21K-4827

Code to read data from a CSV file

```
import pandas as pd
import numpy as np

# Read data from CSV file
df = pd.read_csv('traffic.csv')
print(df.head())
# Convert data to numpy arrays
x = df['Time'].values
y = df['No of vehicles'].values
```

```
   Time  No of vehicles
0     0             9.0
1     1             6.0
2     2             9.0
3     3             8.0
4     4            11.0
```

Function for getting Lagrange Polynmial

```
# Function to calculate Lagrange polynomial
def lagrange_poly(x, y):
    n = len(x)
    p = np.poly1d(0.0)
    for i in range(n):
        L = np.poly1d(y[i])
        for j in range(n):
            if j != i:
                L *= np.poly1d([1.0, - x[j]]) / (x[i] - x[j])
        p += L
    return p

# Calculate Lagrange polynomial
p = lagrange_poly(x[1:8], y[1:8])
print(p)
```

```
           6         5         4         3         2
-0.05694 x + 1.396 x - 13.55 x + 66.02 x - 167.9 x + 208.1 x - 88
```

For Interpolating at a specific point

```
# Interpolate at a specific point
point = float(input("Enter x-coordinate to interpolate: "))
interp_value = p(point)

# Print Lagrange polynomial and interpolated value
print("Lagrange polynomial is:")
print(p)
print("Interpolated value at x =", point, "is:", interp_value)
```

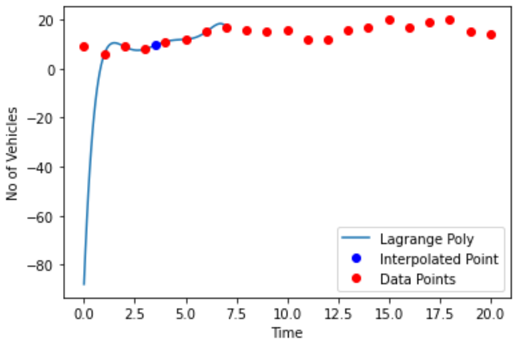
```
Enter x-coordinate to interpolate: 3.5
Lagrange polynomial is:
           6         5         4         3         2
-0.05694 x + 1.396 x - 13.55 x + 66.02 x - 167.9 x + 208.1 x - 88
Interpolated value at x = 3.5 is: 9.5283203125001
```

Plotting of Lagrange Polynomial

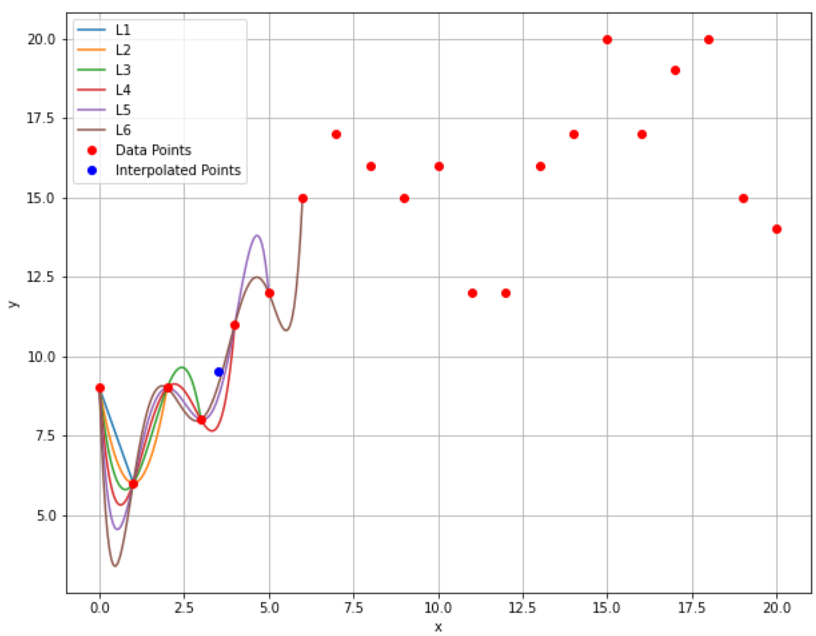
```
import matplotlib.pyplot as plt
xi=3.5
yi=9.5283203125001
p = lagrange_poly(x[1:8], y[1:8])
print(p)
xp=np.linspace(0,x[7],100)
yp=p(xp)

plt.plot(xp, yp, label='Lagrange Poly')
```

```
plt.plot(xi, yi, 'bo', label='Interpolated Point')
plt.plot(x, y, 'ro', label='Data Points')
plt.xlabel('Time')
plt.ylabel('No of Vehicles')
plt.legend()
plt.show()
```

```
           6         5         4         3         2
-0.05694 x + 1.396 x - 13.55 x + 66.02 x - 167.9 x + 208.1 x - 88
```




```
fig = plt.figure(figsize = (10,8))

n=6
for i in range(1,n+1,1):
  p = lagrange_poly(x[0:i+1], y[0:i+1])
  xp=np.linspace(0,x[i],100)
  yp=p(xp)
  plt.plot(xp, yp, label = f"L{i}")
plt.plot(x,y,'ro',label="Data Points")
plt.plot(xi,yi,'bo',label="Interpolated Points")
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid()
plt.show()
```

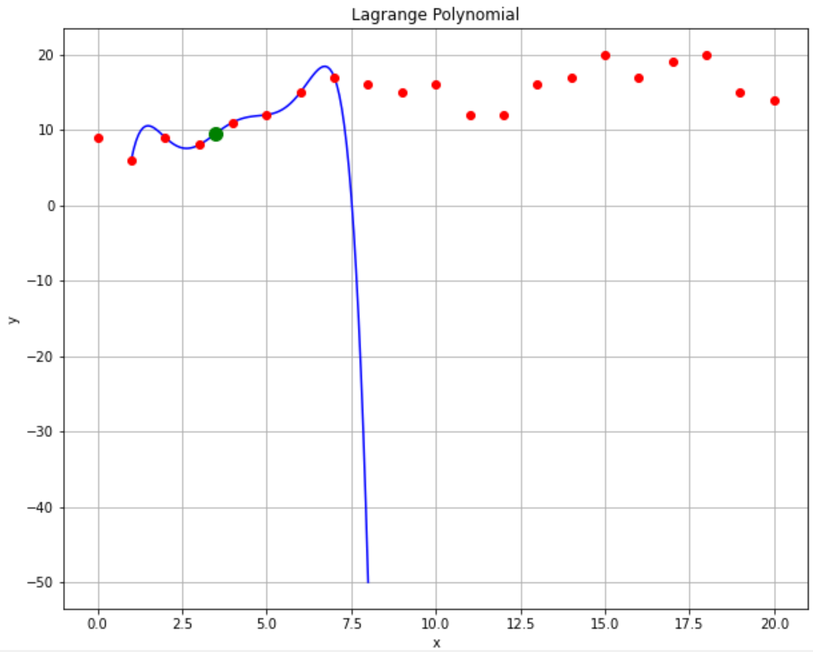


Scipy Implimentation of Lagrange Polynomial

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange

# Define the Lagrange Polynomial
f = lagrange(x[1:8], y[1:8])
```

```
# Find P(50) by evaluating the polynomial at x=50
p_3_5 = f(3.5)
print("P(3.5) =", p_3_5)

# Print the polynomial coefficients
print("Lagrange Polynomial:", np.poly1d(f).coefficients)

# Plot the Lagrange Polynomial and the data points
x_new = np.linspace(1, x[8], 100)
fig = plt.figure(figsize = (10,8))
plt.plot(x_new, f(x_new), 'b', x, y, 'ro')
plt.plot(3.5, p_3_5, 'go', markersize=10)
plt.title('Lagrange Polynomial')
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

```
P(3.5) = 9.5283203125001
Lagrange Polynomial: [-5.69444444e-02  1.39583333e+00 -1.35486111e+01  6.60208333e
 -1.67894444e+02  2.08083333e+02 -8.80000000e+01]
```

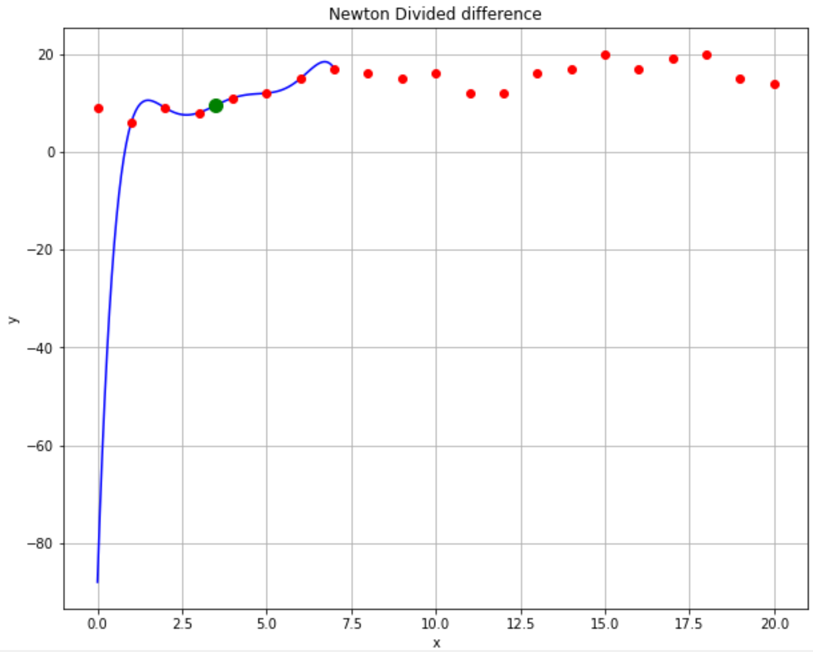


Code for Newton divided difference Method

```
import numpy as np
def divided_difference_table(x, y):
    n = len(x)
    F = [[0] * n for i in range(n)]
    for i in range(n):
        F[i][0] = y[i]
    for j in range(1, n):
        for i in range(j, n):
            F[i][j] = (F[i][j-1] - F[i-1][j-1]) / (x[i] - x[i-j])
    return F
def newton_div_dif_poly(x,y,xi):
   F=divided_difference_table(x,y) # Saving divided difference in a variable F
   n=len(x)
   prod=np.poly1d(1)
   N=np.poly1d(F[0][0])
   for i in range(1,n):
     prod=np.poly1d(x[0:i],True)
     N+=np.poly1d(F[i][i]*(prod.c))


   return (N)

nw = newton_div_dif_poly(x[1:8], y[1:8],3.5)
print("Newton Divided Differences(poly1d):", np.poly1d(nw).coefficients)

x_new = np.linspace(0, x[7], 100)
```

```
fig = plt.figure(figsize = (10,8))
plt.plot(x_new, f(x_new), 'b', x, y, 'ro')
plt.plot(3.5, nw(3.5), 'go', markersize=10)
plt.title('Newton Divided difference')
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

```
Newton Divided Differences(poly1d): [-5.69444444e-02  1.39583333e+00 -1.35486111e+
 -1.67894444e+02  2.08083333e+02 -8.80000000e+01]
```



✓ 0s completed at 10:47 PM