Computational and Numerical Methods

Group 16

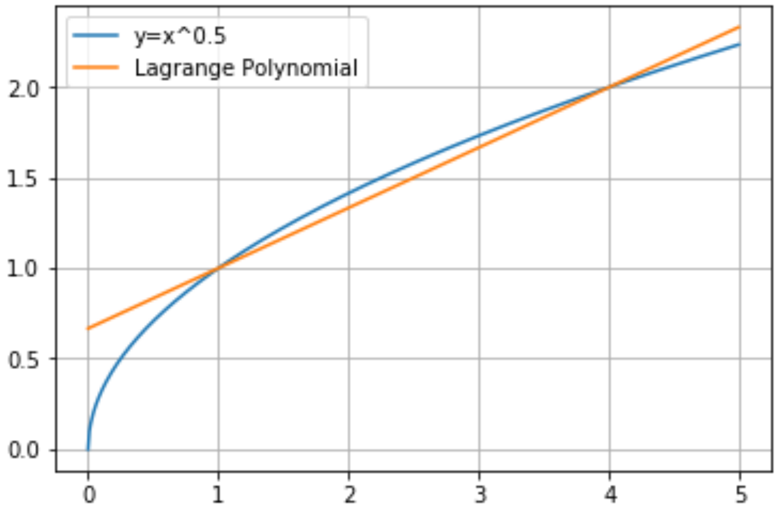
Set 6 (10-09-2018): Lagrange and Newton Interpolation

Vidhin Parmar 201601003

Parth Shah 201601086

Show Code

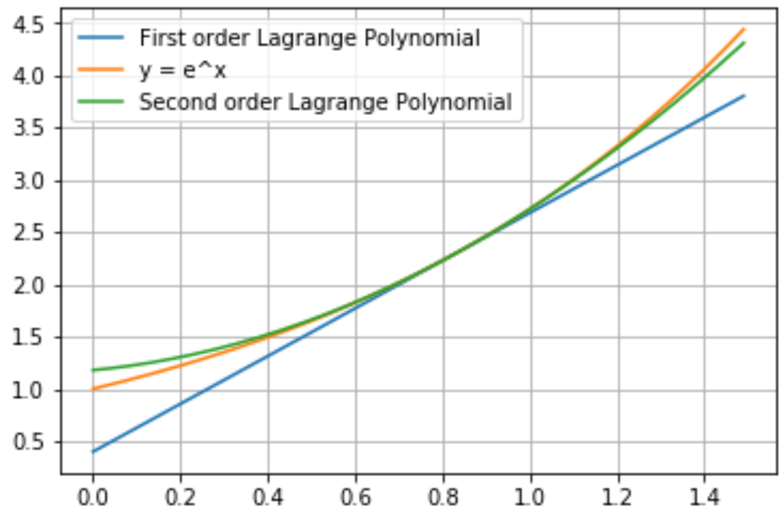
Q1: $\sqrt{x}$




```
The Lagrangian polynomial is:

0.3333 x + 0.6667
```

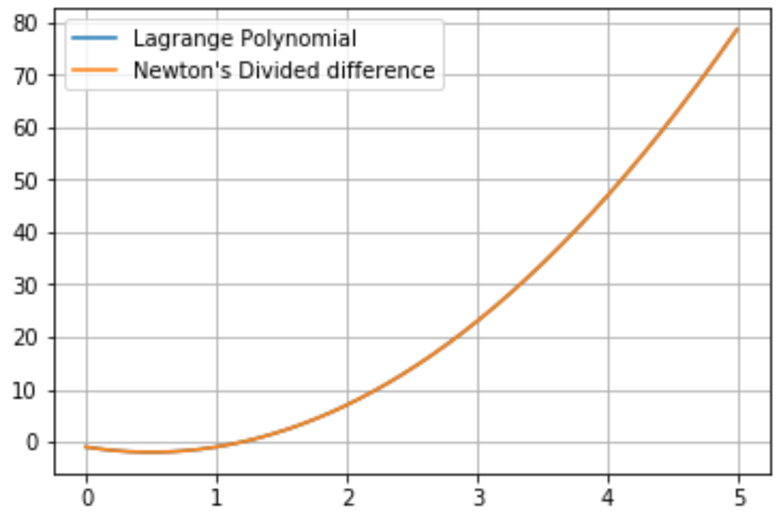
Q2: e^x

```
The Lagrangian polynomial is:

2.282 x + 0.3993
The Lagrangian polynomial is:
       2
1.145 x + 0.3926 x + 1.179
```



Q3 x = [0, 1, 2] y = [-1, -1, 7]

```
The Lagrangian polynomial is:
   2
4 x - 4 x - 1
```

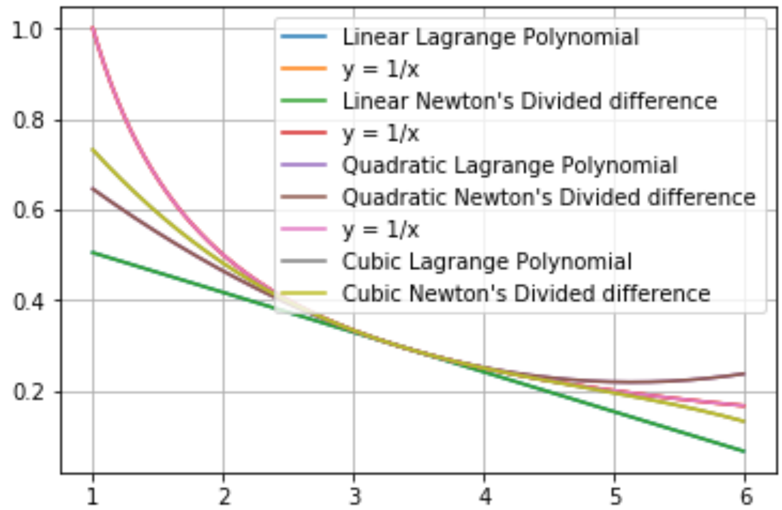



```
Newton's divided difference coefficients are:
[-1.  0.  4.]
```

We see that the interpolating function turns out to be the same for the Lagrange method and the Newton Divided Difference method.

Q4 Interpolation of $\frac{1}{x}$:

```
The Lagrangian polynomial is:

-0.08778 x + 0.5926
Linear Newton's divided difference coefficients are:
[ 0.298507 -0.08778 ]


The Lagrangian polynomial is:
         2
0.02493 x - 0.2561 x + 0.8766
Quadratic Newton's divided difference coefficients are:
[ 0.298507   -0.08778     0.02493333]


The Lagrangian polynomial is:
           3          2
-0.006133 x + 0.0878 x - 0.4708 x + 1.121
Cubic Newton's divided difference coefficients are:
[ 0.298507   -0.08778     0.02493333 -0.00613333]
```

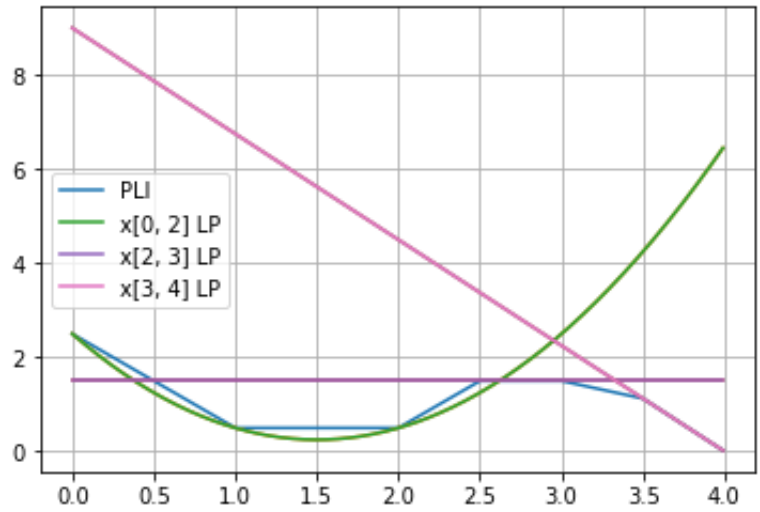


Lagrange interpolation is mostly just useful for theory. Computing with it requires huge numbers and catastrophic cancellations. In floating point arithmetic this is very bad. It does have some small advantages: for instance, the Lagrange approach amounts to diagonalizing the problem of finding the coefficients, so it takes only linear time to find the coefficients. This is good if you need to use the same set of points repeatedly. But all of these advantages do not make up for the problems associated with trying to actually evaluate a Lagrange interpolating polynomial.

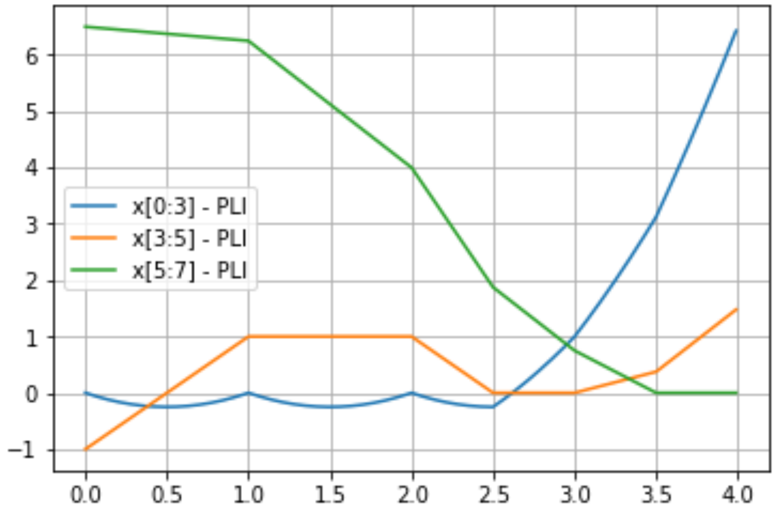
With Newton interpolation, you get the coefficients reasonably fast (quadratic time), the evaluation is much more stable (roughly because there is usually a single dominant term for a given x), the evaluation can be done quickly and straightforwardly using Horner's method, and adding an additional node just amounts to adding a single additional term. It is also fairly easy to see how to interpolate derivatives using the Newton framework.

Q5 x = [0, 1, 2, 2.5, 3, 3.5, 4] y = [2.5, 0.5, 0.5, 1.5, 1.5, 1.125, 0]

```
The Lagrangian polynomial is:
   2
1 x - 3 x + 2.5

x[0:3] Newton's divided difference coefficients are:
[ 2.5 -2.   1. ]


The Lagrangian polynomial is:

1.5

x[3:5] Newton's divided difference coefficients are:
[1.5 0. ]


The Lagrangian polynomial is:

-2.25 x + 9

x[5:7] Newton's divided difference coefficients are:
[ 1.125 -2.25 ]
```



Now comparing with Piecewise Linear Interpolation



The Lagrange Polynomial/Newton Divided Difference is closest to the Piecewise Linear function in the interval in which the Polynomial is made i.e. The Lagrangian Polynomial made by taking two points is closest to the Piecewise Linear function between those points.