

# Numerical Analysis HW09 : Polynomial Interpolations

Ming-Chang Chiu 100060007

May 5, 2015

## 1 Objective

In this assignment, we are given different numbers of support points in each .dat file, with  $475 \leq x \leq 775$ . We are required to use Lagrange interpolation to interpolate the Y values for each  $475 \leq x \leq 775$  and then observe the results as well as the difference between given values and interpolated values. I implemented

```
double Lagrange(double x,VEC &XDATA,VEC &YDATA);
```

as my interpolating function.

## 2 Implementation of Lagrange

In my implementation, I use the non-recursive form of Neville's algorithm. The theorem of Neville's algorithm is like the following:

Given  $n+1$  support points  $(x_i, y_i)$ ,  $i = 0 \dots n$ , with  $x_j \neq x_k$  if  $j \neq k$ , then the Lagrange interpolation at the point  $x$ ,  $F_{0 \dots n}(x)$ , can be calculated using the following recursion formula:

$$F_i(x) = y_i$$
$$F_{i_0 i_1 \dots i_k}(x) = \frac{(x - x_0)F_{i_1 i_2 \dots i_k}(x) - (x - x_{i_k})F_{i_0 i_1 \dots i_{k-1}}(x)}{x_{i_k} - x_{i_0}}$$

I implemented its non-recursive form for efficiency issue. I create an array, NS, to store the intermediate value such as  $F_{23}$  etc.; in each iteration of *forloop*, part of the NS will be updated and then given to next iteration. Finally the interpolated result will be stored in NS[0] and returned. The whole code snippet can be viewed as follow:

```

double Lagrange(double x,VEC &XDATA,VEC &YDATA){
    int n = XDATA.len();
    double NS[n];
    int i,j,k;
    for (i=0; i<n; i++) NS[i]=YDATA[i];
    for (k=1; k<n; k++) {
        for (j=0; j<n-k; j++) {
            NS[j]=((x-XDATA[j])*NS[j+1]-(x-XDATA[k+j])*NS[j])
                /(XDATA[j+k]-XDATA[j]);
        }
    }
    return NS[0];
}

```

### 3 Workflow

**Usage:** For example, `./hw09.out < f3.dat`

**Solve:** Non-recursive Neville's algorithm is applied.

**Desired output:** The program will build a file named *data.txt*, which will store value of  $x_i$ , interpolated value, error at  $x_i$ , and original value at  $x_i$  in f301.dat sequentially each row. One can load the .txt file for further analysis.

### 4 Results

Input File	Max error	Max error within $550 \leq x \leq 700$
f3.dat	372.8669	372.8669
f5.dat	248.3406	233.3644
f7.dat	379.1073	147.5782
f13.dat	1283.4489	39.6189
f21.dat	16728.5648	17.804

### 5 Plot Analysis

The error is defined as  $|Lagrange(x_i) - f301(x_i)|$

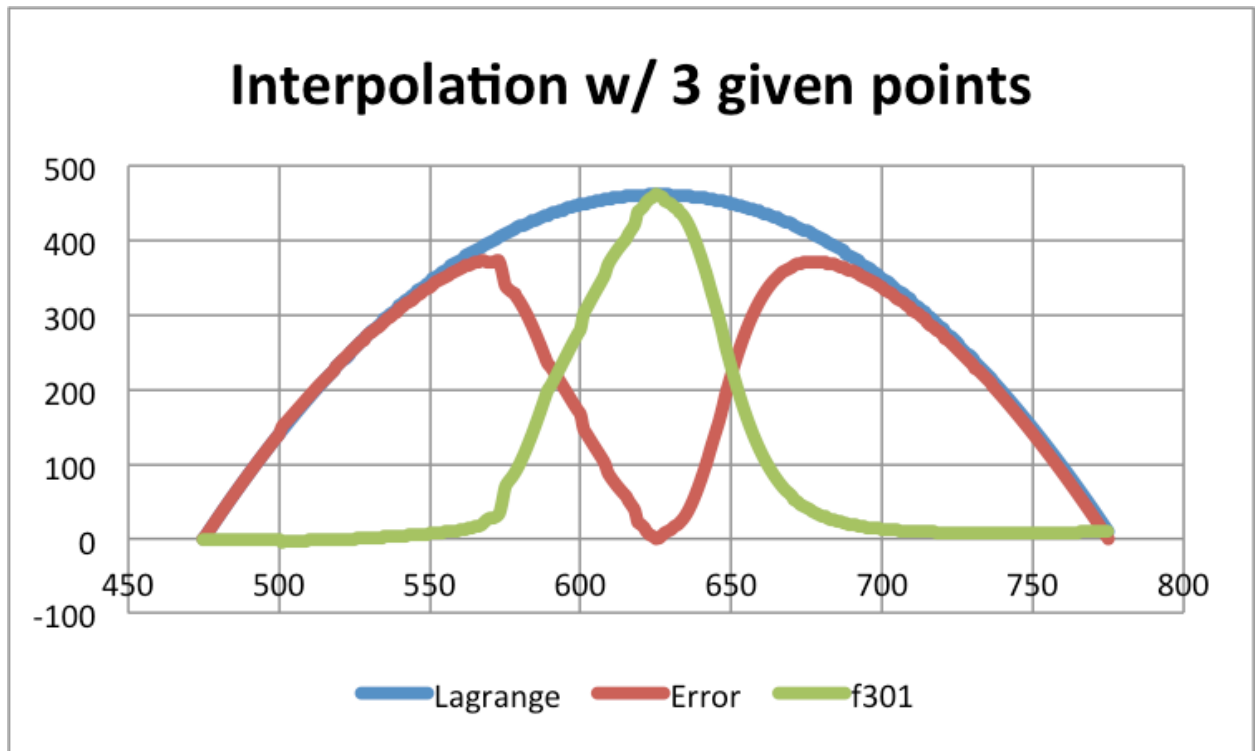


Figure 1: Interpolation with 3 support points

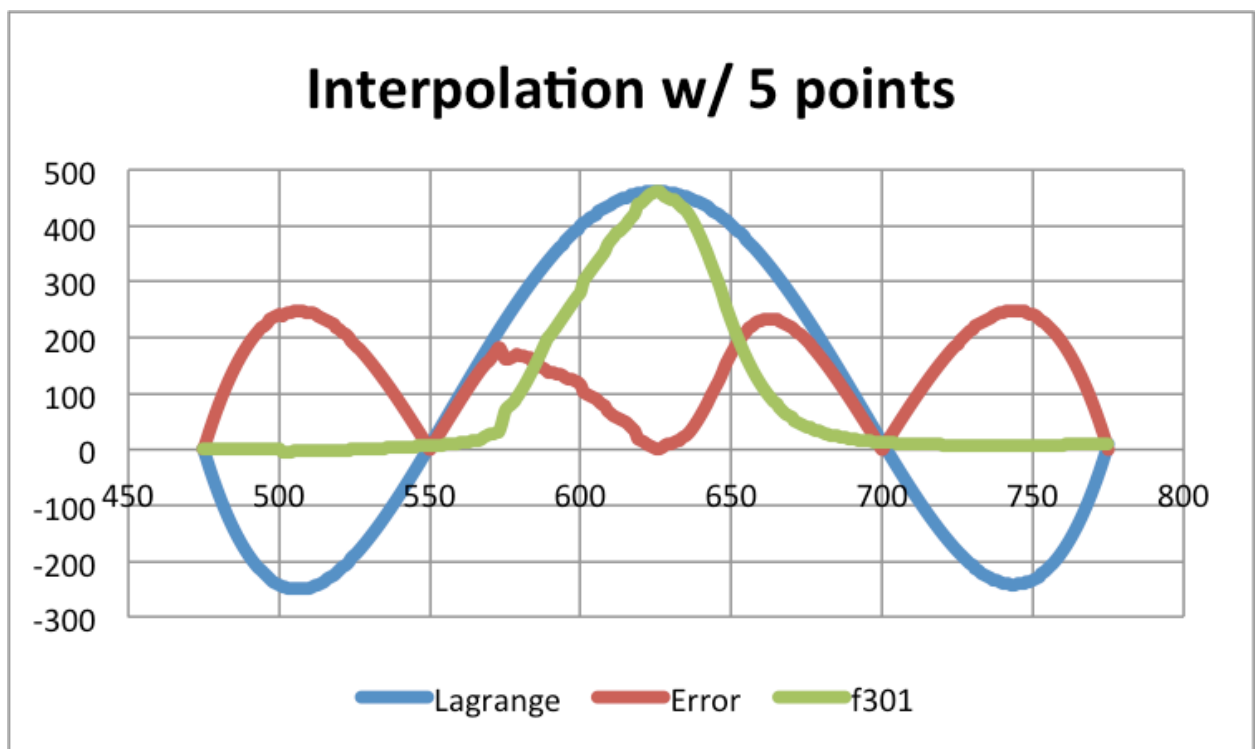


Figure 2: Interpolation with 5 support points

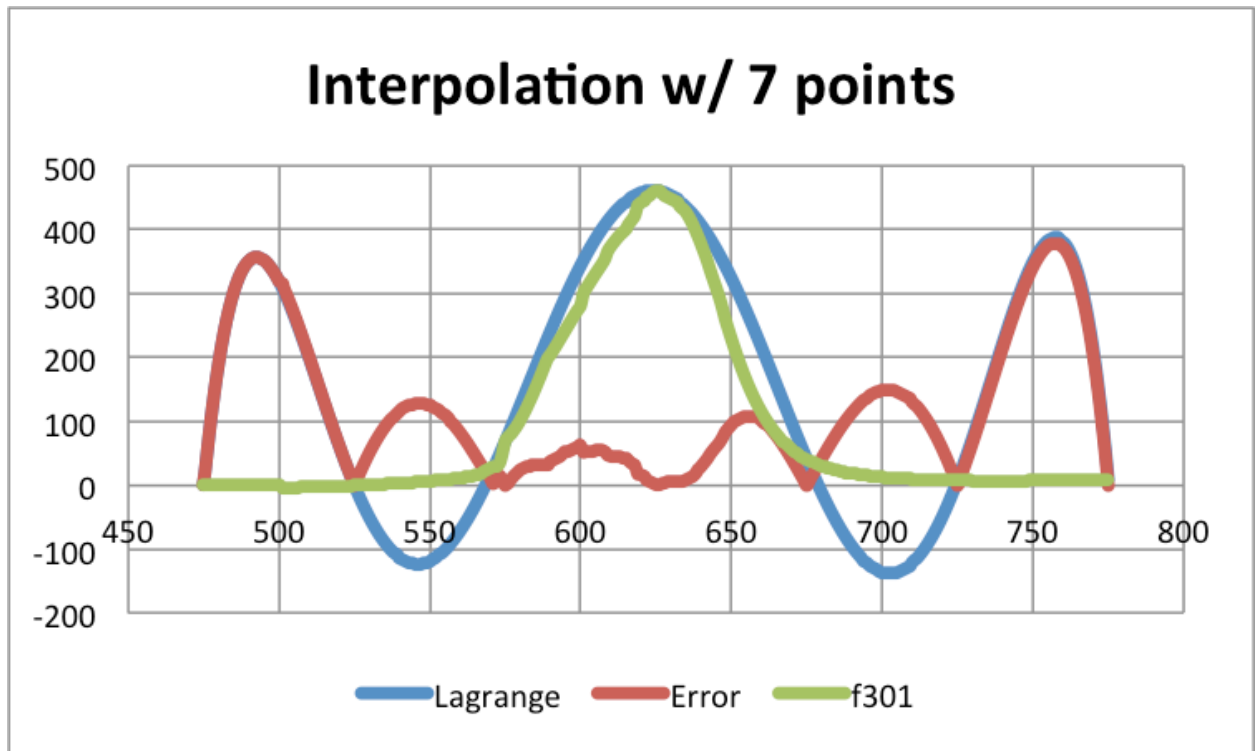


Figure 3: Interpolation with 7 support points

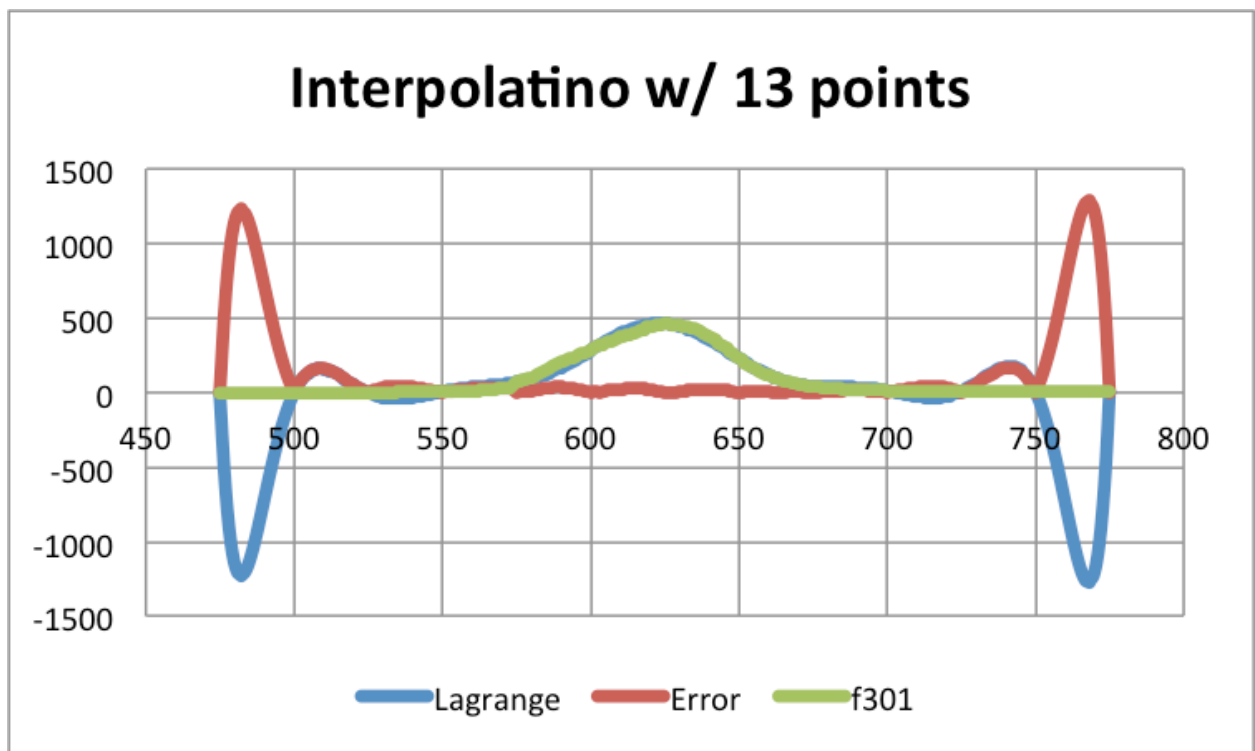


Figure 4: Interpolation with 13 support points

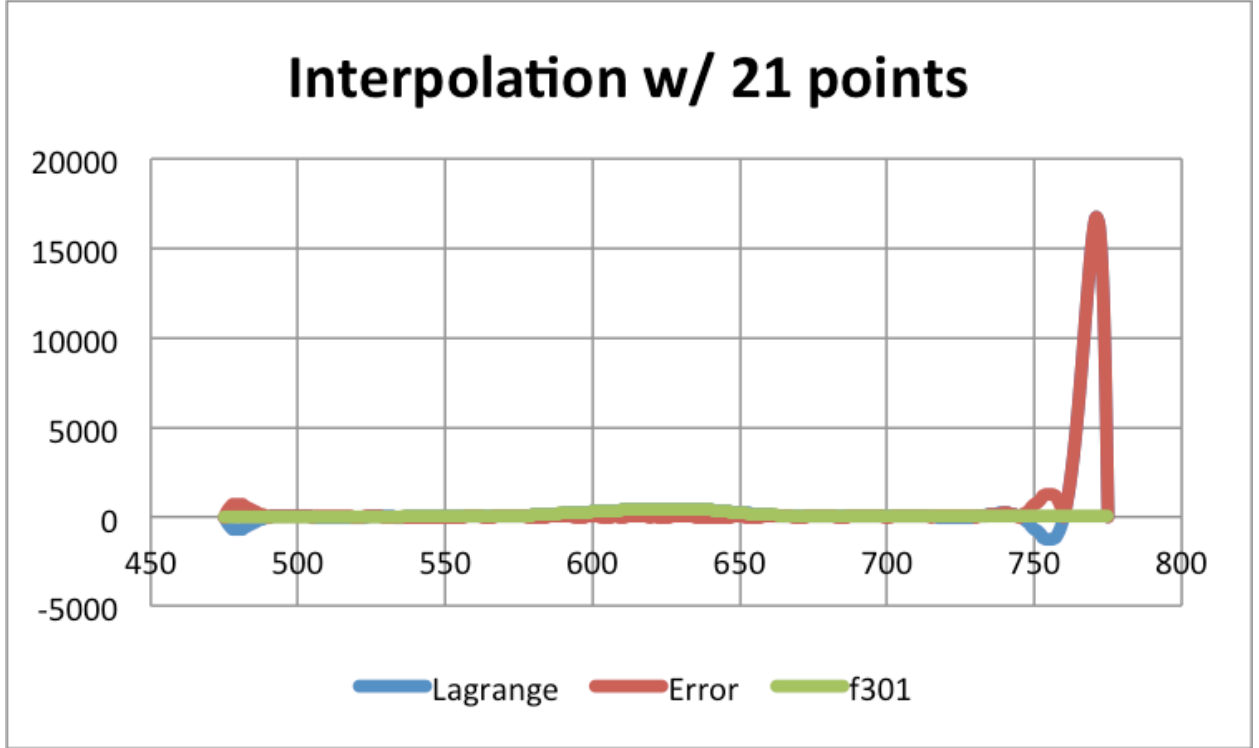


Figure 5: Interpolation with 21 support points

## 6 Observations

In general, I would expect that with more support points, the more approximate the interpolated waveform will be. The result tells me that it is partially true. From Figure1, we can see that at the center of interval  $[475, 775]$  the error is the smallest and it will ascend and then descend as  $X$  increases or decreases, the error plot is like a humped camel's back. This is because we have only three support points and the order of Lagrange interpolation is 2.

In sum, from the table in Section 4, the maximal error within  $[550, 700]$  is decreasing as the order of Lagrange formula increases (i.e. more support points), which means Lagrange is more and more approximate to the original waveform. But for all range of  $X$ , the output could have very large numerical error. From my plots, at right and left side of these plots, the error is getting larger since f5.dat, a extreme case is: in Figure 5, with 21 support points, Lagrange formula order is 20, the error can reach about 16728, which is quite massive. But the waveforms of Lagrange and f301 overall tend to be more and more similar within  $[550, 700]$  as more support points we have, which partially matches my original conjecture.

One interesting thing should be noticed is that in Figure1, 2, 3 and 4, the Lagrange waveforms are symmetric but in Figure 5 the waveform is asymmetric and has extreme

numerical error within  $[750,775]$ (Error and Lagrange behave the same within this interval).