c):
    return a * np.exp(c * x)

p0 = np.array([1, 1])
popt, pcov = curve_fit(f, x_temp, y, p0)
popt
```

Out[ ]:
```
array([1.39443865e+07, 7.36522038e-01])
```

In [ ]:
```
plt.plot(x, y, 'o', label = 'source')
plt.plot(x, f(opt(x), *popt), '-', label = 'curve_fit')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x25a48519810>
```

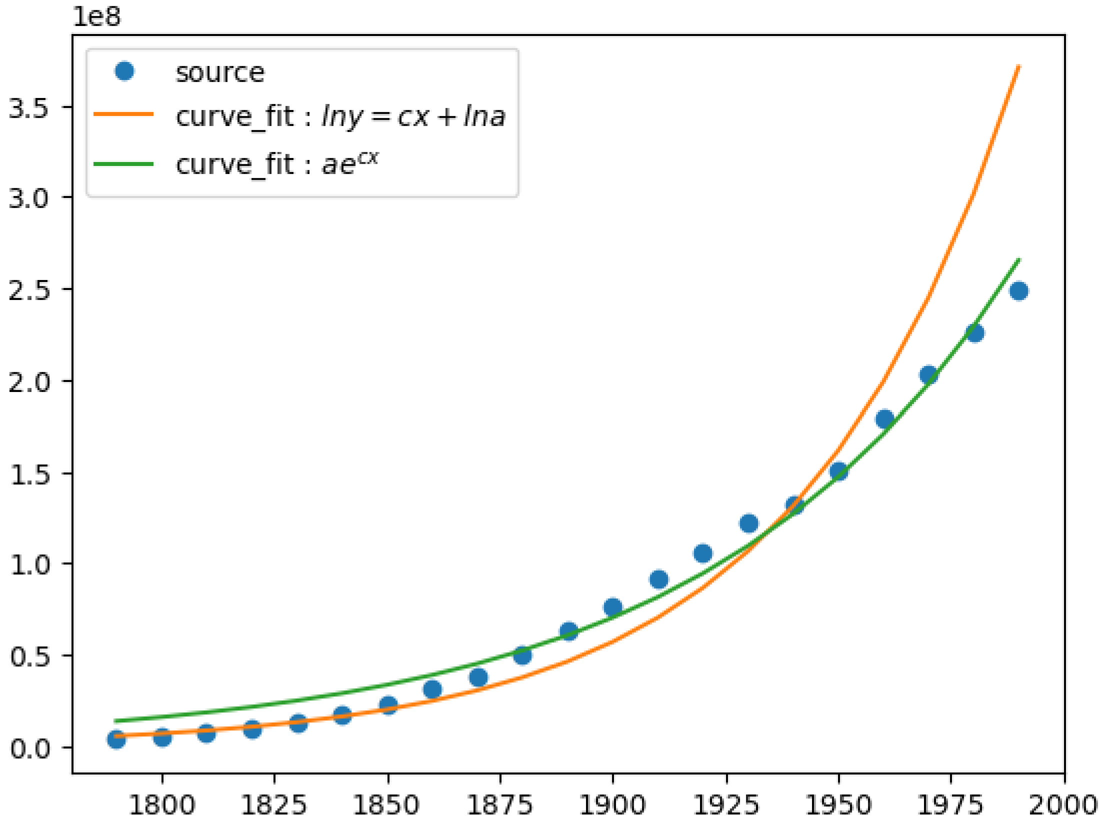


4. 变换比较

变换为$lny = cx + lna$

In [ ]:
```
lny = np.log(y)

result_change = linregress(x, lny)

plt.plot(x, y, 'o', label = 'source')
plt.plot(x, np.exp(result_change.intercept + result_change.slope * x), '-', labe
plt.plot(x, f(opt(x), *popt), '-', label = 'curve_fit : $ae^{cx}$')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x25a4994fe50>
```



从结果上来说，这两者的拟合是不一样的：

- 当y较小时，变换后拟合误差更小
- 当y较大时，原始ae^{cx}拟合误差更小

MATLAB 作图

请用 isosurface 函数渲染出:

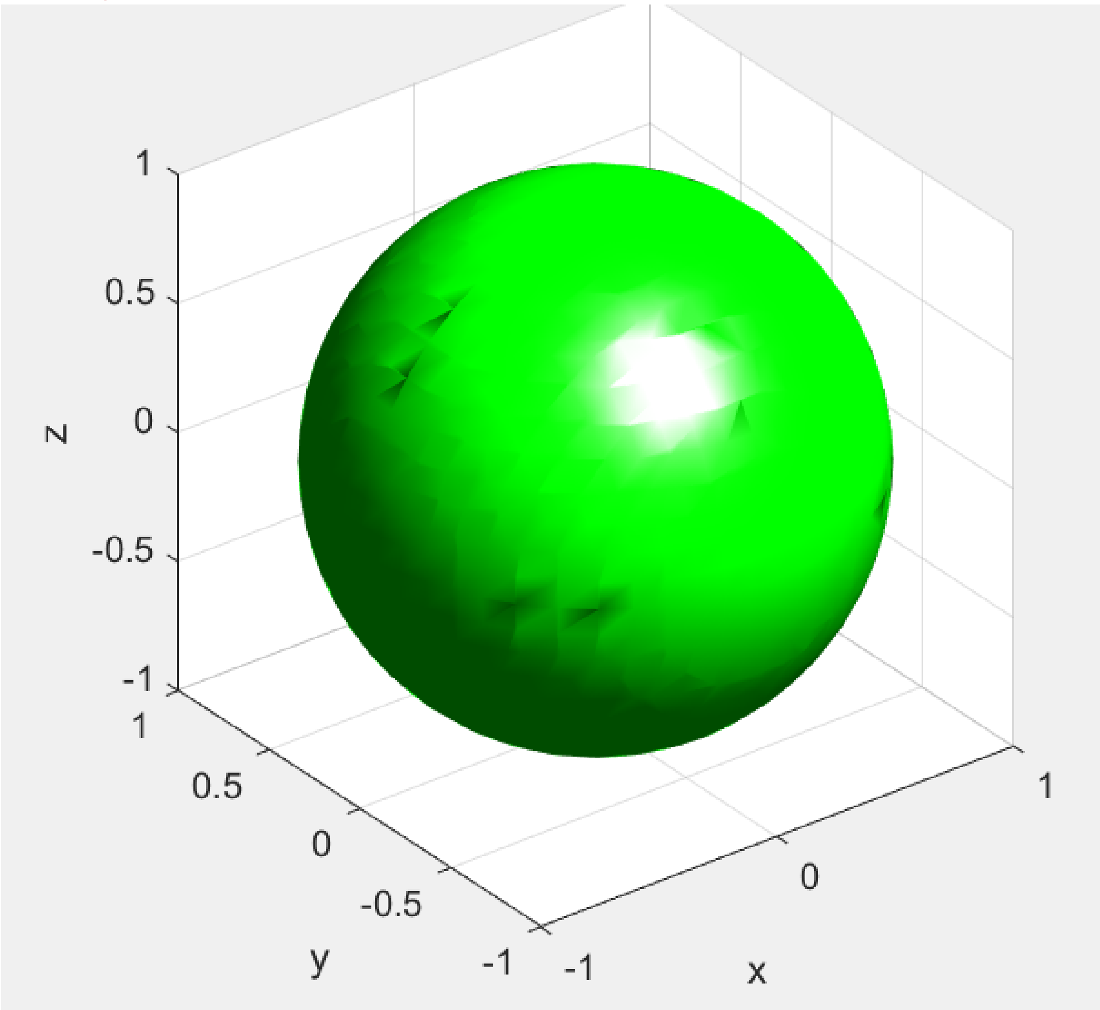
1. 一个半径为1的球面，球面为绿色。
2. 一个长轴为2，短轴为1的椭球面，其中椭球面由椭圆绕长轴旋转得到，椭球面为红色。

1. 一个半径为1的球面，球面为绿色。

代码如下：

```
[x, y, z] = meshgrid(-2:0.1:2);
V = x.^2 + y.^2 + z.^2;
s = isosurface(x, y, z, V, 1);
p = patch(s);
set(p, 'FaceColor', 'green');
set(p, 'EdgeColor', 'none');
view(3);
camlight;
lighting gouraud;
grid on;
xlabel('x');
```

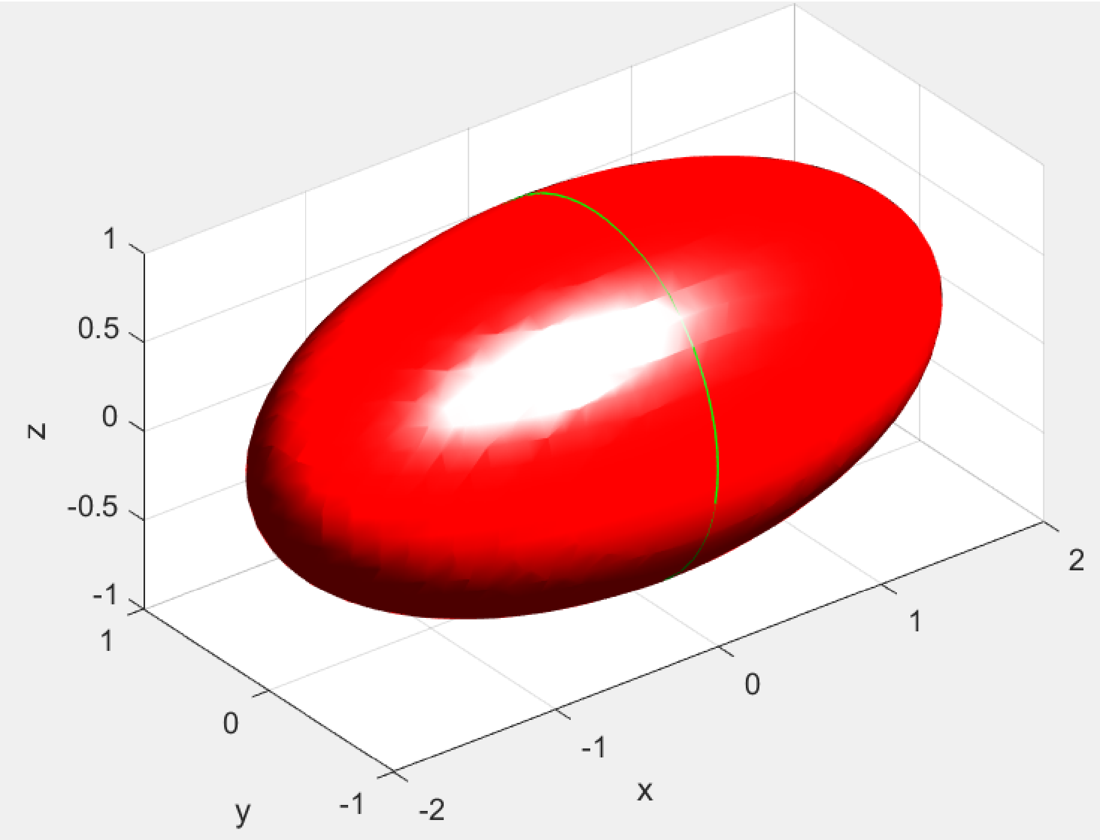
```
ylabel('y');
zlabel('z');
axis equal;
```



2. 一个长轴为2，短轴为1的椭球面，其中椭球面由椭圆绕长轴旋转得到，椭球面为红色。

代码如下：

```
a = 2;
b = 1;
[x, y, z] = meshgrid(-a:0.1:a, -b:0.1:b, -b:0.1:b);
V = (x.^2 / a^2) + (y.^2 / b^2) + (z.^2 / b^2);
s = isosurface(x, y, z, V, 1);
p = patch(s);
set(p, 'FaceColor', 'red');
set(p, 'EdgeColor', 'none');
view(3);
camlight;
lighting gouraud;
grid on;
axis equal
```



Mathematica 积分

使用mathematica进行以下积分：

$$\int \frac{\sin(x) - \sin(3x) + \sin(5x)}{\cos(x) + \cos(3x) + \cos(5x)} dx$$

在 Mathematica 中，我们可以使用 Integrate 或者积分符号进行积分，如下：

```
f1 = Sin[x] - Sin[3 x] + Sin[5 x]
     正弦      正弦        正弦
f2 = Cos[x] + Cos[3 x] + Cos[5 x]
     余弦      余弦        余弦
Integrate[f1 / f2, x]
积分

Out[2]= Sin[x] - Sin[3 x] + Sin[5 x]

Out[3]= Cos[x] + Cos[3 x] + Cos[5 x]

Out[4]= -Log[Cos[x]]

In[6]:= ∫(f1 / f2) dx

Out[6]= -Log[Cos[x]]
```

需要注意的是, mathematica 不定积分没有 $+C$:)

Latex 渲染

Lorenz Attractor

The Lorenz attractor is an attractor that arises in a simplified system of equations describing the two-dimensional flow of fluid. In the early 1960s, Lorenz accidentally discovered the chaotic behavior of this system when he found that, for a simplified system, periodic solutions of the form

$$\psi = \psi_0 sin\left(\frac{\pi a x}{H}\right) sin\left(\frac{\pi z}{H}\right)$$

$$\theta = \theta_0 sin\left(\frac{\pi a x}{H}\right) sin\left(\frac{\pi z}{H}\right)$$

grew for Rayleigh numbers larger than the critical value, $Ra > Ra_c$. Furthermore, vastly different results were obtained for very small changes in the initial values, representing one of the earliest discoveries of the so-called butterfly effect.

Lorenz obtained the simplified equations

$$\dot{X} = \sigma(Y - X)$$

$$\dot{Y} = X(\rho - Z) - Y$$

$$\dot{Z} = XY - \beta Z$$

now known as the Lorenz equations.