

```

'''
1.1.1
Проведем прямую  $y = mx + b$  через экспериментальные точки.
'''

import numpy as np
import matplotlib.pyplot as plt
x = np.array([0,1,2,3,])
y = np.array([-1, 0.2, 0.9, 2.1])

A = np.vstack([x, np.ones(len(x))]).T
print(A)
m,c = np.linalg.lstsq(A, y, rcond = None)[0]
print(m,c)

plt.plot(x,y, 'o', label = 'Исходные данные', markersize = 10)
plt.plot(x, m*x+c, 'r', label = 'Линейная экстраполяция')
plt.legend()
plt.savefig('1.1.1.1.png')

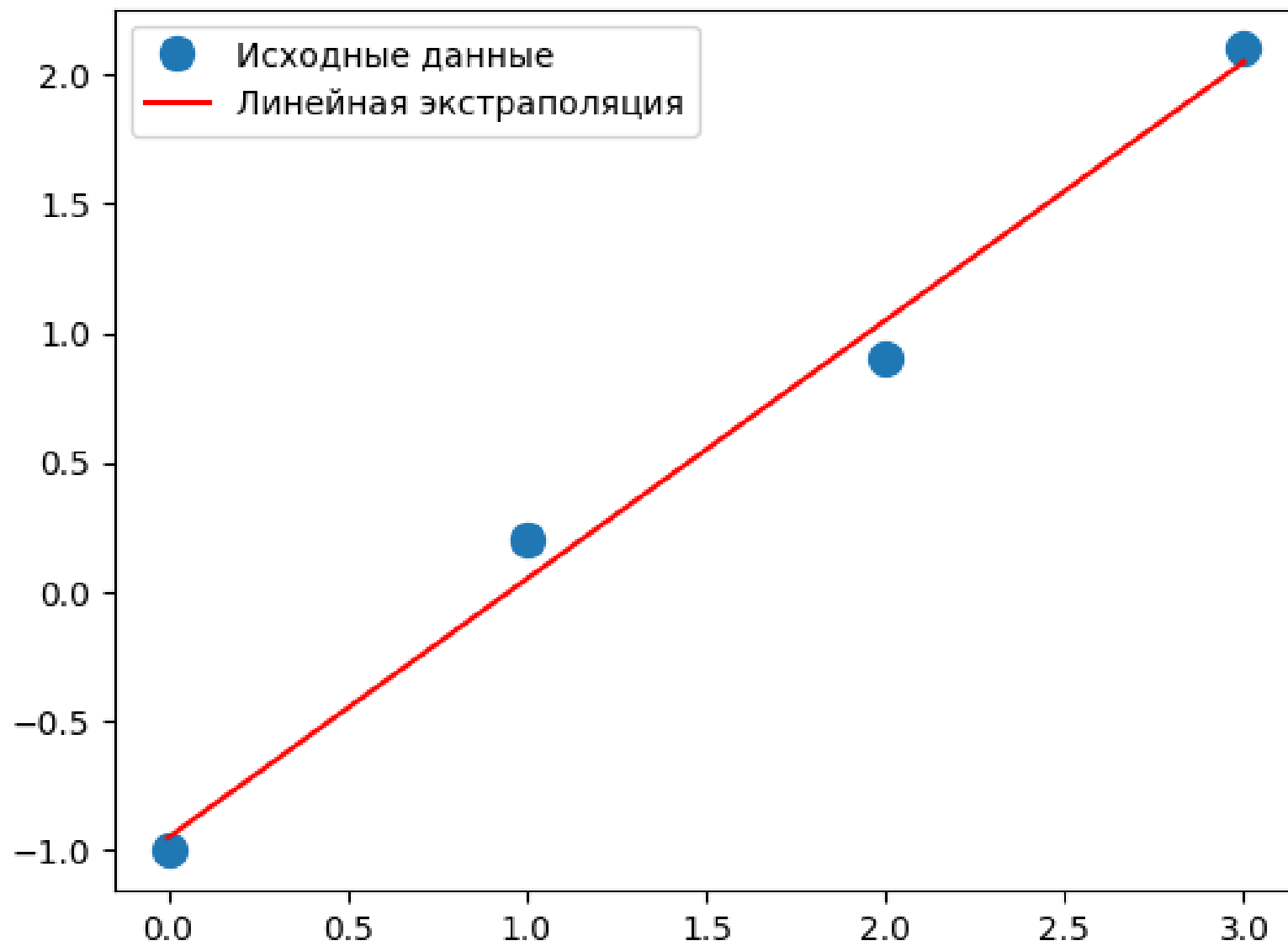
```

Program output:

```

[[0. 1.]
 [1. 1.]
 [2. 1.]
 [3. 1.]]
0.9999999999999999 -0.9499999999999994

```



```
'''
```

1.1.2.py

Пусть x , y – Вектора длиной $n > 3$ (точек > 3). Задача заключается в построении эстраполяционного полинома Второго порядка (параболы). Таким образом, необходимо найти такие коэффициенты полинома a , b , c по методу наименьших квадратов. Данные могут быть получены в результате измерений. Покажем пример генерации данных случайным образом и загрузки их из файла.

```
'''
```

```
import numpy as np
import matplotlib.pyplot as plt

delta = 1.0
x = np.linspace(-5,5,11)
y = x**2+delta*(np.random.rand(11)-0.5)
x+=delta*(np.random.rand(11)-0.5)

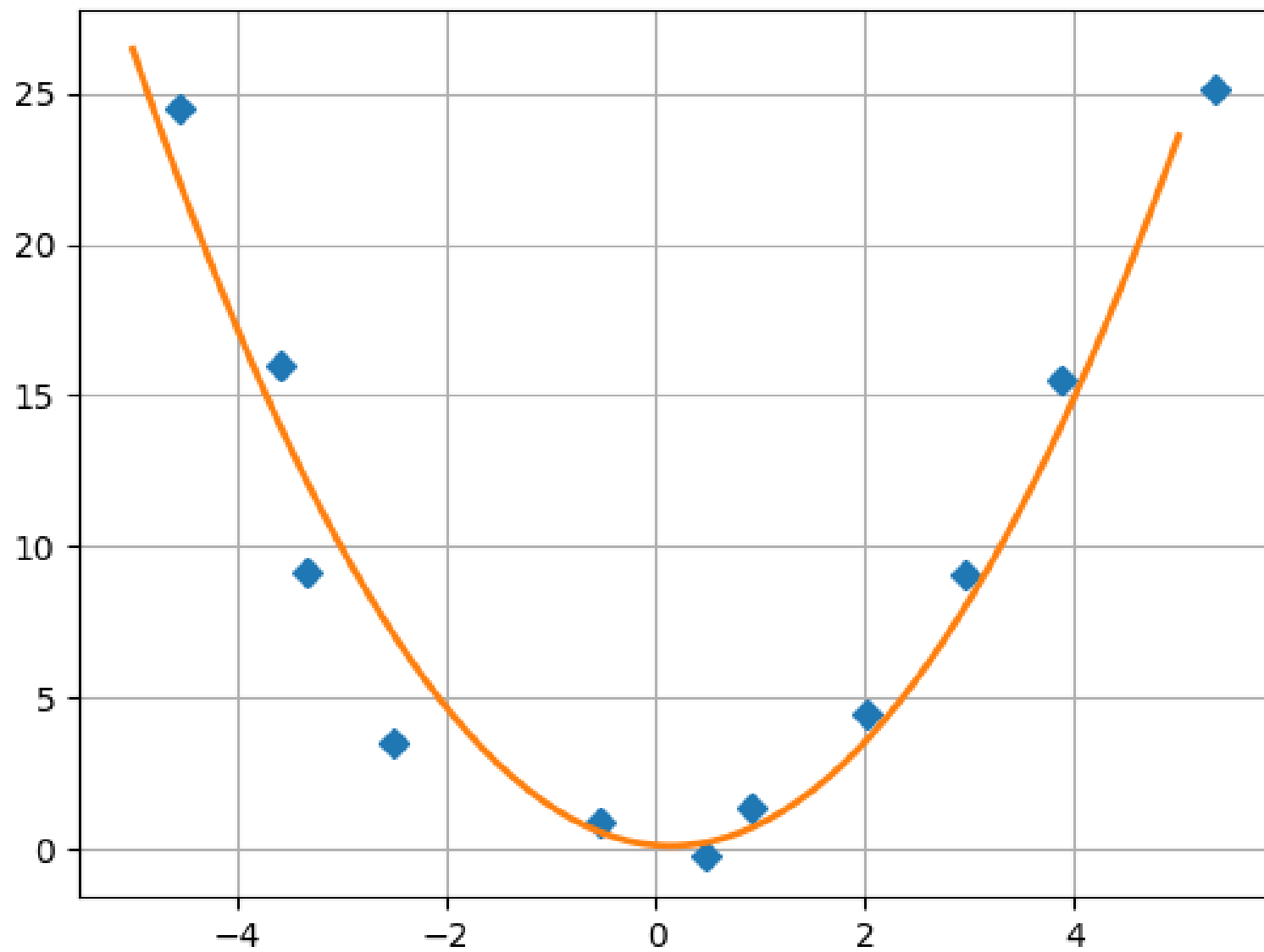
x.tofile('x_data.txt', '\n')
y.tofile('y_data.txt', '\n')
x = np.fromfile('x_data.txt', float, sep='\n')
y = np.fromfile('y_data.txt', float, sep='\n')

print(x)
print(y)

m = np.vstack((x**2, x, np.ones(11))).T
s = np.linalg.lstsq(m,y,rcond = None)[0]
x_prec = np.linspace(-5,5,101)
plt.plot(x,y,'D')
plt.plot(x_prec, s[0]* x_prec**2 + s[1] * x_prec+s[2], '-', lw=2)
plt.grid()
plt.savefig('1.1.2.1.png')
```

Program output:

```
[-4.54542168 -3.57754888 -3.33633459 -2.49921294 -0.52213739  0.46865606
  0.9066506   2.01472472  2.96579728  3.8797479   5.36031282]
[24.51670725 15.95804673  9.15735652  3.5509821   0.89417995 -0.25430938
  1.33719444  4.48754009  9.08489125 15.53791444 25.13739342]
```



```

'''
1.1.3
По данным предыдущего примера постройте эстраполяционный полинома третьего порядка
'''

from numpy import *
from numpy.random import *
import matplotlib.pyplot as plt

delta = 1.0
x = linspace(-5,5,11)
y = x**2+delta*(rand(11)-0.5)
x+=delta*(rand(11)-0.5)
x.tofile('x_data.txt', '\n')
y.tofile('y_data.txt', '\n')

x = fromfile('x_data.txt', float, sep='\n')
y = fromfile('y_data.txt', float, sep='\n')
print(x)
print(y)

m = vstack((x**3, x**2, x, ones(11))).T
s = linalg.lstsq(m,y,rcond = None)[0]
x_prec = linspace(-5,5,101)
plt.plot(x,y,'D')
plt.plot(x_prec, s[0]* x_prec**3 + s[1] * x_prec**2 + s[2]*x_prec+s[3], '-', lw=3)
plt.grid()
plt.savefig('1.1.3.1.png')

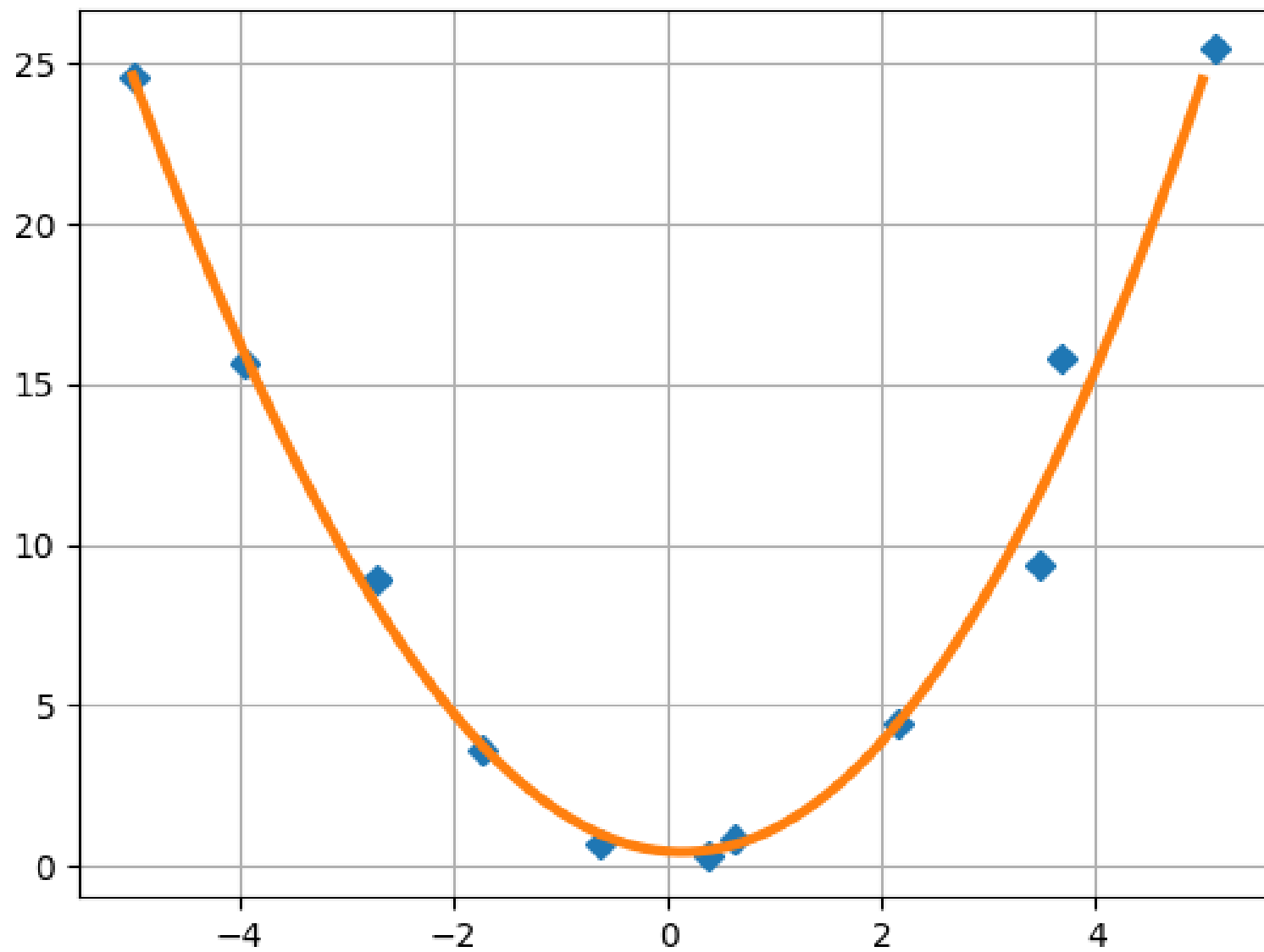
```

Program output:

```

[-4.99439651 -3.95423429 -2.72813604 -1.74310433 -0.62090463  0.38614772
  0.63247361  2.15805144  3.48542739  3.68756574  5.1287641 ]
[24.54599829 15.63335906  8.90582034  3.59667809  0.67560409  0.27525831
  0.8139668  4.41286438  9.39413194 15.83650485 25.44250386]

```



```
'''
```

1.1.3 Задание:

Представьте собственные данные и постройте экстраполяцию полиномами первой, второй и третьей степени.

```
'''
```

```
from numpy import *
from numpy.random import *
import matplotlib.pyplot as plt

delta = 1.0
x = linspace(-5,5,11)
y = x**2+delta*(rand(11)-0.5)
x+=delta*(rand(11)-0.5)

x.tofile('x_data.txt', '\n')
y.tofile('y_data.txt', '\n')
x = fromfile('x_data.txt', float, sep='\n')
y = fromfile('y_data.txt', float, sep='\n')
print(x)
print(y)
#2 шаг
m = vstack((x**2, x, ones(11))).T
s = linalg.lstsq(m,y,rcond = None)[0]
x_prec = linspace(-5,5,101)
plt.plot(x,y,'D')
plt.plot(x_prec, s[0]* x_prec**3 + s[1] * x_prec**2 + s[2], '-', lw=2)
plt.grid()
plt.savefig('1.1.3.2.3.png')
# 3 шаг
m = vstack((x**3, x**2, x, ones(11))).T
s = linalg.lstsq(m,y,rcond = None)[0]
x_prec = linspace(-5,5,101)
plt.plot(x,y,'D')
plt.plot(x_prec, s[0]* x_prec**3 + s[1] * x_prec**2 + s[2]*x_prec+s[3], '-', lw=3)
plt.grid()
plt.savefig('1.1.3.2.4.png')
```

Program output:

```
[-4.99501143 -3.65878695 -2.59544163 -2.21051295 -1.39816458 -0.22758887  
 0.84987088 1.70088242 2.91158271 4.17853465 5.26680981]  
[25.13137775 15.97934787 9.09241243 3.52609803 1.21172404 0.24499472  
 0.58589745 3.67290548 9.1398448 16.14437086 25.35622204]
```