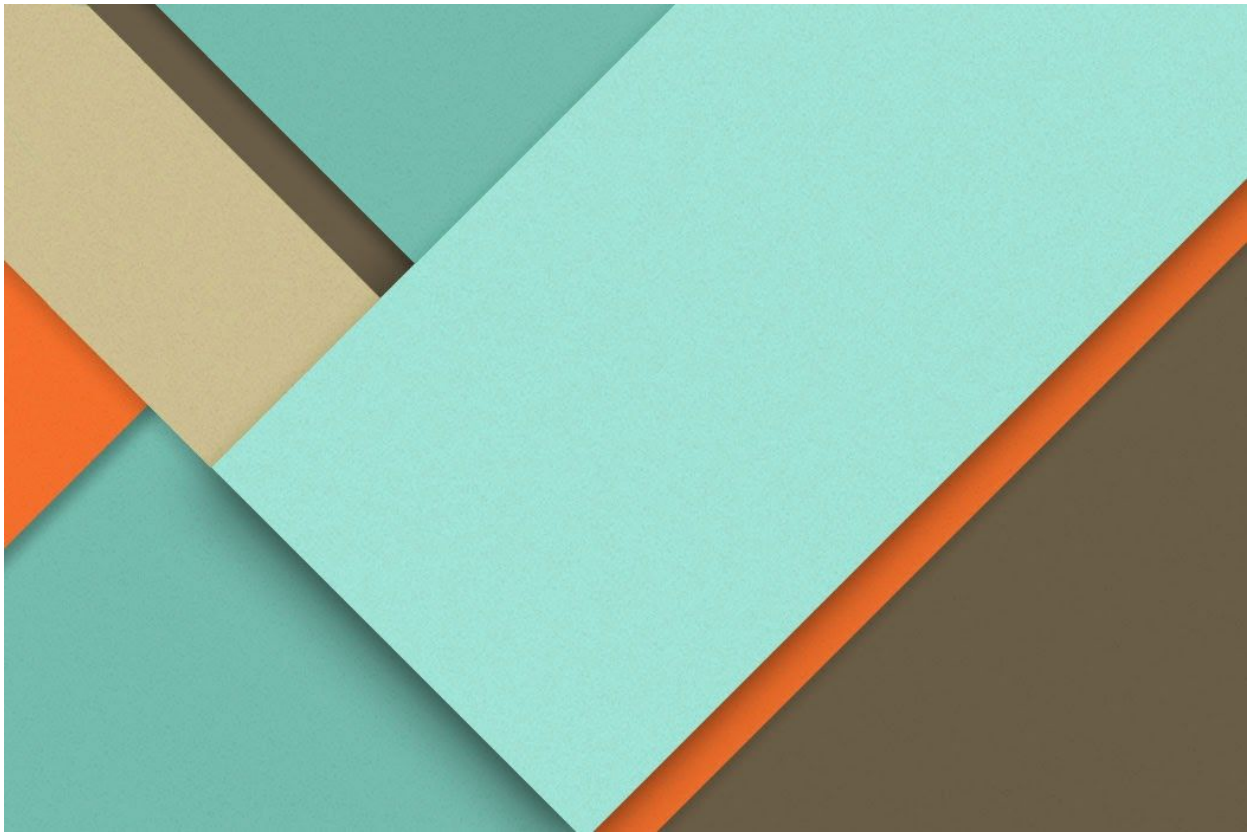


Numerical analysis

Part II

Afnan Mousa	15
Enas Morsy	20
Sara Mohammad	31
Shimaa kamal	34
Nada Fathy	68

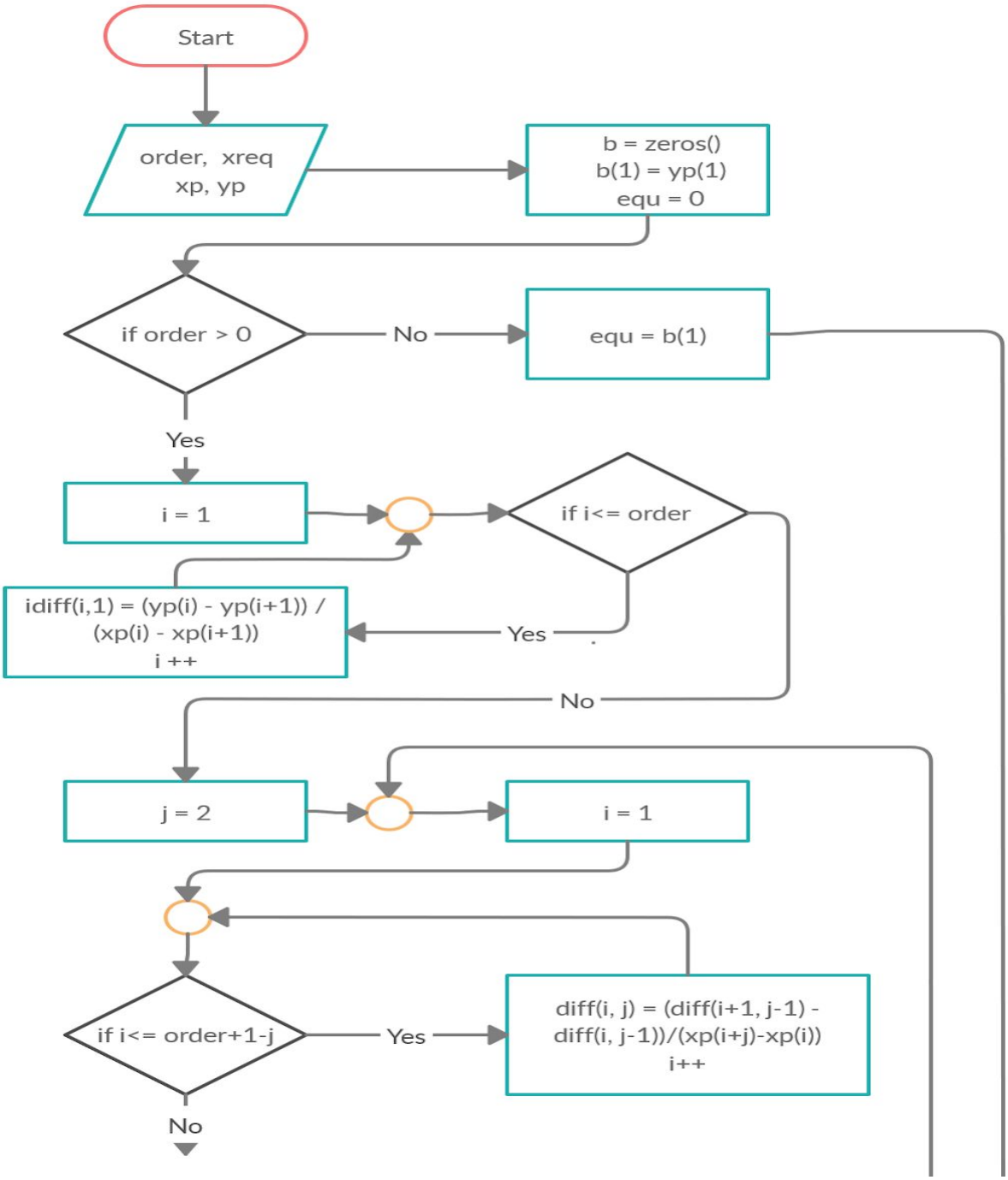
Attention please!!!

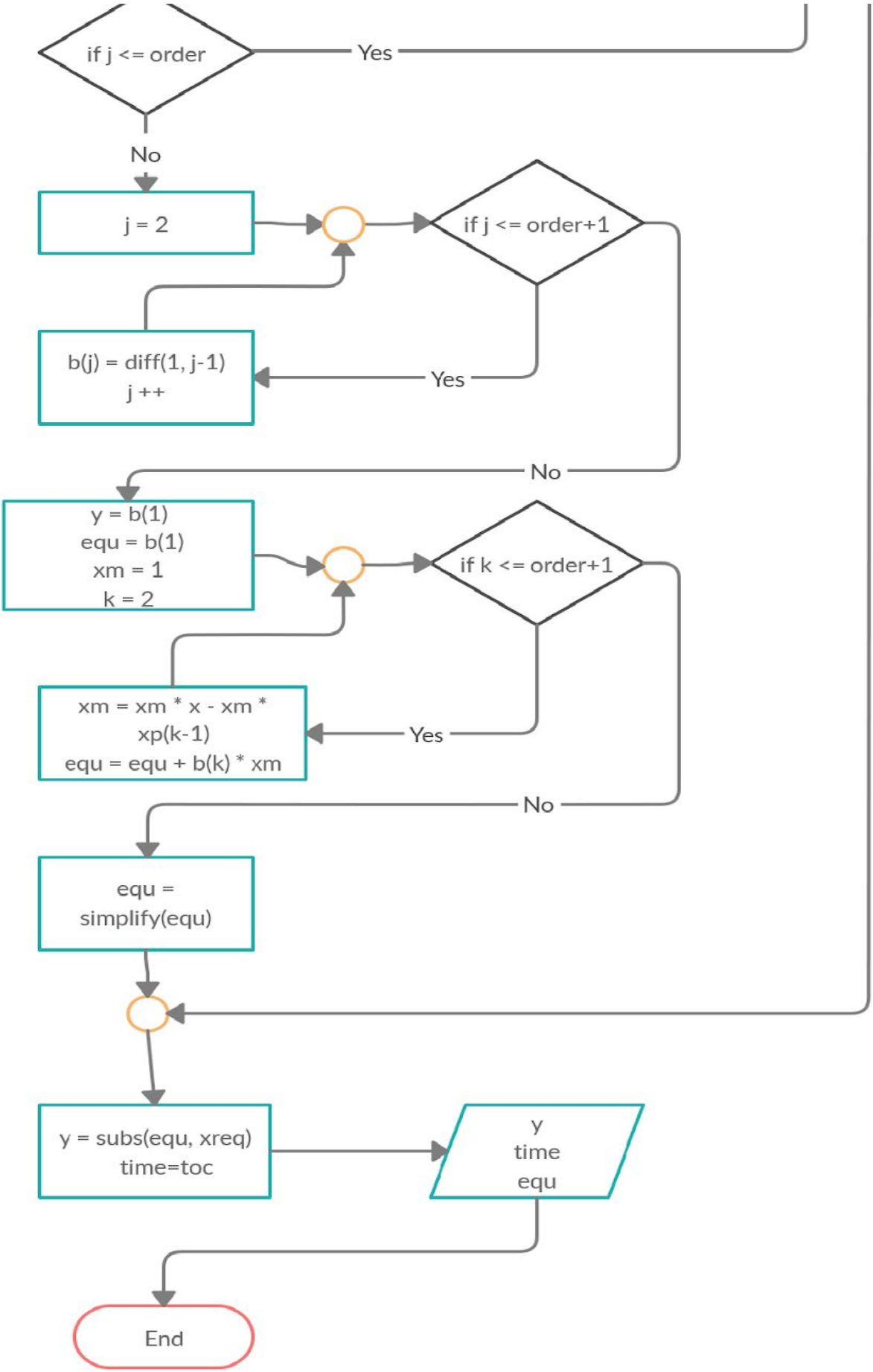
- This project basically is handled by matlab versions 2018, 2019 so it may cause troubles running this project with older versions.

- User guide is supplied at the end of this report with sample runs.

1. Newton interpolation

Flowchart




if j <= order
Yes
No
j = 2
if j <= order+1
b(j) = diff(1, j-1)
j ++
Yes
No
y = b(1)
equ = b(1)
xm = 1
k = 2
if k <= order+1
xm = xm * x - xm *
xp(k-1)
equ = equ + b(k) * xm
Yes
No
equ =
simplify(equ)
y = subs(equ, xreq)
time=toc
y
time
equ
End

Data structure

I. Matrices (2D arrays)

Used to store divided differences of given points.

II. Vectors (1D arrays)

- Used to store the calculated coefficient for each iterations in order to reuse them to form the final hypothesis equation.
- The given points stored in two vectors one for x coordinates and the other for y coordinates.

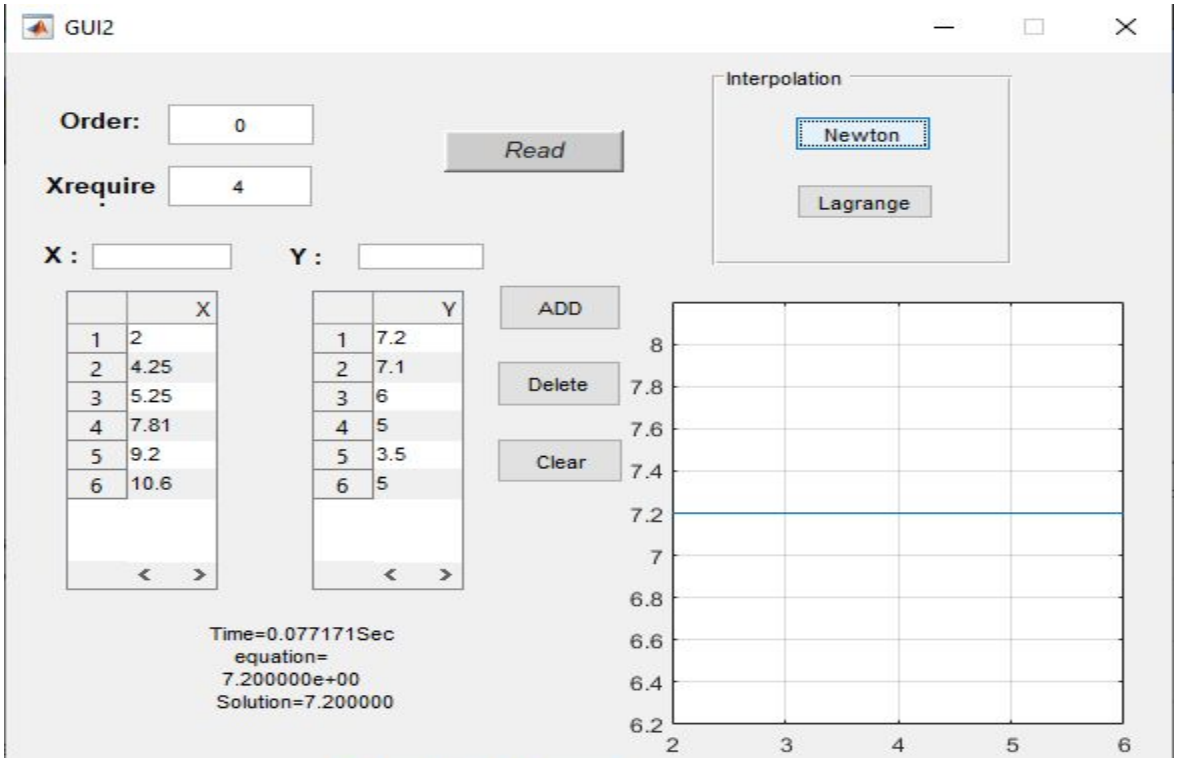
Analysis for the behavior

I. example one

In case of inputs:

- x =[2, 4.25, 5.25, 7.81, 9.2, 10.6]
- y = [7.2, 7.1, 6, 5, 3.5, 5]
- Newton(0, x, y, 10)

The order of the required equation is **zero** so the hypothesis function is constant and it is considered as the first item in y components vector.

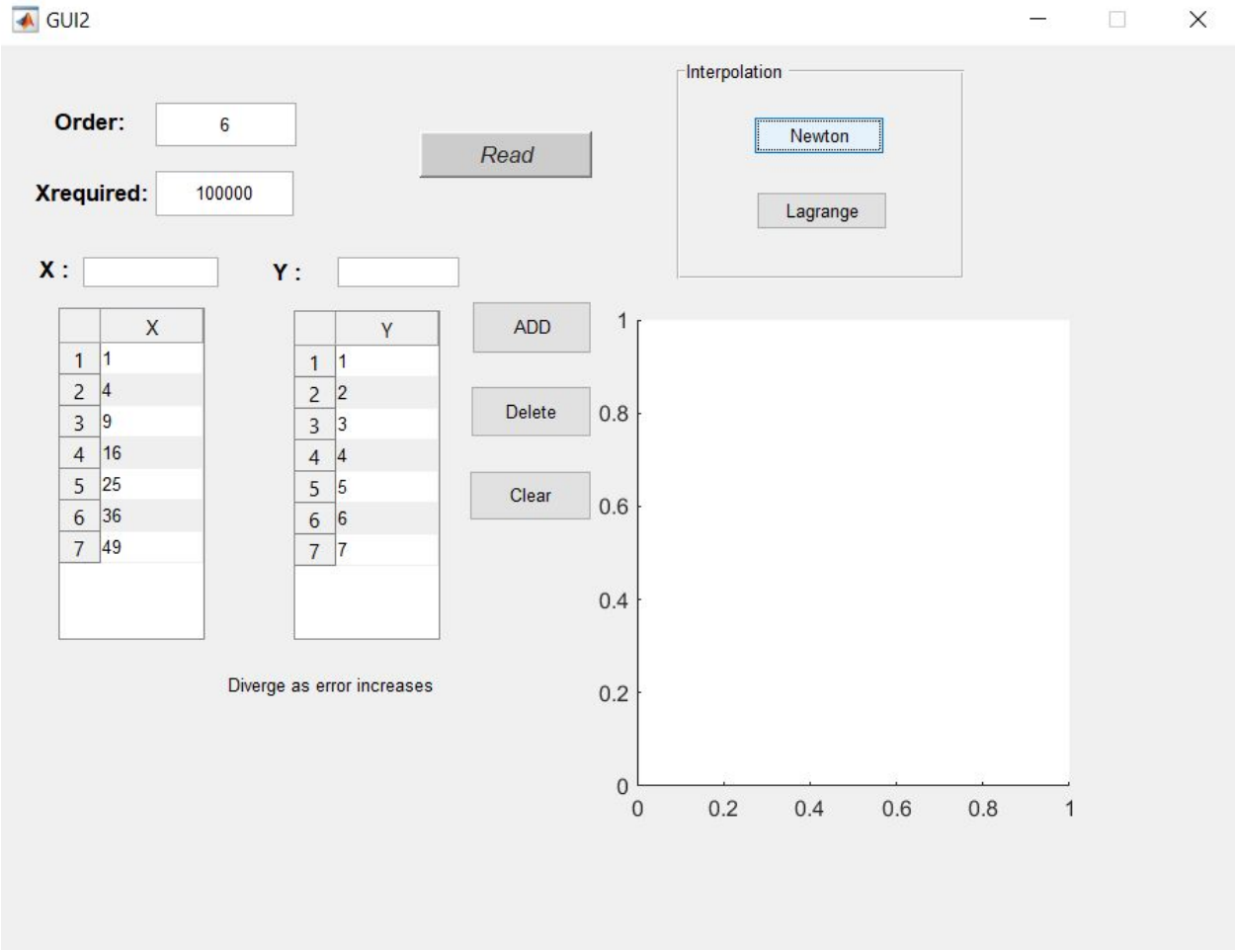


II. example two

In case of inputs:

- x =[1, 4, 9, 16, 25, 36, 49]
- y = [1, 2, 3, 4, 5, 6, 7]
- Newton(6, x, y, 1000000)

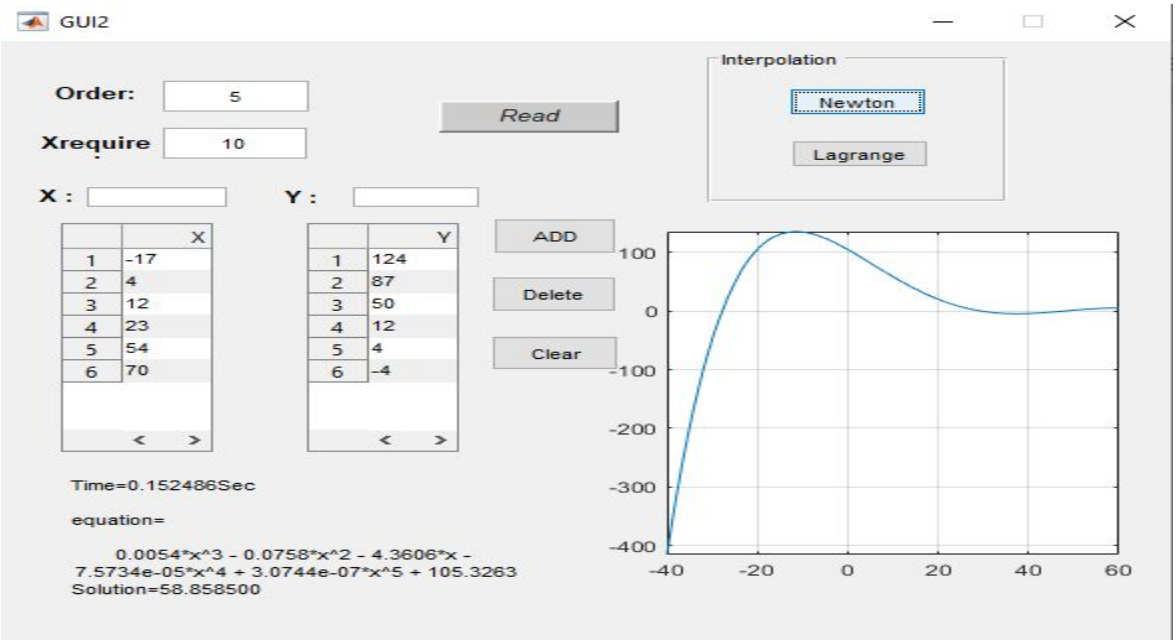
The root will still be overshot and the sequence will oscillate so it **diverges** here



III. example three

In case of inputs:

- x =[-17, 4, 12, 23, 54, 70]
- y = [124, 87, 50, 12, 4, -4]
- Newton(5, x, y, 10)



conclusion

- It works well with approximating complex functions.
- It may diverge in different cases
- Points interval with respect to the root's one is very important

Problematic functions

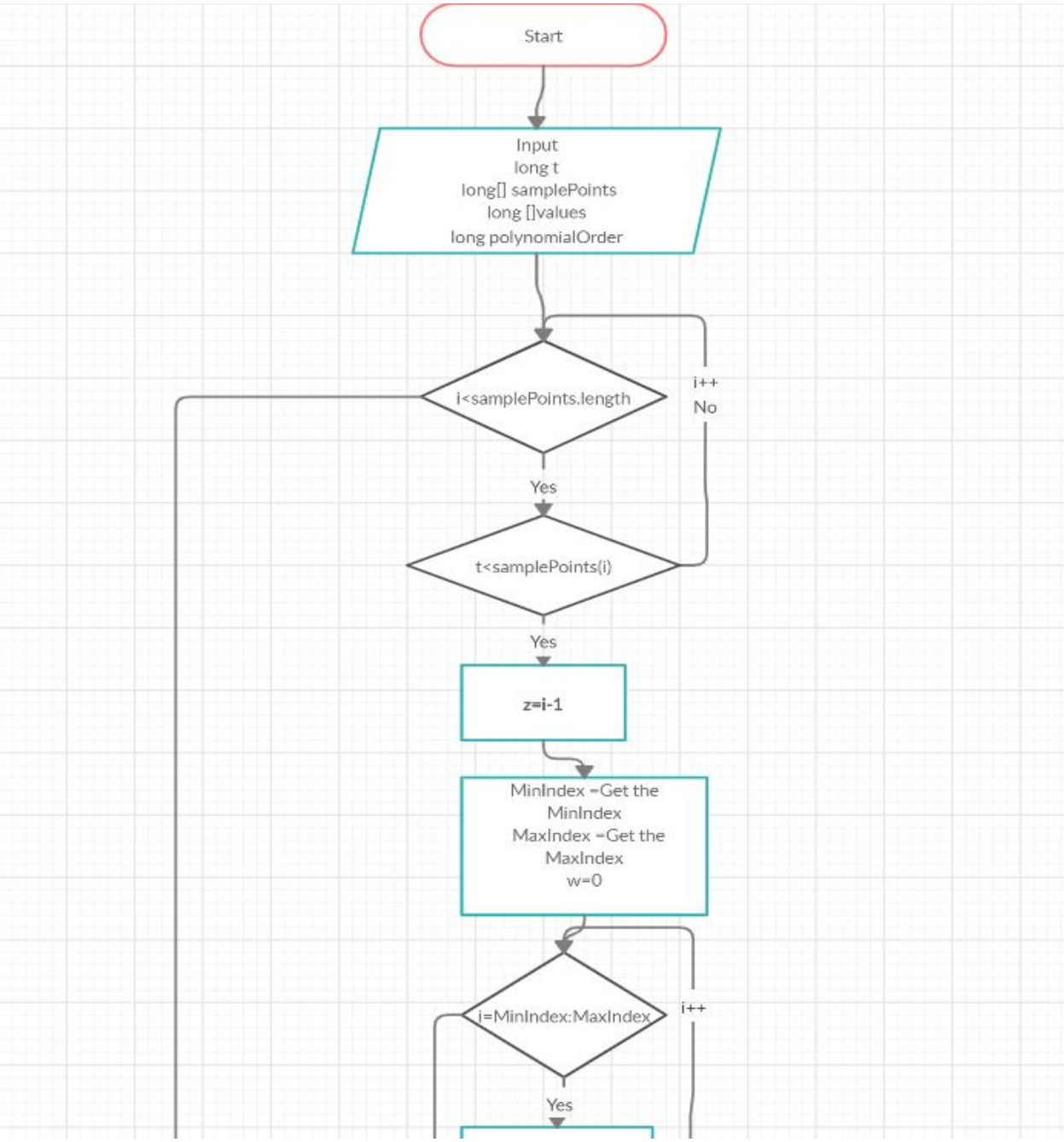
- There are functions with necessary convergence conditions satisfied, however they may diverge as the initial point is not in the interval where the method converges.
- Functions with saturation points may lead to dividing by zero or even if the derivative is small but not zero, the next iteration will be a far worse approximation.
- Some functions may stuck into infinite cycle where they hesitate between specific values

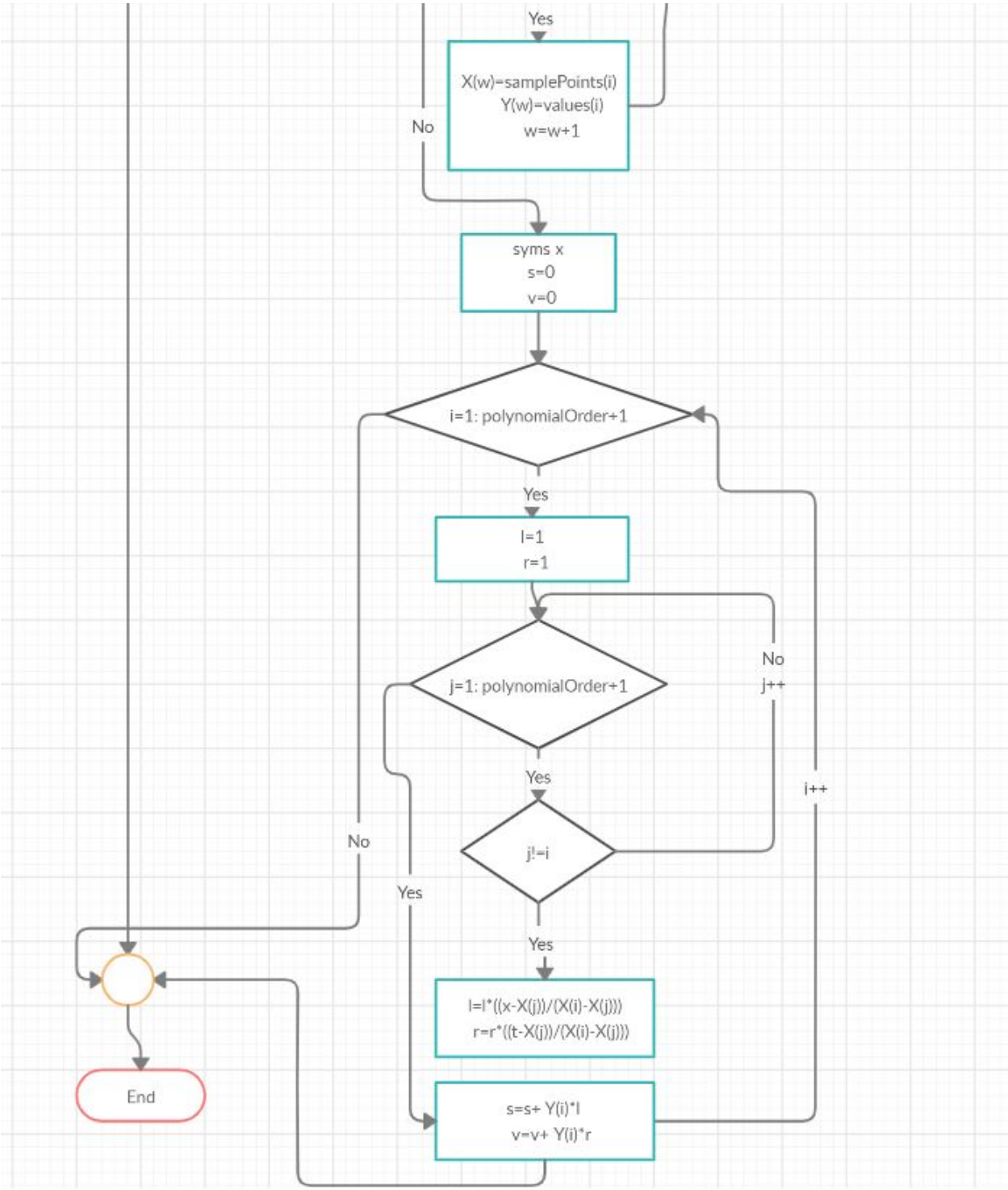
Suggestions:

1. try to guess a good initial point approximately near the root interval .
2. Try to use different sets of points as method inputs in order to avoid saturation points as much as possible.

2. Lagrange interpolation

Flowchart




Yes
X(w)=samplePoints(i)
Y(w)=values(i)
w=w+1
No
syms x
s=0
v=0
i=1: polynomialOrder+1
Yes
l=1
r=1
j=1: polynomialOrder+1
No
j++
Yes
j!=i
No
Yes
Yes
i++
l=l*((x-X(j))/(X(i)-X(j)))
r=r*((t-X(j))/(X(i)-X(j)))
s=s+ Y(i)*l
v=v+ Y(i)*r
End

Data structure

I. Array one dimension

Use it to determine the total points which work on it .

II. Array two dimension

Use to create the table which contains all the data to be shown to the user in GUI.

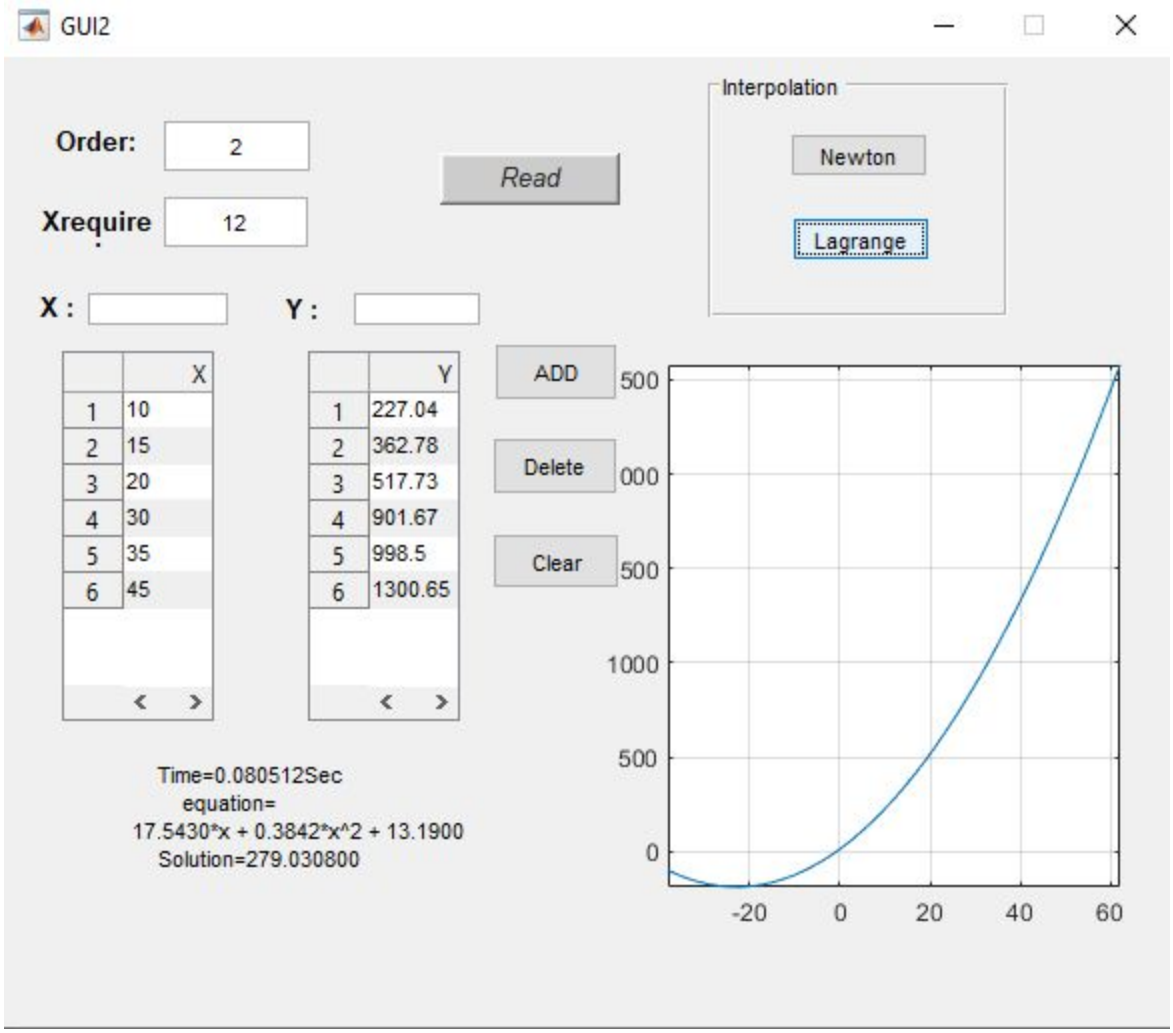
This table contains every iteration with each calculation .

Analysis for the behavior

I. example one

If the order is two and the X_required is equal 12 ,the Output is:

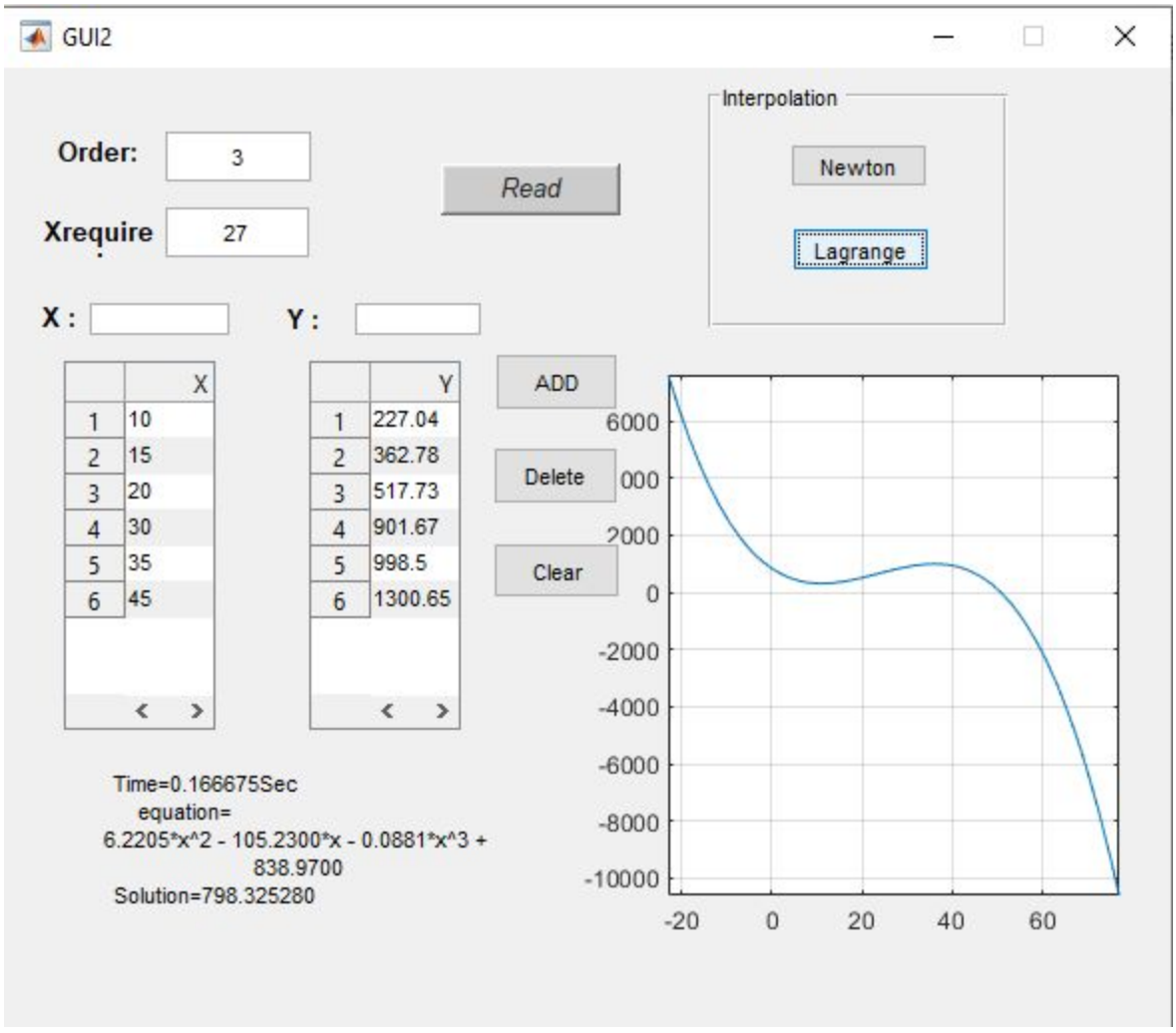
- The equation containing the high power of X is two .
- The value of Y corresponding to the X_required equals 279.030800 which Confined between two values 227.04 , 362.78 in Y and in X between 10,15that is true .


GUI2
Order:
2
Xrequire
12
Read
Interpolation
Newton
Lagrange
X :
Y :
ADD
Delete
Clear
X
1 10
2 15
3 20
4 30
5 35
6 45
Y
1 227.04
2 362.78
3 517.73
4 901.67
5 998.5
6 1300.65
Time=0.080512Sec
equation=
17.5430*x + 0.3842*x^2 + 13.1900
Solution=279.030800
500
000
500
1000
500
0
-20
0
20
40
60

II. example two

If the order is three and the X_required is equal 27 ,the Output is:

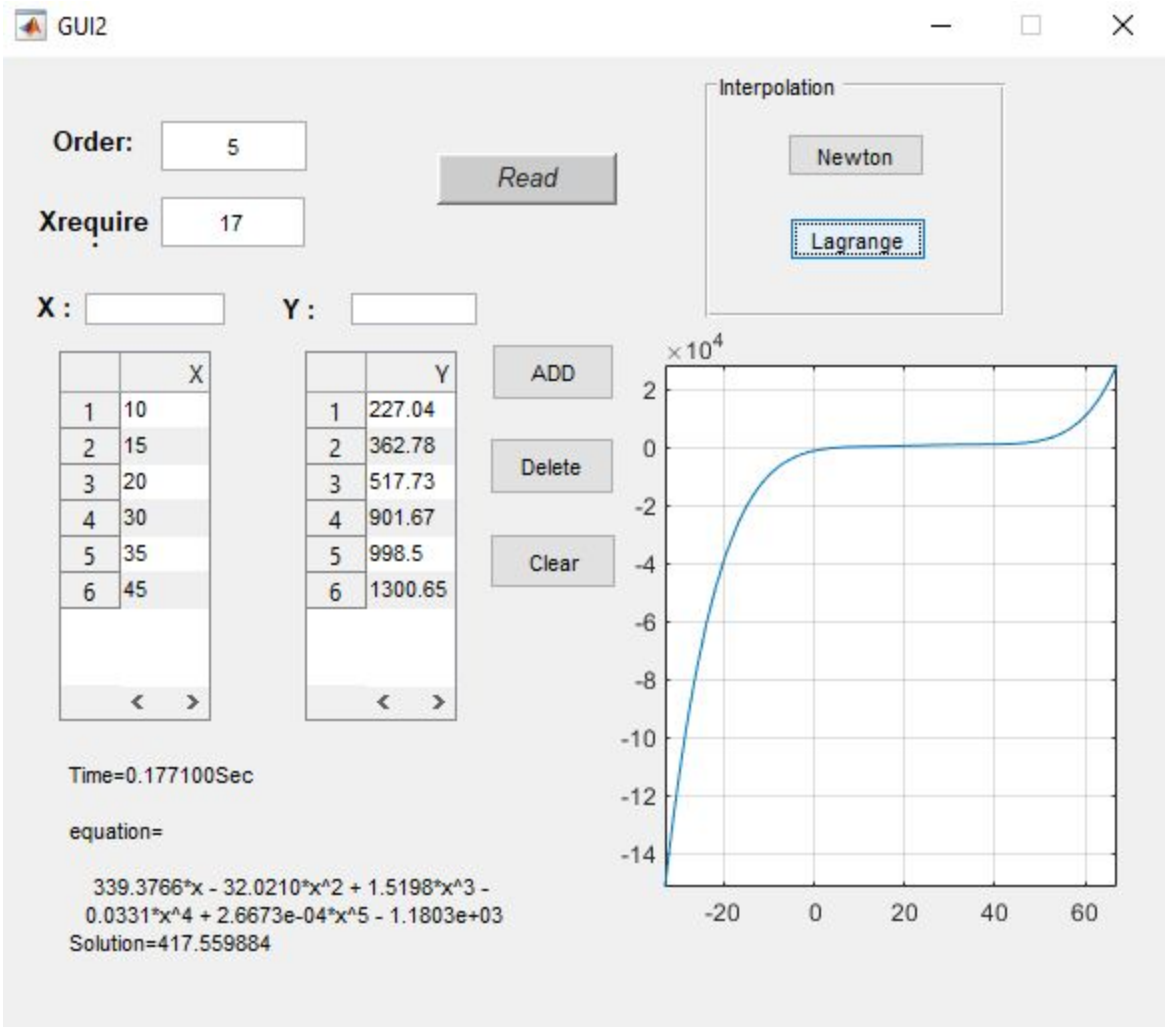
- The equation containing the high power of X is three .
- The value of Y corresponding to the X_required equals 798.32580 which Confined between two values 517.73 , 901.67 in Y and in X between 20,30that is true .



III. example three

If the order is five and the X_required is equal 17 ,the Output is:

- The equation containing the high power of X is five.
- The value of Y corresponding to the X_required equals 417.559884 which Confined between two values 362.78 , 517.73 in Y and in X between 15,20 that is true .



conclusion

- The greatest term in the equation depends on the order of polynomials.
- The number of points which Precedes the specific X_required must be greater then the point in the next X_required.
- The order must be less than the number of X table elements .
- We can read the input from the file or input it from the GUI handle .

Problematic functions

➢ One of the problematic functions :

Is not determine the points that are limited to the values of the point which need get the interpolation for it solve it

- by specific these points by specific the Minimum and maximum index in the origin table and make an iterator from minimum index to the max to take the value of all these points and save it in the new array .

➢ The difference between minimum index and the maximum index depend on the order of polynomials.

General algorithm

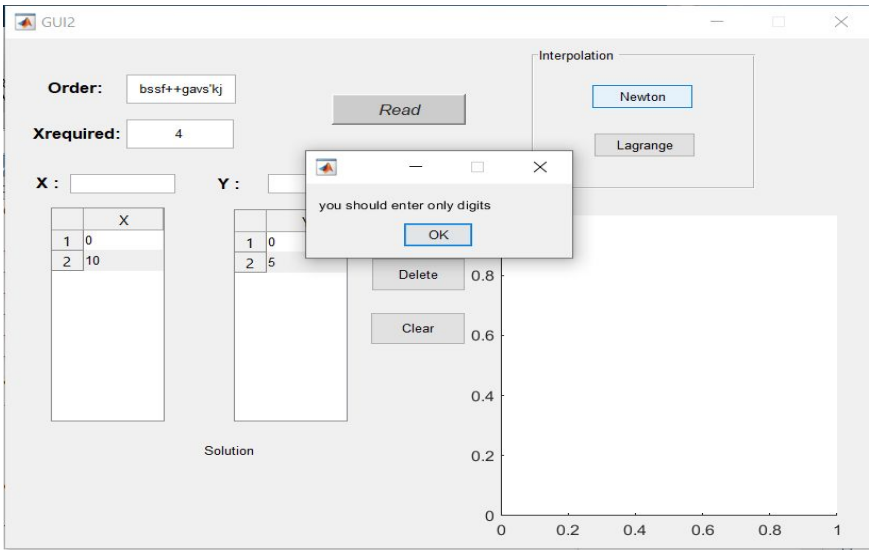
Lagrange interpolation is mostly just useful for theory. Actually computing with it requires huge numbers and catastrophic cancellations. In floating point arithmetic this is very bad. It does have some small advantages: for instance, the Lagrange approach amounts to diagonalizing the problem of finding 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.

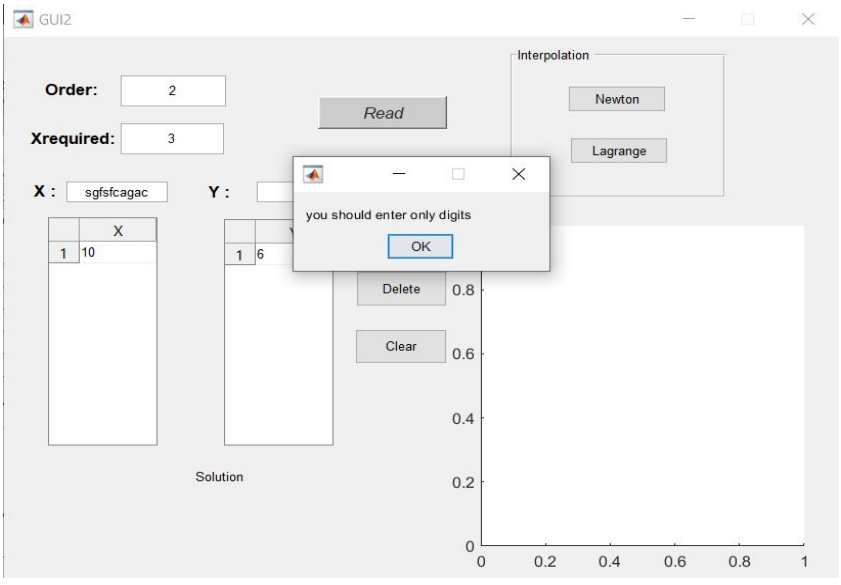
- Lagrange's form is more efficient when you have to interpolate several data sets on the same data points.
- Newton's form is more efficient when you have to interpolate data incrementally.

Sample runs

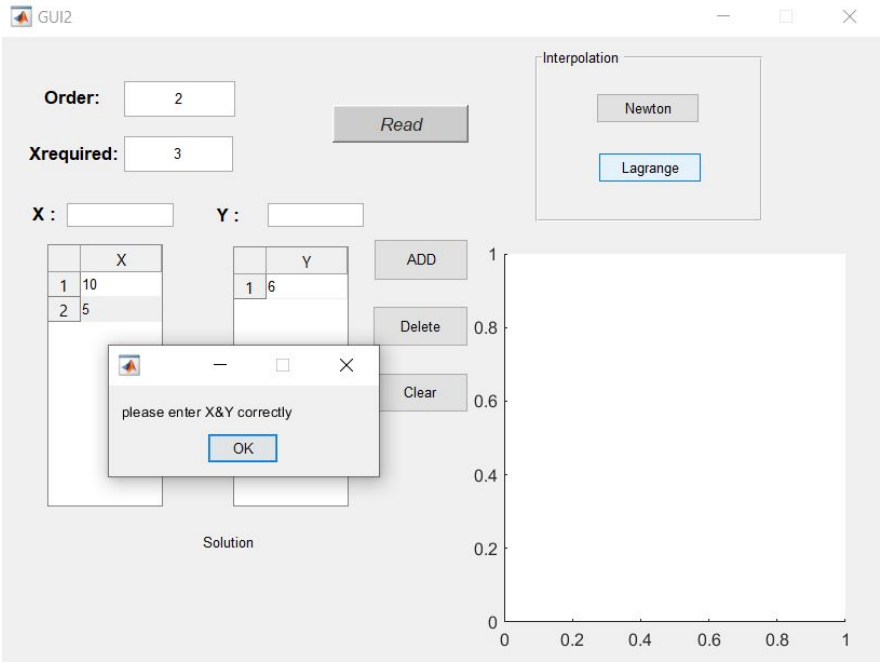
I. The user cannot enter any characters as order & Xrequired should be digits.



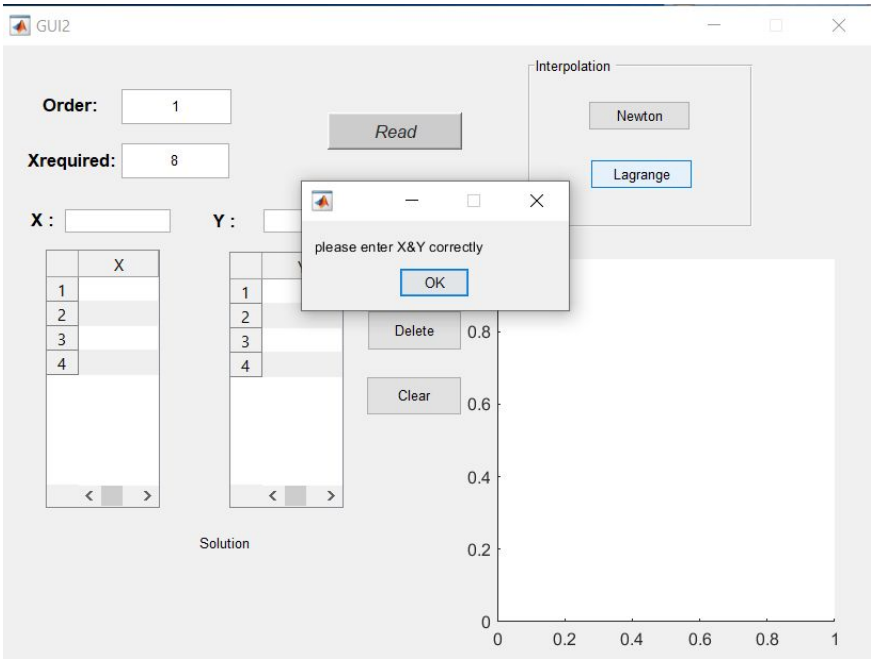
II. X&Y also should be digits.



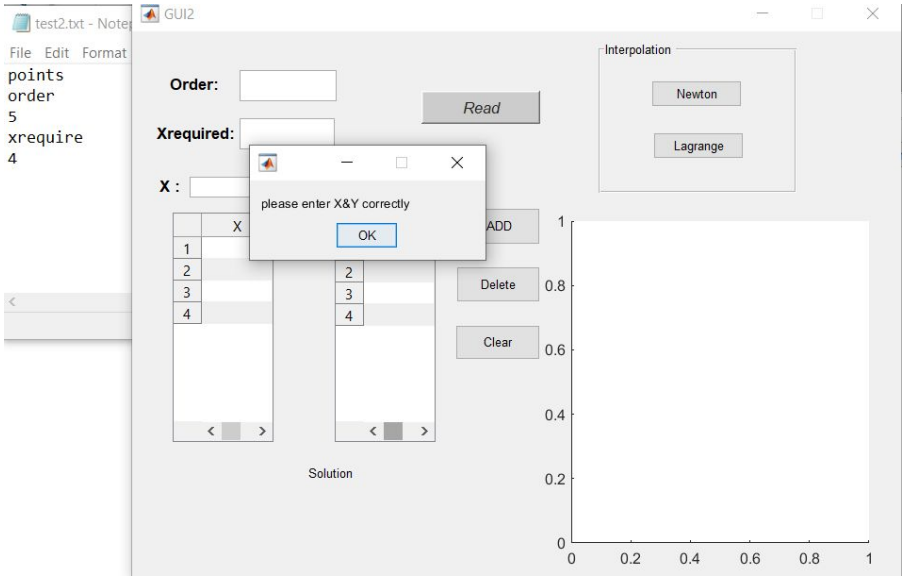
III. as the user enters points the length of X should equal the length of Y.



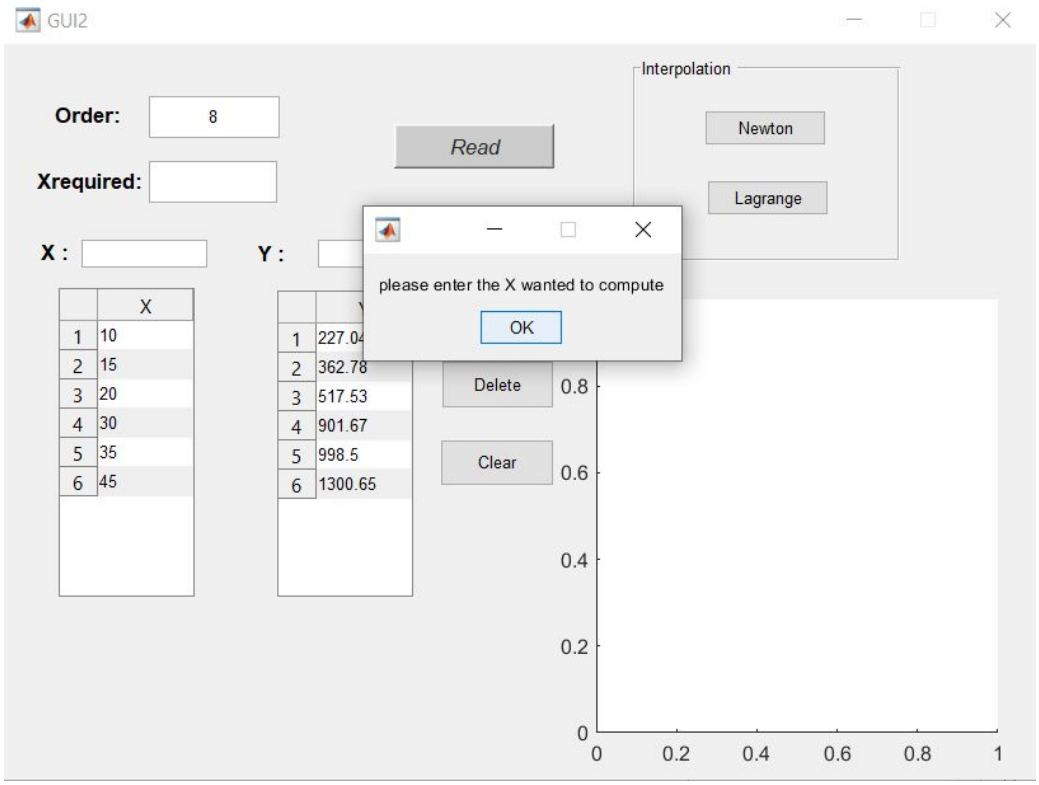
IV. X & Y cannot be empty.



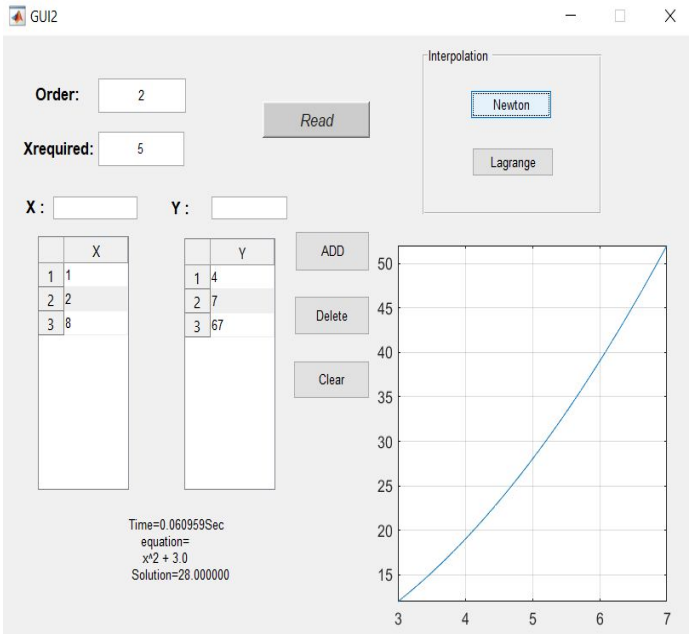
V. When the user writes in the file,X&Y cannot be empty.

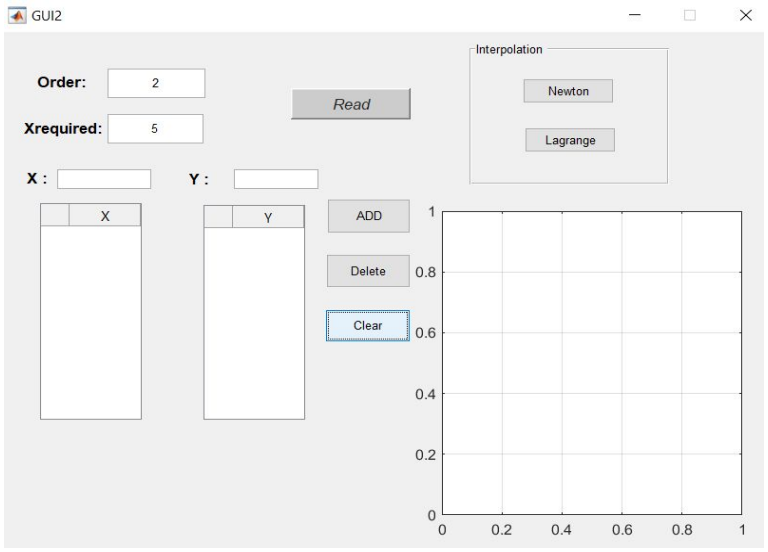


VI. X req & order should be supplied.

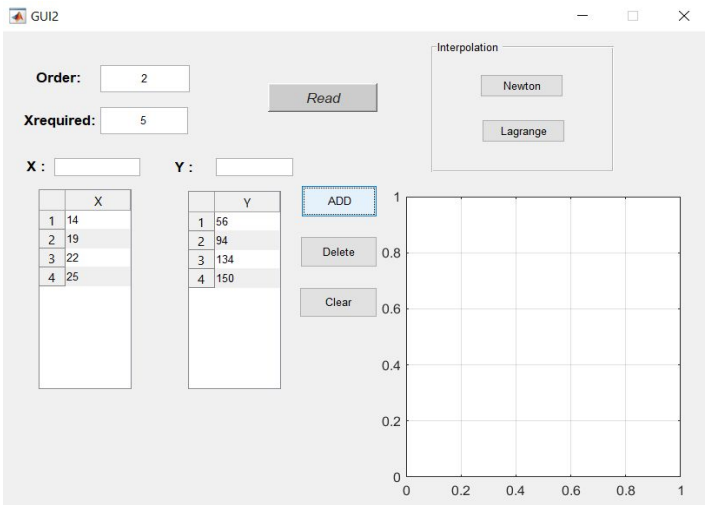


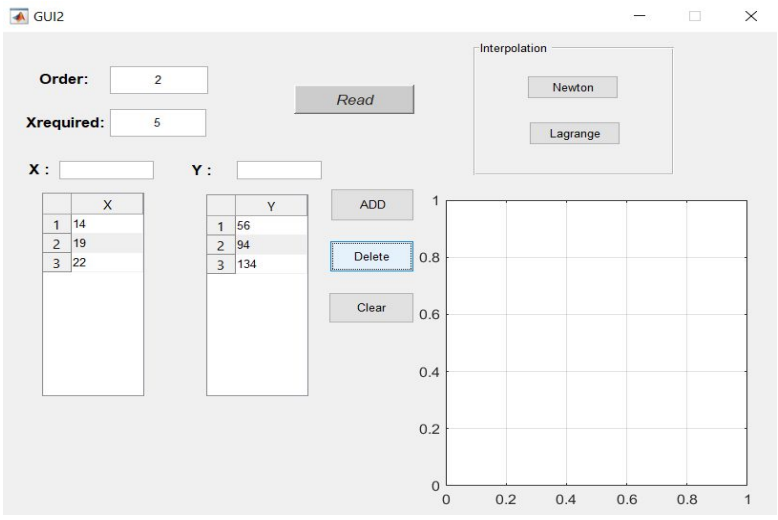
VII. Clear Button.



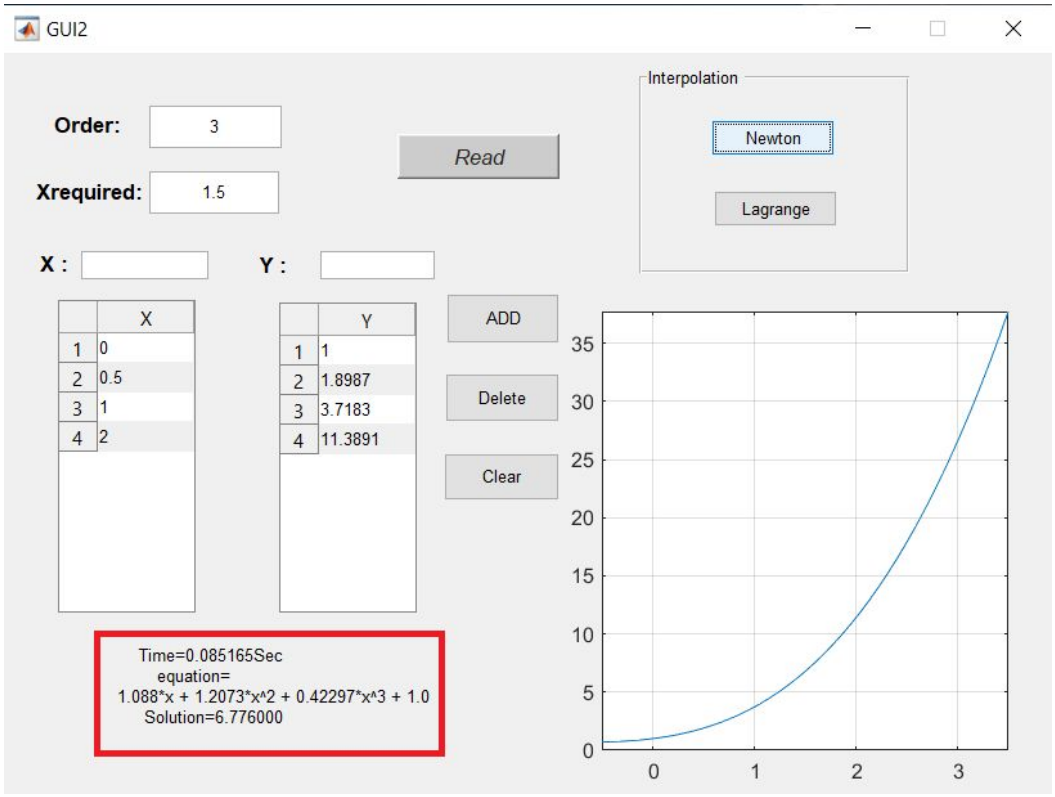


VIII. Delete Button.

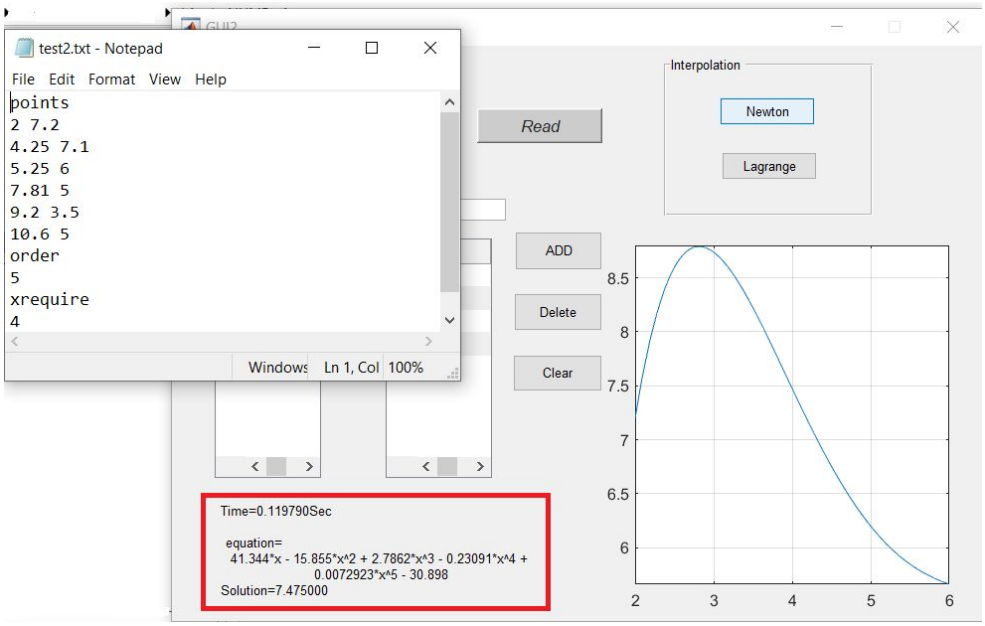




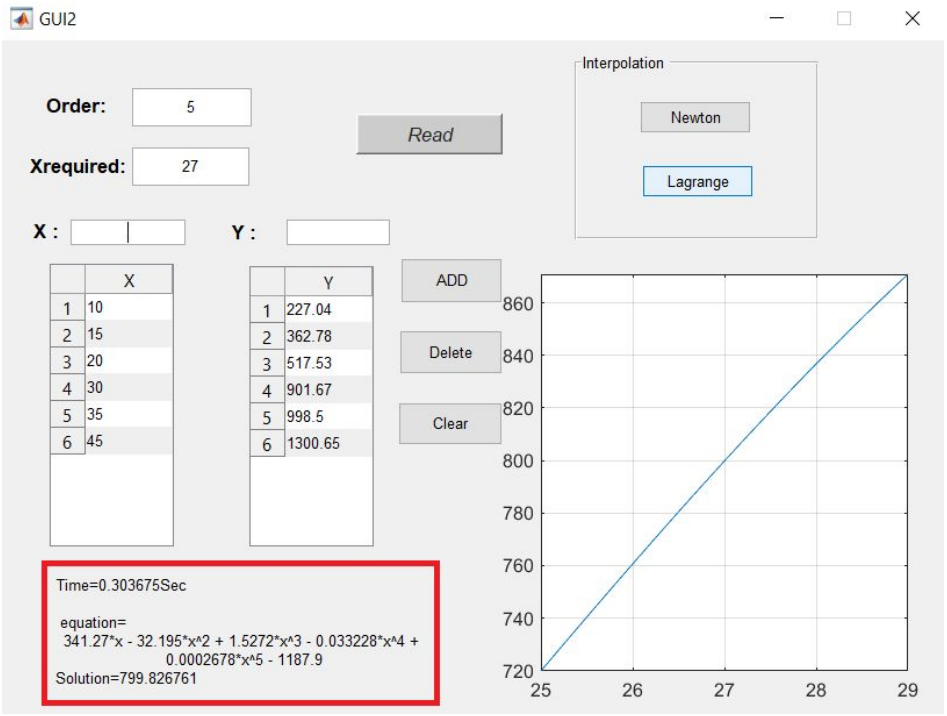
IX. Sample runs for newton method.



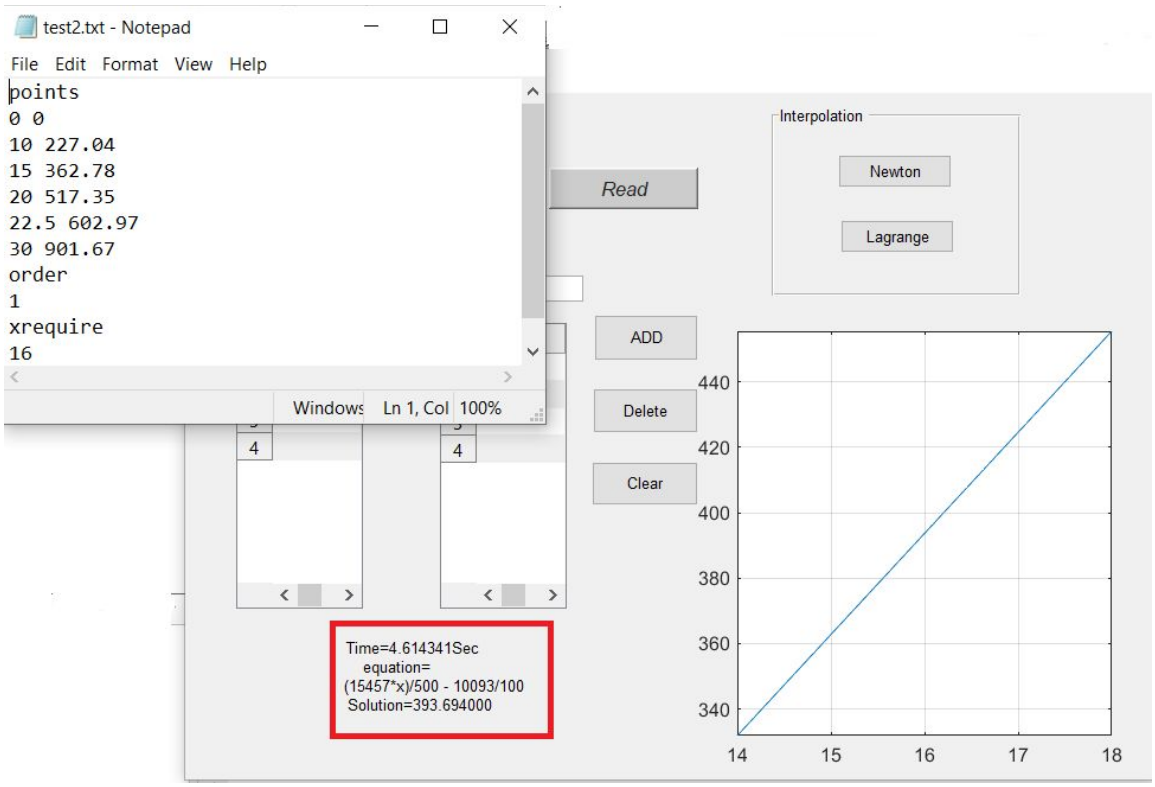
X. Sample runs for Newton method "reading from file".



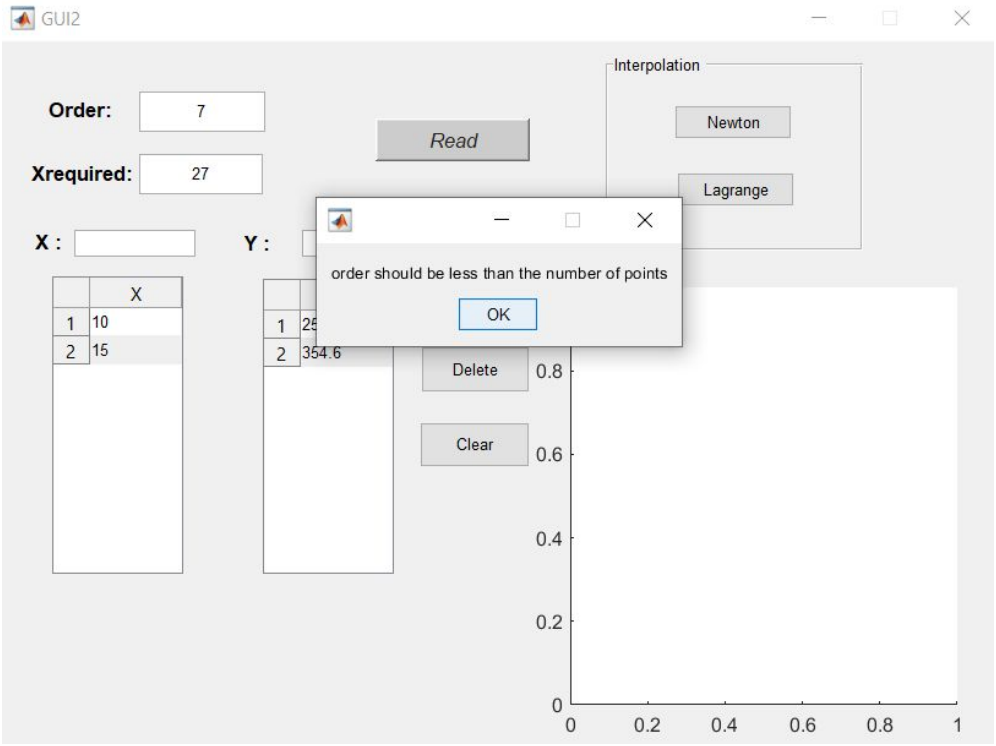
XI. Sample runs for Lagrange method.



XII. Sample runs for Lagrange method "reading from file".

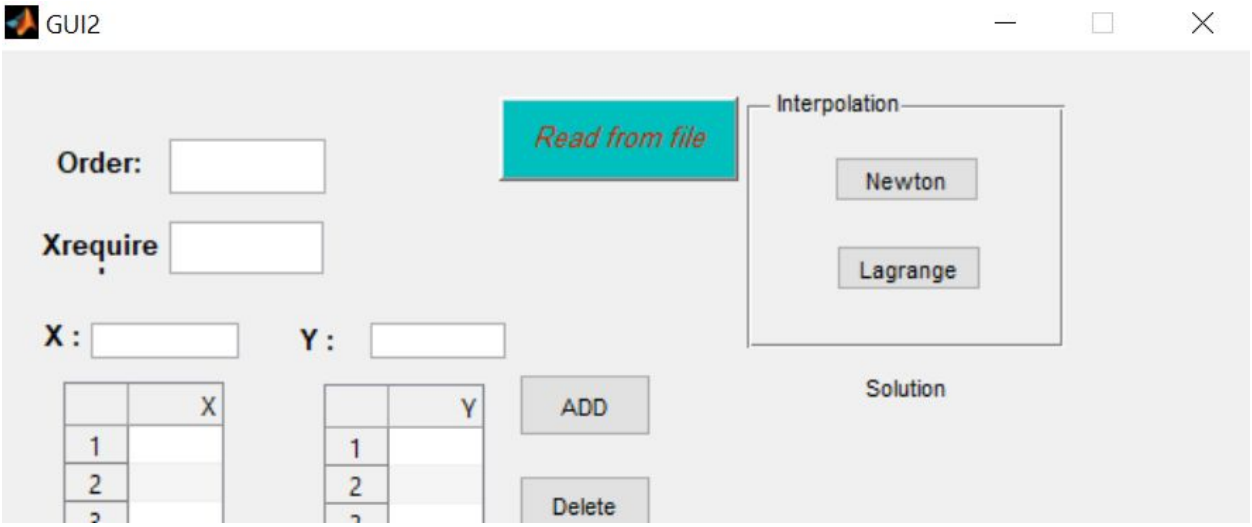


XIII. order should be less than the no. of points.

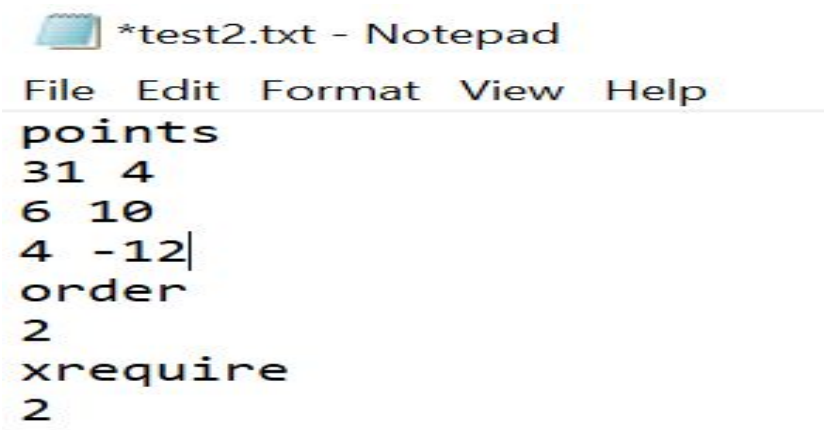


User Guide to read from file

- Input arguments can be read from a text file using the button (Read from file)



- Users should consider the order of arguments .
- Both methods have the same order of arguments.
- The input should be in the following format:



1. To add input points you write ‘ points ’ in a separate line,then put every point in a line in the format:

   X1 y1

   X2 y2 and so on.

2. To add the order you write ‘order ’ in a separate line,then put its value.
3. To add input X you write ‘ xrequire ’ in a separate line,then its value.