Homework of Computational Physics (1st)

March 13, 2018

Wirtten by Liyanxing, 41524537, Huangkun Class

1 Question 1:

My source codes using Python3:

```
In [4]: """
        @brief: Lagrange 2 interpolation
        @param: x       the array of x
        @param: fx      the array of y
        @return: the 2nd power interpolation function
        """
        def get_sub_two_interpolation_func(x = [], fx = []):

            def sub_two_interpolation_func(Lx):
                result = 0
                a = (Lx-x[1])*(Lx-x[2])/(x[0]-x[1])/(x[0]-x[2])
                b = (Lx-x[0])*(Lx-x[2])/(x[1]-x[0])/(x[1]-x[2])
                c = (Lx-x[0])*(Lx-x[1])/(x[2]-x[0])/(x[2]-x[1])
                result = a*fx[0]+b*fx[1]+c*fx[2]
                return result

            return sub_two_interpolation_func

        print("The result:")

        #calculate f(0.472)
        x1 = [0.46,0.47,0.48]
        y1 = [0.4846555,0.4937452,0.5027498]
        interplt = get_sub_two_interpolation_func(x1,y1)
        print("f(0.472) = ", interplt(0.472))

        #calculate f(0.5)
        x2 = [0.47,0.48,0.49]
        y2 = [0.4937452,0.5027498,0.5116683]
        interplt = get_sub_two_interpolation_func(x2,y2)
```

```
        print("f(0.5) = ", interplt(0.5))
        #------------------------------------------------------
```

```
The result:
f(0.472) =  0.495552928
f(0.5) =  0.5205006999999999
```

2 Question 2:

According to the definition of Spline interpolation, we can get the functions as follow:

$$S_0 = 2x^3, S_1 = x^3 + ax_2 + bx + c$$

$$S_0^{'} = 6x^2, S_1^{'} = 3x^2 + 2ax + b$$

$$S_0^{''} = 12x, S_1^{''} = 6x + 2a$$

take $x = 1$ into the equations and we can get:

$$a = 3$$

$$b = -3$$

$$c = 1$$

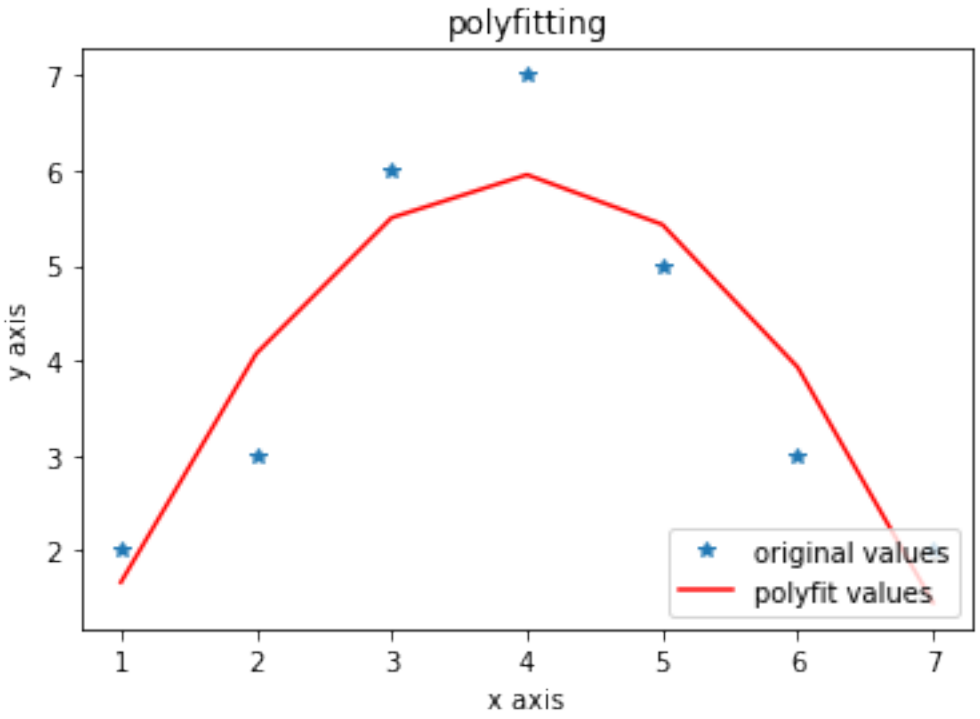
3 Question 3:

I use numpy as the tool to solve this problem. Here are my source codes using Python3:

```
In [3]: import matplotlib.pyplot as plt
        import numpy as np

        x = np.arange(1,8,1)
        y = np.array([2,3,6,7,5,3,2])

        z1 = np.polyfit(x, y, 2)
        # p1 is the fitting function
        p1 = np.poly1d(z1)
        curv = p1(x)

        # print(p1)
        # since the printed string is too ugly, I will type it myself latter in this page
        print("The figure of curve")

        #plot the figure of the curve
        plot1=plt.plot(x, y, '*',label='original values')
        plot2=plt.plot(x, curv, 'r',label='polyfit values')
        plt.xlabel('x axis')
        plt.ylabel('y axis')
```

```
plt.legend(loc=4)
plt.title('polyfitting')
plt.show()
#----------------------------------------------------
```

The figure of curve



The fitting function result: $p1 = -0.4881x^2 + 3.869x - 1.714$