HW2: Ch19.3 Lagrange Interpolation

Bailey Williams

For this assignment, we perform Lagrange Interpolation for two data sets.

Description of Method

Given n data points

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

the Lagrange helping functions

$$L_k = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

are used to define the Lagrange interpolating polynomial as follows:

$$P(x) = y_1 L_1(x) + y_2 L_2(x) + \ldots + y_n L_n(x)$$

The polynomial $P(x)$ *interpolates* the data points, meaning that it passes through all the data points. We will use `scipy` and its lagrange function to determine the Lagrange interpolating polynomial, as well as write Python program to use the formula above to evaluate $P(x)$ at a given point.

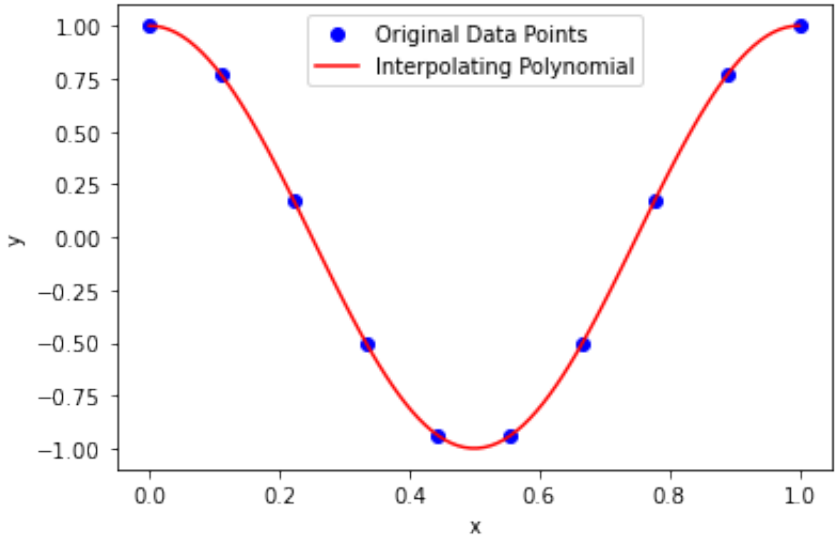
Problem 1

For this problem, we use $cos(2\pi x)$ on [0,1] to generate the data points. We use `scipy` to obtain the Lagrange interpolating polynomial and plot the results.

Python Program

The following Python program will implement the numerical method for this problem. The program automatically generates the graph of the $P(x)$ together with the original data points.

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange

x = np.linspace(0,1,10)
y = np.cos(2*np.pi*x)
f = lagrange(x, y)
t = np.linspace(0,1,100)

plt.plot(x,y,'bo',t,f(t),'r')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(('Original Data Points','Interpolating Polynomial'),loc=0)
plt.show()
```



Discussion of Results

The program generates 10 data points on the graph of $cos(2\pi x)$ and plots the data points in blue. The Lagrange interpolating polynomial $P(x)$ is generated as well, and is plotted in red. The graph of $P(x)$ passes through all of the data points in blue, and has the shape we would expect from the graph $cos(2\pi x)$ over [0,1].

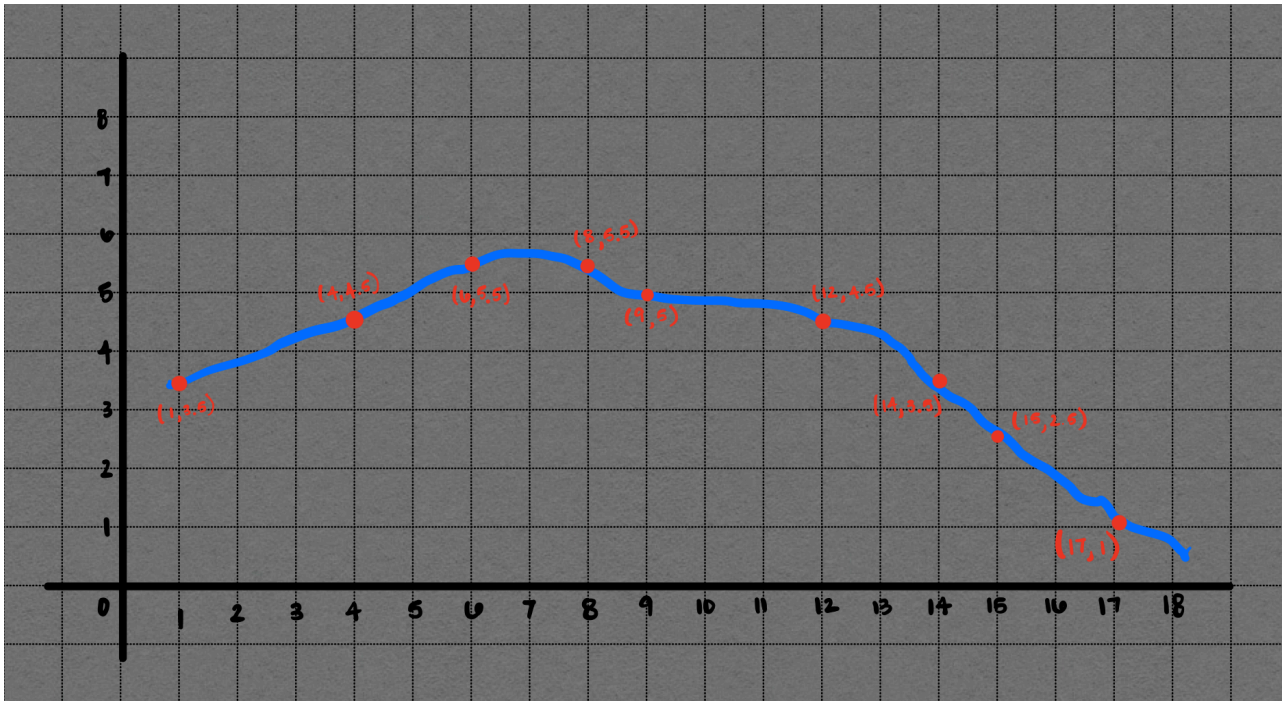
In [ ]:

In [ ]:

In [ ]:

Problem 2

For this problem, we use Python to generate the interpolating polynomial for our individual hand profile data. This data was obtained by tracing the profile of my hand onto graph paper; see figure below. We will write our ow code to compute the Lagrange polynomial interpolation at $x = 2.5$, and `scipy` to generate and plot the Lagrange interpolating polynomial.



Python Program

The following Python program will implement the numerical method for this problem. The program includes a definition for a function that is run using the command in the next subsection.

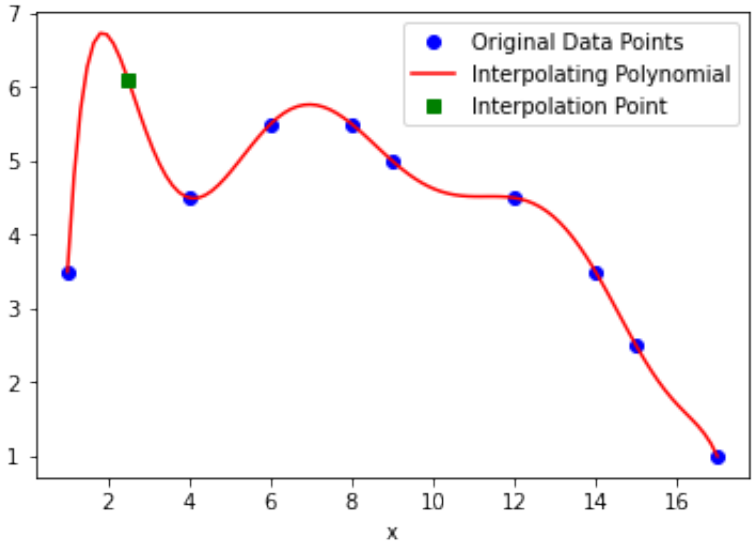
In [15]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange

#Enter Data
xdata = [1,4,6,8,9,12,14,15,17]
ydata = [3.5,4.5,5.5,5.5,5,4.5,3.5,2.5,1]

#Evaluate Lagrange Interpolating Polynomial at x = xp
def LagrangeInterp(xdata,ydata,xp):
    n = len(xdata)
    yp = 0
    for k in range(0,n):
        factor = 1
        for i in range(0,n):
            if i != k:
                factor = factor*(xp-x[i])/(x[k]-x[i])
        yp = yp + factor*y[k]

    #Displaying Output
    print('Interpolated value at %.4f is %.4f' % (xp,yp))

    #Use scipy to find interpolating polynomial for graphing
    f = lagrange(xdata, ydata)

    #Plot commands
    t = np.linspace(x[0],x[n-1],100)
    plt.plot(xdata,ydata,'bo',t,f(t),'r',xp,yp,'sg')
    plt.xlabel('x')
    plt.legend(('Original Data Points','Interpolating Polynomial','Interpolat
    plt.show()
```

Python Command and Output

The following command implements the Lagrange interpolation program for the hand data and $x = 2.5$.

In [16]:
```
LagrangeInterp(xdata,ydata,2.5)
```

```
Interpolated value at 2.5000 is 6.0911
```



Discussion of Results

The program plots the hand profile data points in blue together with the Lagrange interpolating polynomial $P(x)$ in red. The graph of $P(x)$ passes through all olf the data points in blue. The value of $P(2.5)$ is shown in green on the graph of $P(x)$. For the data shown here, the graph of $P(x)$ can be used to make accurate predictions for roughly $4 \leq x \leq 16$. Outside of that, the graph of $P(x)$ has large oscillations that would result in inaccurate predictions.

In [ ]: