

SC-374  
Computational and Numerical Methods  
Assignment 3

Amarnath Karthi 201501005  
Chahak Mehta 201501422

August 19, 2018

**1**  $y = \sqrt{x}$

Carry out the Lagrange linear interpolation between  $(1, 1)$  and  $(4, 2)$ . Plot your interpolation function together with  $y = \sqrt{x}$  for comparison.

The first order lagrange interpolation polynomial is defined as follows:

$$l_1(x) = \frac{x - x_1}{x_0 - x_1}y_0 + \frac{x - x_0}{x_1 - x_0}y_1 \quad (1)$$

Placing in the given values  $(1, 1)$  and  $(4, 2)$  we get the following simplified equation:

$$l_1(x) = \frac{x}{3} + \frac{2}{3} \quad (2)$$

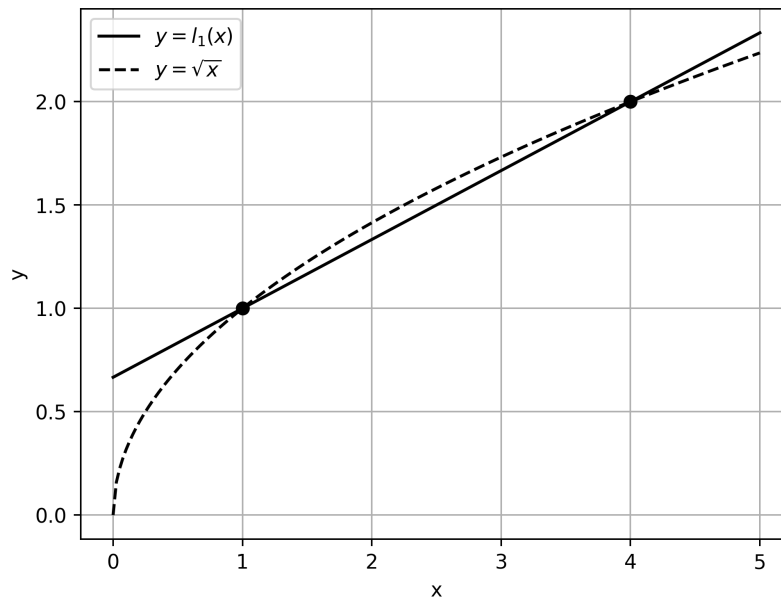


Figure 1: First order lagrange polynomial for the function  $y = \sqrt{x}$

Notice in the figure above that the first order Lagrange interpolation polynomial  $l_1(x)$  is just a **secant** line passing through the two points of interpolation.

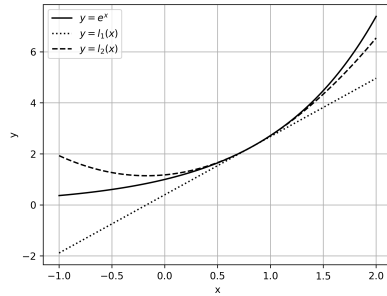
## 2 $y = e^x$

Construct out a Lagrange linear interpolation for  $(0.82, 2.270500)$  and  $(0.83, 2.293319)$ . Extend your study with a Lagrange quadratic polynomial using  $(0.84, 2.316367)$ . Compare your polynomials with the function  $y = e^x$ , plotting all of them on the same graph.

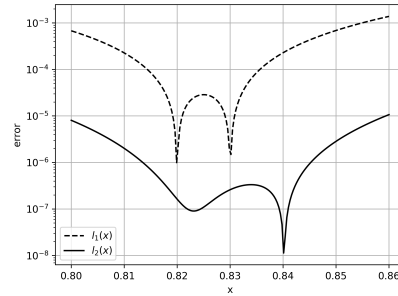
See **equation 1** for the first order Lagrange interpolation of a given set of points. The quadratic Lagrange interpolation is given by:

$$l_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}y_2$$

Using the above formulae we perform the quadratic interpolation and match the results with  $e^x$ .



(a)  $e^x$  and its interpolations



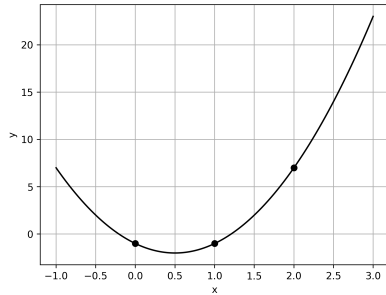
(b) error

**Figure 2a** shows the plots of  $e^x$ , as well as the first and the second order Lagrange interpolation polynomials obtained from its samples. Notice that while the first order interpolation just gives us a line which barely represents the curve in a very small neighbourhood, the second order interpolation is a more accurate representation of the curve.

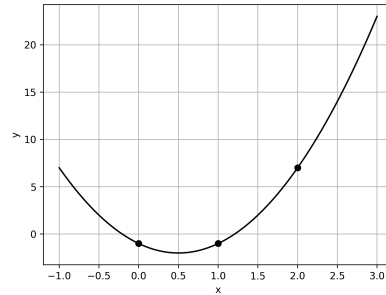
**Figure 2b** is the log plot of the magnitude of error between the actual function and the interpolations. Notice that irrespective of the point in question, the quadratic interpolation is always more accurate than its linear counterpart for the same points of interpolation. Also notice that there are spikes at the points of interpolation. This is because at these points, the interpolation curves intersect the actual function, thereby giving a 0 error at those points. As we approach the point of interpolation, the error decreases, becomes 0 at the point and then again starts increasing.

### 3 Newton's divided-difference method - NDD

Construct a quadratic Lagrange polynomial using the points  $(0, -1)$ ,  $(1, -1)$ , and  $(2, 7)$ . Plot your result. Extend this entire exercise with Newton's divided-difference quadratic polynomial and compare the two methods.



(a) Second order Lagrange interpolation



(b) Second order NDD interpolation

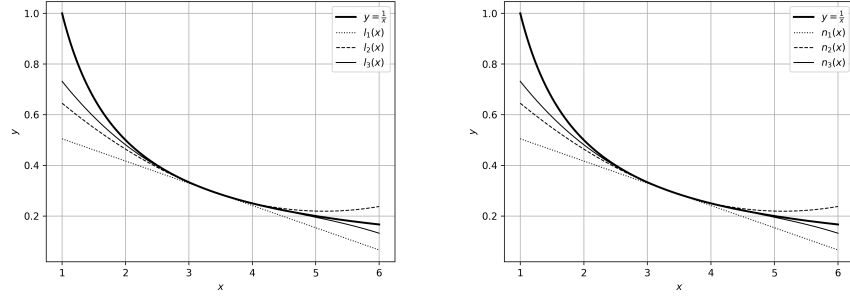
In this case, irrespective of the interpolation method used, we get the same results for the second order interpolation. Therefore the 2 methods give us the same results here.

This can be shown to be the case in general. Suppose we are interpolating over  $n$  distinct sample points. Therefore we will be getting an  $(n - 1)^{th}$  order polynomial. There is one and only one polynomial of this order:

$$p_n(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$$

Substituting the  $n$  sample points in the equation, we will get a linear equation in  $n$  variables, each being the coefficients in the polynomial. These will therefore have one and only one solution. This proves that both the Lagrange interpolation and the Newton's divided difference method will give the same polynomial.

4  $y = \frac{1}{x}$



(a) Second order Lagrange interpolation (b) Second order NDD interpolation

Figure 4: Interpolations of  $\frac{1}{x}$

In the above figures we can see that both the methods give use the same results for a given order of interpolation. Also note that as we increase the order, the range over which the interpolated curve resembles the function curve increases.

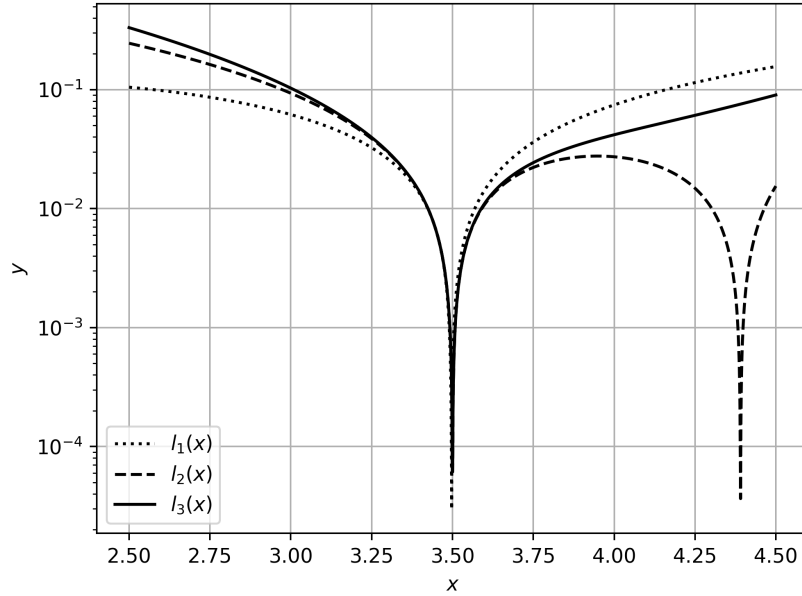


Figure 5: Error log plot

Notice in the above figure, spikes occur wherever the interpolated curve intersects the function.

## 5 Piecewise interpolation

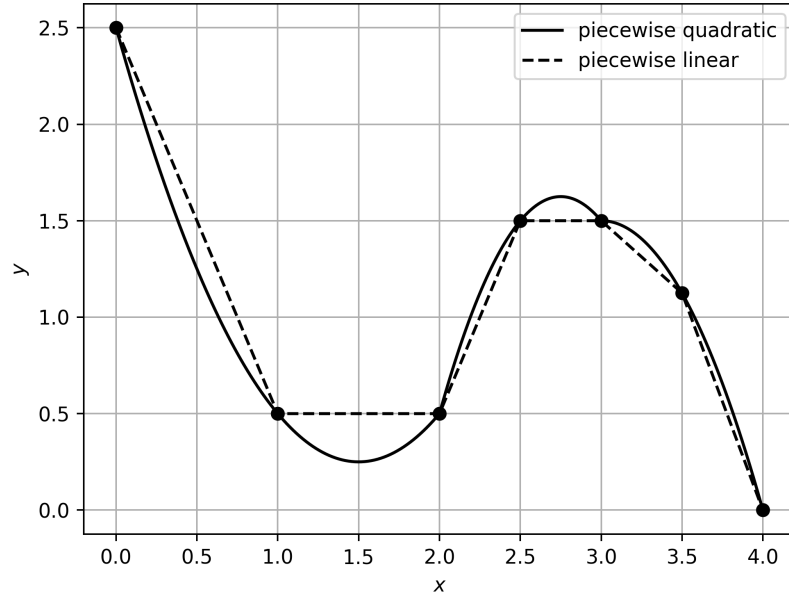
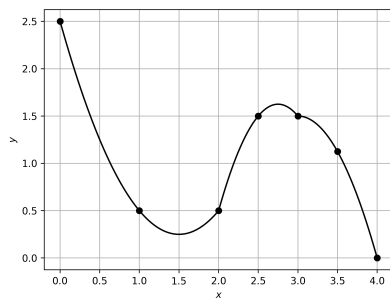
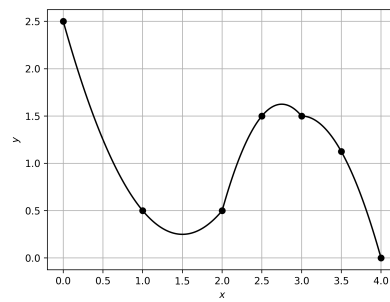


Figure 6: Piecewise linear and quadratic interpolations

In the above figure, we can see that even though the interpolation curves are continuous, they are not differentiable at the points of join.



(a) Piecewise Lagrange interpolation



(b) Piecewise NDD interpolation

Figure 7: Lagrange and NDD piecewise interpolation

Notice that Lagrange and NDD methods give the same interpolation polynomial.

## 6 A comparison of the 2 interpolation methods

### 6.1 Errors

Both the methods give the same polynomials and therefore have the same errors. Thus there is no difference between the 2 in the aspect of results theoretically. Also both of these methods have a drawback of exhibiting **Runge's Phenomenon**, which is basically a case where a higher order interpolation gives more error than a lower order one, because of oscillations at the end points. Essentially, this implies that *Adding more data sometimes leads to more errors*. Therefore one must choose the interpolation points wisely.

### 6.2 Computational Complexity

In terms of computational complexity, Newtons divided difference is much better than Lagrange's method. This is because Newtons divided method has a recursive approach. Therefore when a new point is added to the data set, there is only a slight calculation involved. On the other hand, Lagrange's polynomial has to be re calculated from scratch on the addition of a new point. Also the evaluation of a Lagrange polynomial is more complex because of the high amount of calculations. Newtons method is more efficient when we want to interpolate data incrementally.

## 7 Cubic spline interpolation

Given  $n$  ordered sample points  $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1})$  the cubic spline interpolation gives us a set of  $n - 1$  cubic order polynomials  $p_0, p_1, p_2, p_3, \dots, p_{n-2}$ , where  $p_i$  is the interpolation polynomial between  $(x_i, y_i)$  and  $(x_{i+1}, y_{i+1})$ , such that for any given point in the range  $(x_0, x_{n-1})$ , the resultant interpolation function is continuous, differentiable and also continuously differentiable.

Therefore we see that the second order derivative of the interpolation function is a linear interpolation of the given points. The interpolation function can be obtained by integrating the linear interpolation twice and adding the restrictions that the function is continuous and differentiable at each sample points.

Let  $M_i$  be the second order derivative of the interpolation function at the point  $(x_i, y_i)$ . Also at the corner points we take the second order derivative to be 0. Therefore we have:

$$\begin{aligned} M_0 &= 0 \\ M_{n-1} &= 0 \end{aligned}$$

Placing these conditions gives us the following recursive equation for  $M_i$ :

$$\frac{x_i - x_{i-1}}{6}M_{i-1} + \frac{x_{i+1} - x_{i-1}}{3}M_i + \frac{x_i - x_{i+1}}{6}M_{i+1} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

The above equation can be solved using simple Matrix Multiplication and inverse operations.

Using these values of  $M$ , we calculate the spline functions.

$$\begin{aligned} S(x) &= \frac{1}{6} \frac{M_{j-1}}{(x_j - x_{j-1})} (x_j - x)^3 + \frac{1}{6} \frac{M_j}{(x_j - x_{j-1})} (x - x_{j-1})^3 + Ax + B \\ A &= D - C \\ B &= Cx_j - Dx_{j-1} \\ C &= \frac{y_{j-1}}{(x_j - x_{j-1})} - \frac{M_{j-1}}{6} (x_j - x_{j-1}) \\ D &= \frac{y_j}{(x_j - x_{j-1})} - \frac{M_j}{6} (x_j - x_{j-1}) \end{aligned}$$

## 7.1 $y = \frac{1}{x}$

|     |   |               |               |               |
|-----|---|---------------|---------------|---------------|
| $x$ | 1 | 2             | 3             | 4             |
| $y$ | 1 | $\frac{1}{2}$ | $\frac{1}{3}$ | $\frac{1}{4}$ |

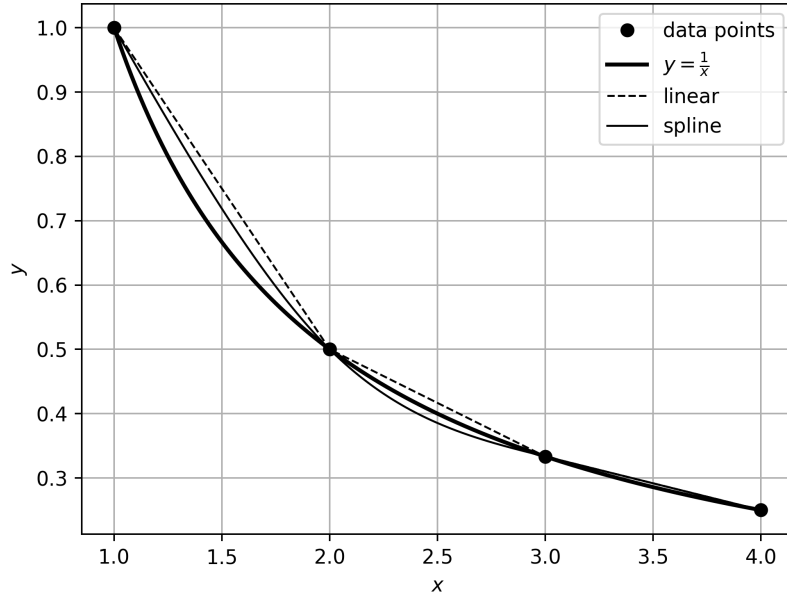


Figure 8: Spline interpolation and linear interpolations of  $y = \frac{1}{x}$



In the above figure notice the smoothness of the spline interpolation, despite of being a piecewise interpolation method. This is the main advantage of using splines. It always gives us smooth functions.

|       |     |
|-------|-----|
| $M_0$ | 0   |
| $M_1$ | 0.5 |
| $M_2$ | 0   |
| $M_3$ | 0   |

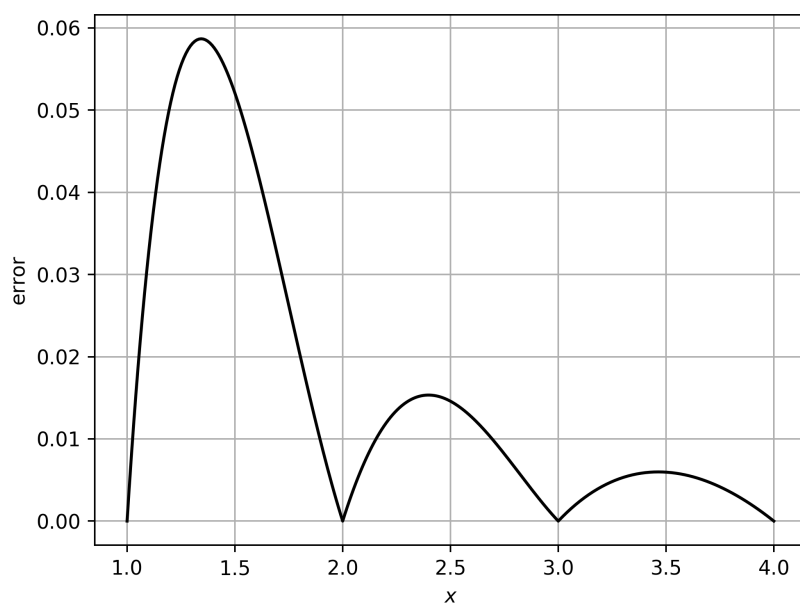
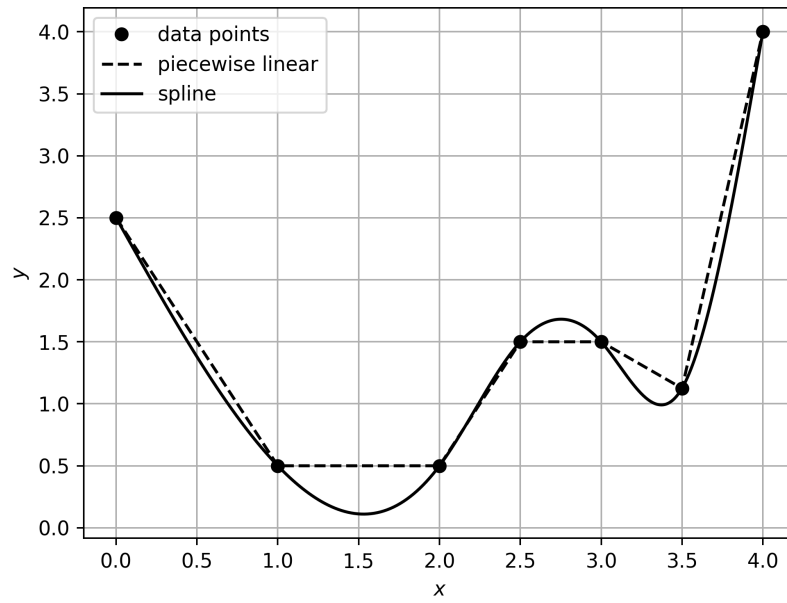


Figure 9: Spline interpolation error

Notice that the spline interpolation is not very accurate even in the region of interpolation.

## 7.2 Spline interpolation and linear interpolation

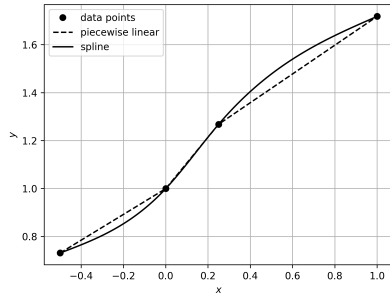
|     |     |     |     |     |     |       |   |
|-----|-----|-----|-----|-----|-----|-------|---|
| $x$ | 0   | 1   | 2   | 2.5 | 3   | 3.5   | 4 |
| $y$ | 2.5 | 0.5 | 0.5 | 1.5 | 1.5 | 1.125 | 0 |



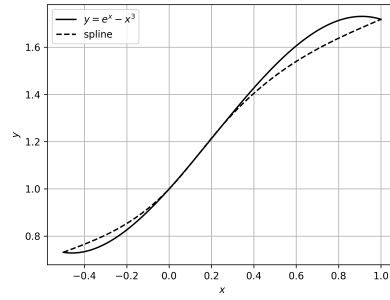
|       |         |
|-------|---------|
| $M_0$ | 0       |
| $M_1$ | 1.8481  |
| $M_2$ | 4.6075  |
| $M_3$ | -7.3412 |
| $M_4$ | 0.7576  |
| $M_5$ | -4.6893 |
| $M_6$ | 0       |

### 7.3 $e^x - x^3$

|     |        |   |        |        |
|-----|--------|---|--------|--------|
| $x$ | -0.50  | 0 | 0.25   | 1      |
| $y$ | 0.7315 | 1 | 1.2684 | 1.7183 |



(a) linear interpolation



(b) spline interpolation

|       |         |
|-------|---------|
| $M_0$ | 0       |
| $M_1$ | 2.4342  |
| $M_2$ | -1.7256 |
| $M_3$ | 0       |

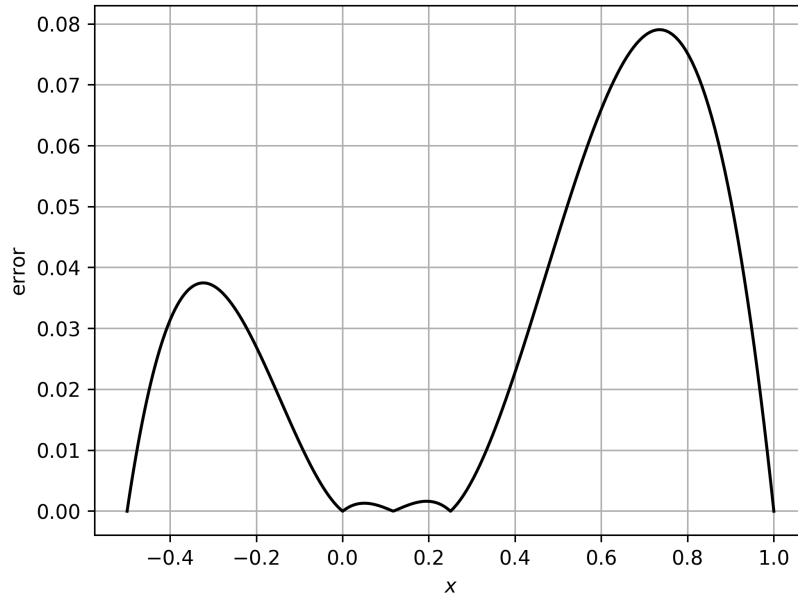


Figure 11: spline interpolation error