Scientific Computing

Project 1

Submitted by:

- Ahmed Hussien Sabry
- Mohammed Hassoubah
- Ramy ElKomy

Scientific Computing coursework 1

Summary

We implemented **Gauss Sidel[gaussseidel.cpp]** and **Gauss Elimination[gausselimination.cpp]** using **strategy design pattern** are they are implementation of the same functionality. They both solve a system of equation.

We used them in the interpolation and regression classes which implement the **Newton & Spline[interpolation.cpp]** , **Linear & Polynomial[regression.cpp**]

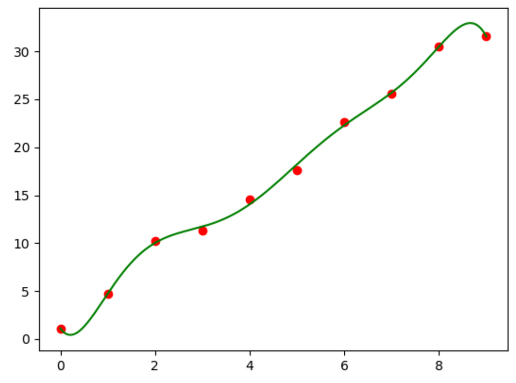
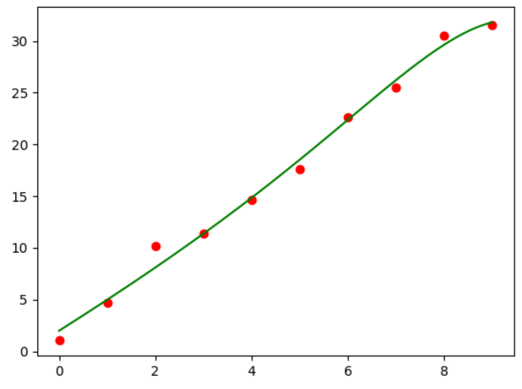
Our program use the **csvreader.cpp** in the **main.cpp** to read the given input data sets and call the functions of the system [Newton interpolation, Spline interpolation, Linear and Polynomial regression] then produce CSV's with the result usually the output Equation coefficients.

We have a python script called **plot.py** this python script is designed to loop on the output CSV's that carry the Equation Coefficients and plot them along with real points and the different requirements that you asked for In the coursework 1 pdf.

Part 1

Both methods (Gauss elimination with scaled partial pivoting AND Gauss-Seidel) was implemented and used in part 2 and part 3.

We noticed that in terms of **accuracy** the data output of the Gauss elimination is more accurate we noticed that in the data of the polynomial shown below. The coefficients generated by the **elimination** is the **image on the left** it fits all the points while the ones **on the right** is for the **seidel** and It's clear from the two images that the error in the seidel is more that the error in the elimination.

In terms of **criteria for convergence**

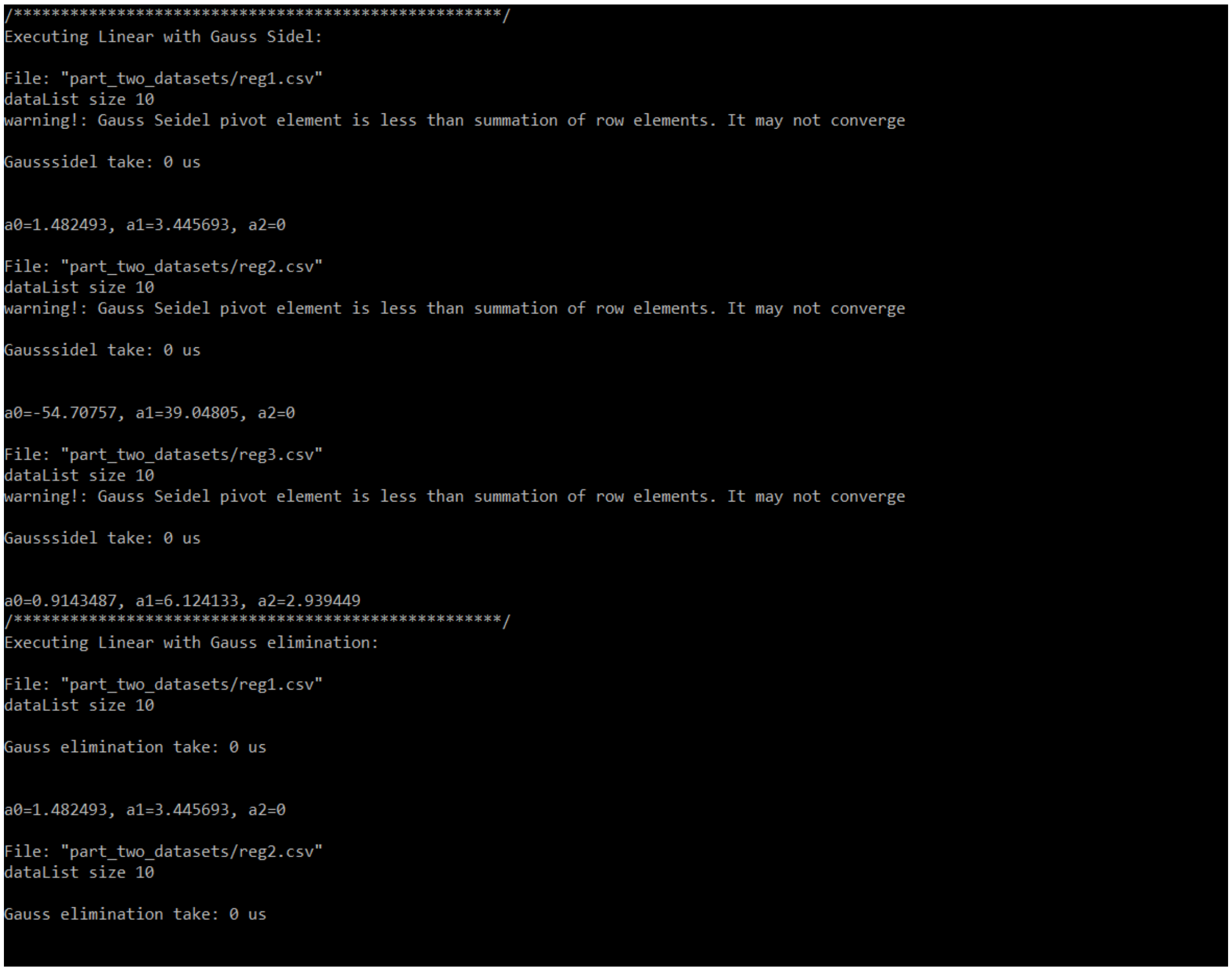
Gauss Elimination: always converges unless the system is singular or have multiple infinite number of solutions.

Gauss Seidel: It has a limitation that diagonal elements must not be equal zero, as in case of a diagonal element equals zero it will cause a division by zero. And as a conversion criterion the diagonal element should have a value greater that the sum of the absolute of the rest values in its row, if this criterion is respected it will converge faster but if now applicable it may take a lot of iteration to converge or it may not converge at all.

In terms of **computational cost**

We know that the **gauss seidel** needs less computational power if it' convergence are is met. We tried to measure the runtime behavior of both algorithms to report with numbers but we always got 0 us for both don't if this is because they really execute so fast as data is small.

We print the execution time of each time one of the algorithms execute on cmd.

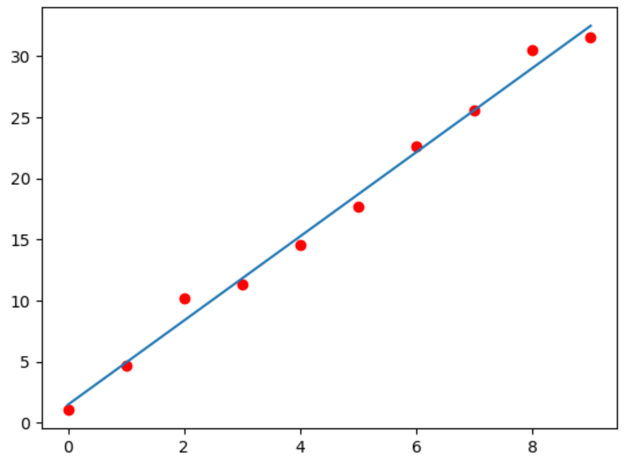


```
/****************************************************/
Executing Linear with Gauss Sidel:

File: "part_two_datasets/reg1.csv"
dataList size 10
warning!: Gauss Seidel pivot element is less than summation of row elements. It may not converge

Gausssidel take: 0 us


a0=1.482493, a1=3.445693, a2=0

File: "part_two_datasets/reg2.csv"
dataList size 10
warning!: Gauss Seidel pivot element is less than summation of row elements. It may not converge

Gausssidel take: 0 us


a0=-54.70757, a1=39.04805, a2=0

File: "part_two_datasets/reg3.csv"
dataList size 10
warning!: Gauss Seidel pivot element is less than summation of row elements. It may not converge

Gausssidel take: 0 us


a0=0.9143487, a1=6.124133, a2=2.939449
/****************************************************/
Executing Linear with Gauss elimination:

File: "part_two_datasets/reg1.csv"
dataList size 10

Gauss elimination take: 0 us


a0=1.482493, a1=3.445693, a2=0

File: "part_two_datasets/reg2.csv"
dataList size 10

Gauss elimination take: 0 us
```

Part 2

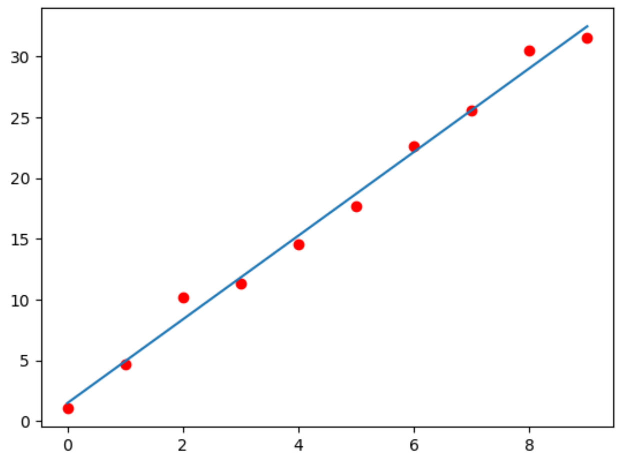
Polynomial and linear is implemented in code in one function that you can give an option to select between them, results is as come in the following sections:

Reg1 Linear using Gauss elimination:



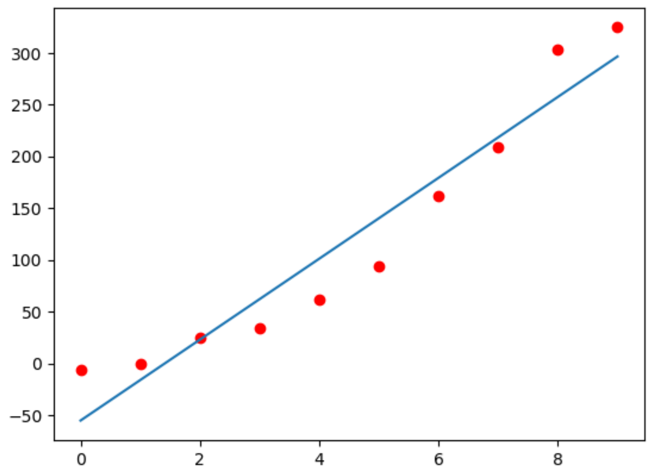
a0	a1
1.482493	3.445693

Reg1 Linear using Gauss Seidel:



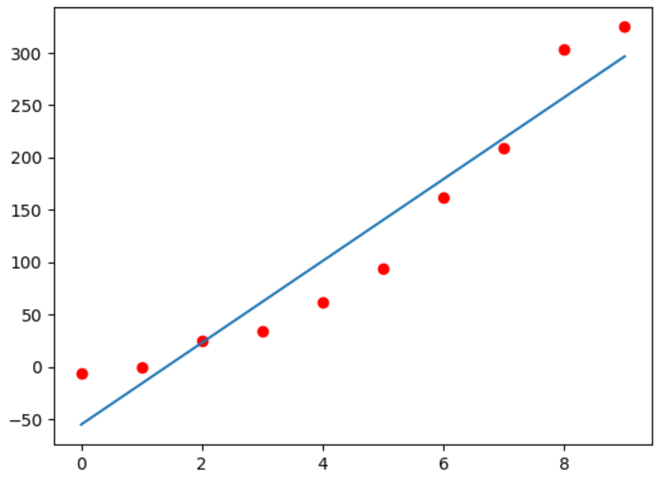
a0	a1
1.482493	3.445693

Reg2 Linear using Gauss elimination:



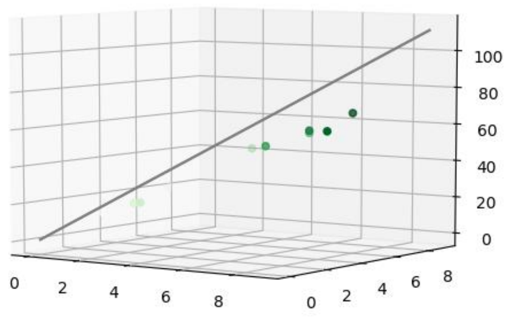
a0	a1
-54.7076	39.04805

Reg2 Linear using Gauss Seidel:



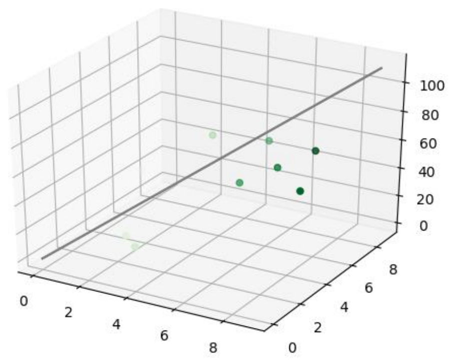
a0	a1
-54.7076	39.04805

Reg3 Linear using Gauss elimination:



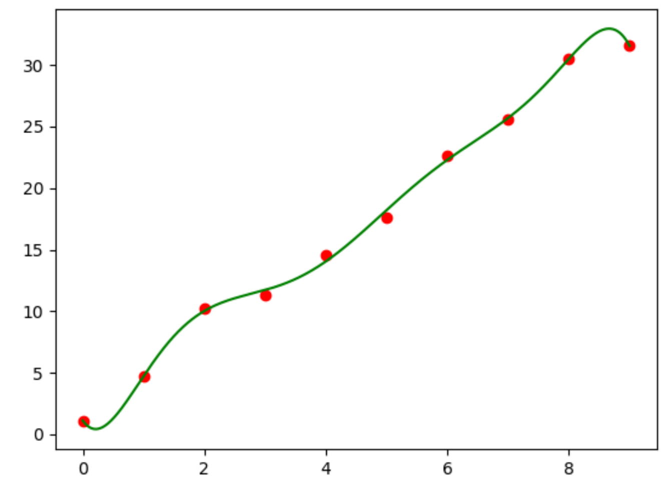
a0	a1	a2
0.914349	6.124133	2.939449

Reg3 Linear using Gauss Seidel:



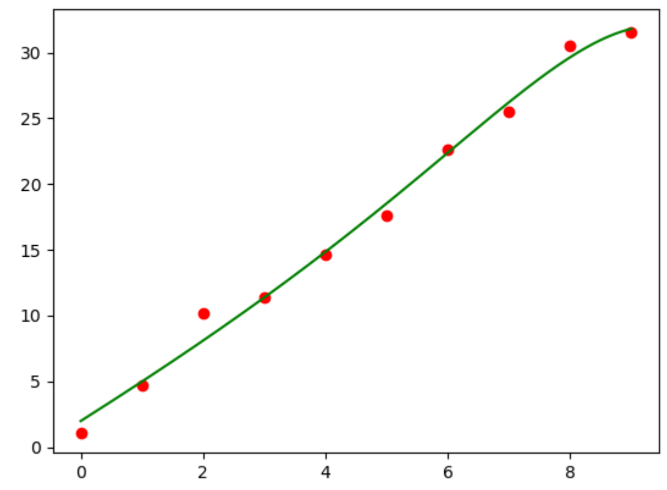
a0	a1	a2
0.914349	6.124133	2.939449

Reg1 polynomial using Gauss elimination:



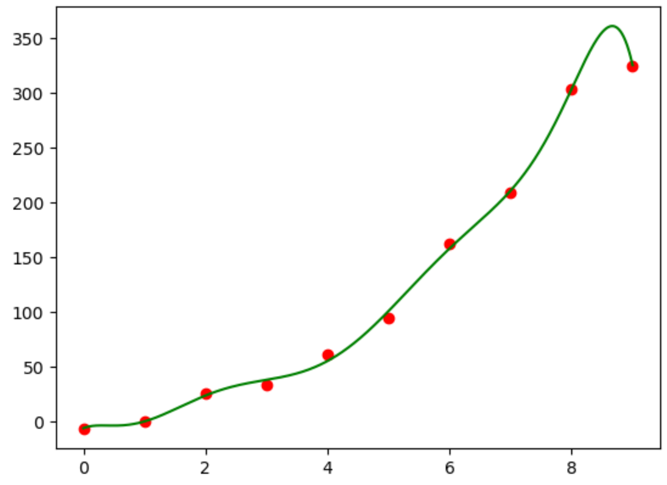
a0	a1	a2	a3	a4	a5	a6	a7	a8
1.117727	-7.00254	20.3679	-13.1669	3.980945	-0.61846	0.047623	-0.00139	-4.52E-06

Reg1 polynomial using Gauss Seidel:



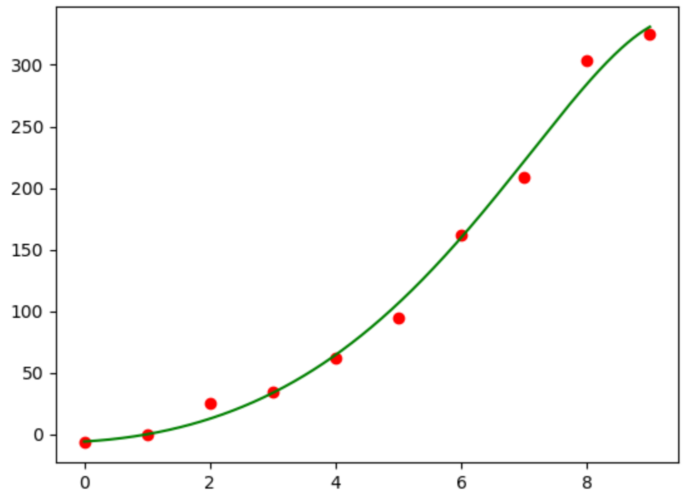
a0	a1	a2	a3	a4	a5	a6	a7	a8
1.987949	2.95079	0.041232	0.005257	0.000306	1.16E-06	-2.46E-06	-4.45E-07	-5.94E-08

Reg2 polynomial using Gauss elimination:



a0	a1	a2	a3	a4	a5	a6	a7	a8
-5.96001	22.88806	-73.0386	96.91521	-53.5537	15.03493	-2.2489	0.171034	-0.0052

Reg2 polynomial using Gauss Seidel:

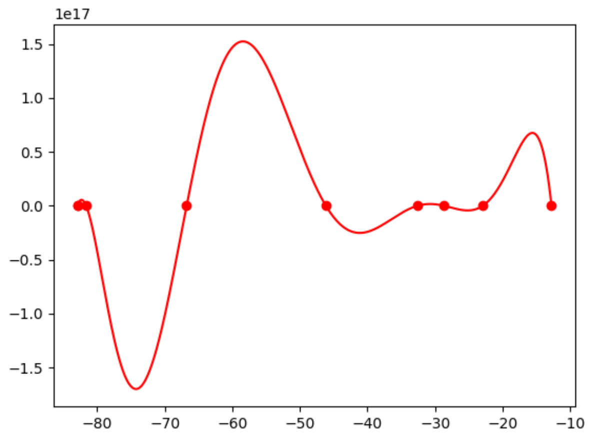


a0	a1	a2	a3	a4	a5	a6	a7	a8
-5.85988	2.989172	2.700233	0.200571	0.010301	0.000172	-4.96E-05	-9.96E-06	-1.37E-06

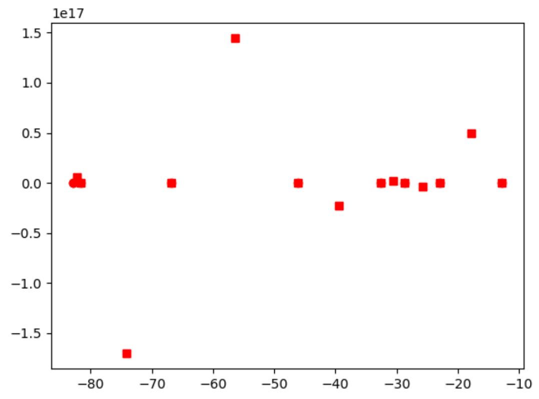
Part 3

SP1 Newton Interpolation:

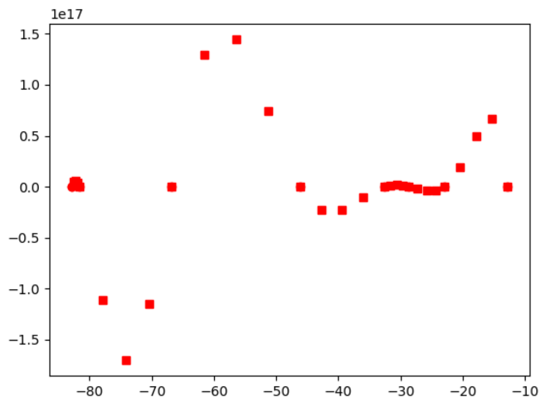
Curve:



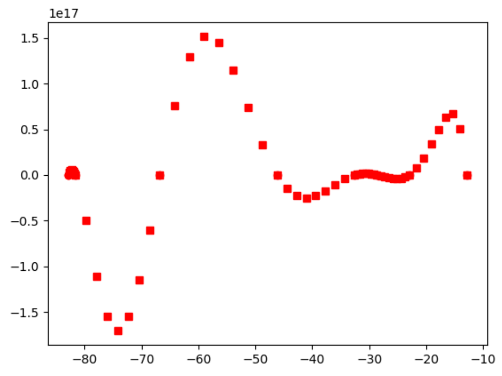
2X Points:



4X Points



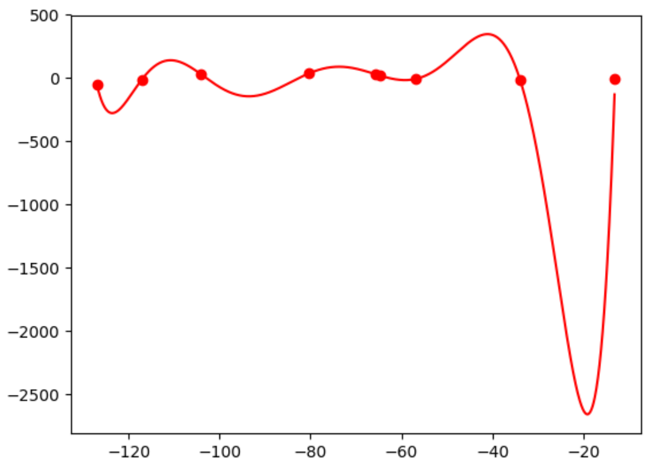
8X Points



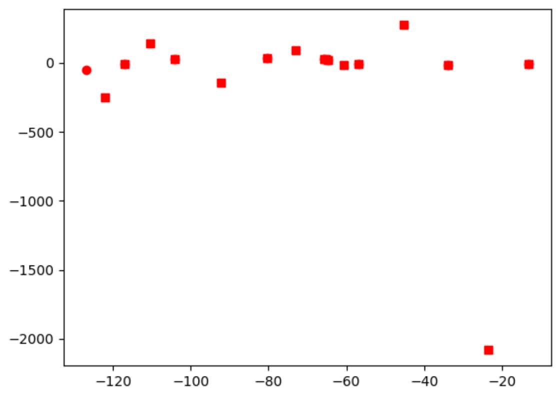
SP1 Spline Interpolation: Spline with this dataset generates singular matrix that could not be solved.

SP2 Newton Interpolation:

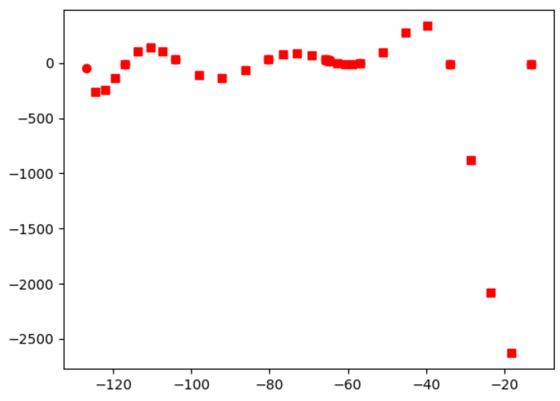
Curve:



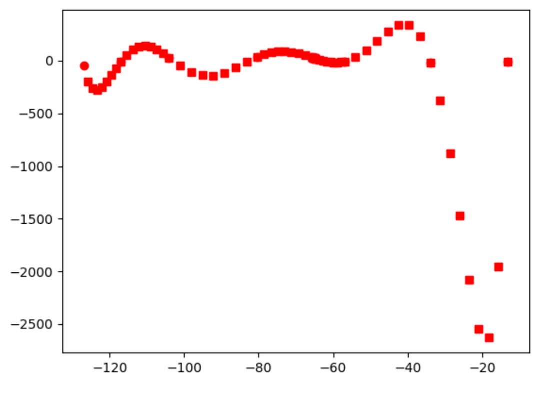
2XPoints



4X Points

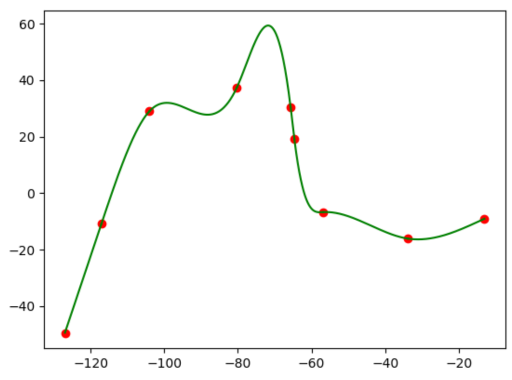


8X Points

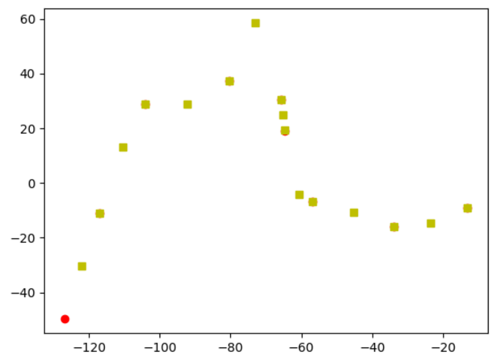


SP2 Spline Interpolation using Gauss Elemenation:

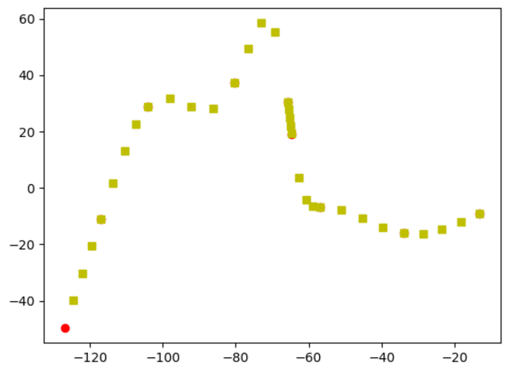
Curve:



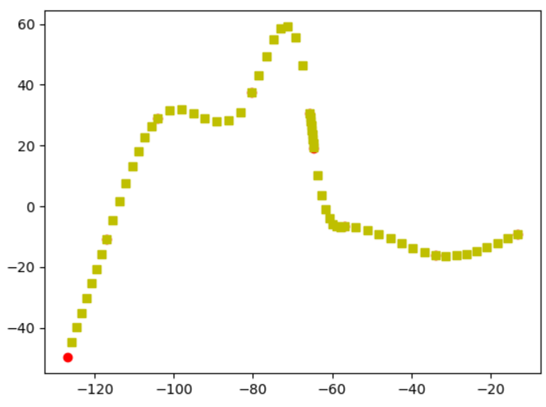
2X Points



4X Points



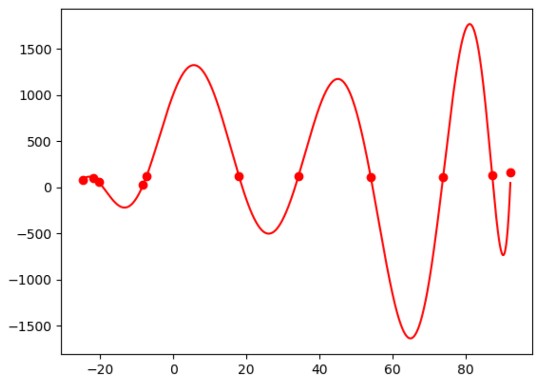
8X Points



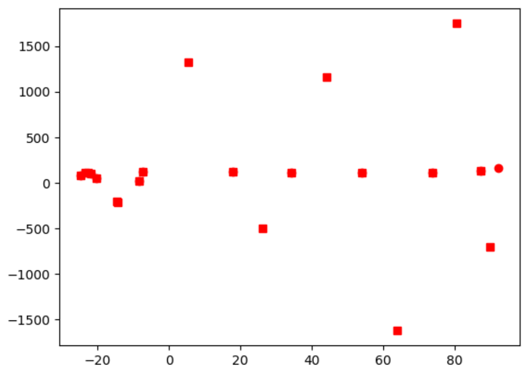
SP2 Spline Interpolation using Gauss Seidel does not converge.

SP3 Newton Interpolation:

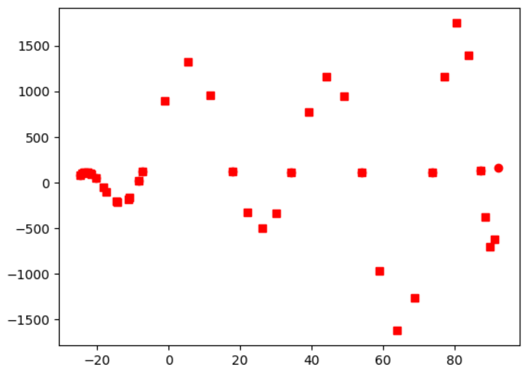
Curve



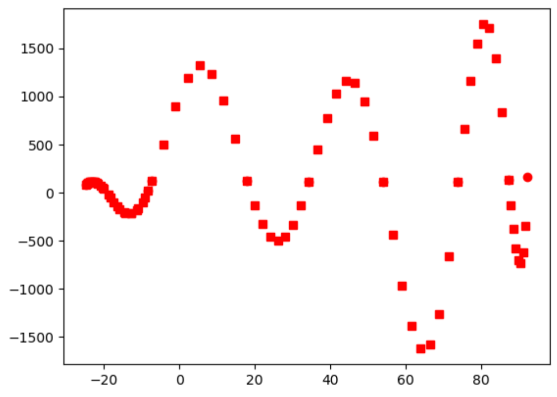
2X Points



4X Points

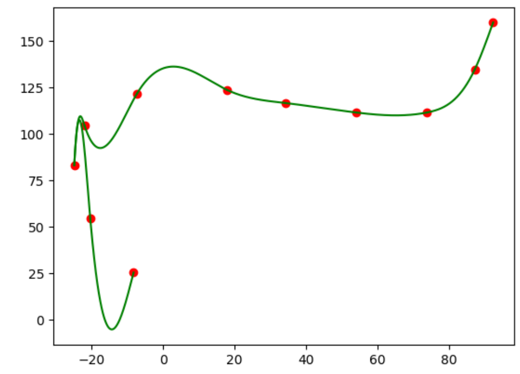


8 X Points

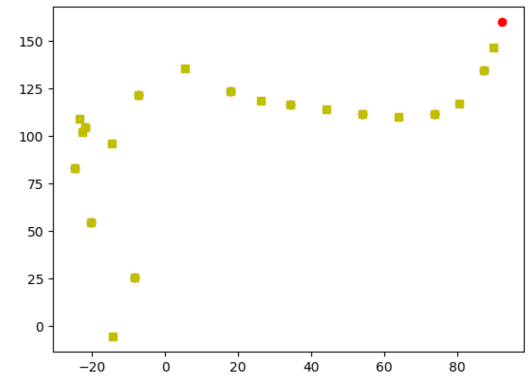


SP3 Spline Interpolation using Gauss Elemenation:

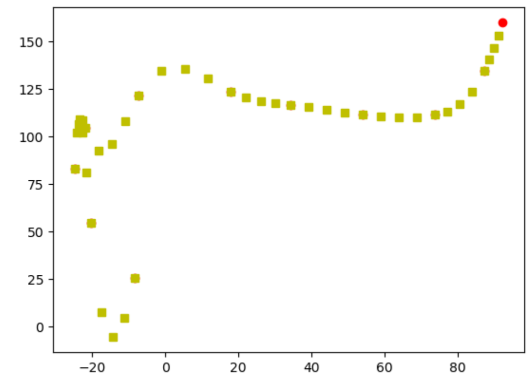
Curve



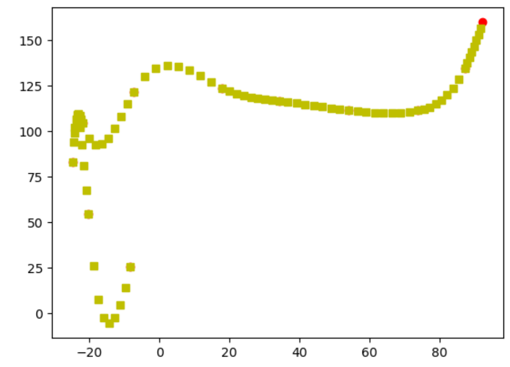
2X Points



4X Points



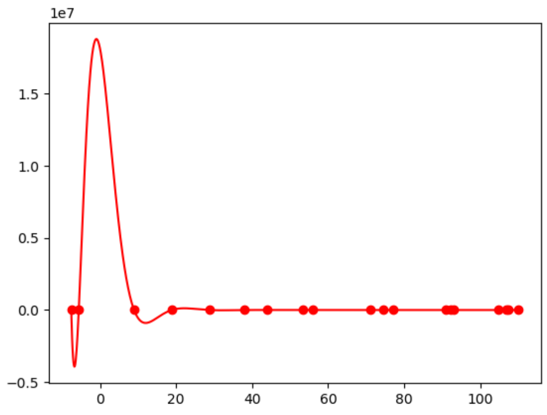
8X Points



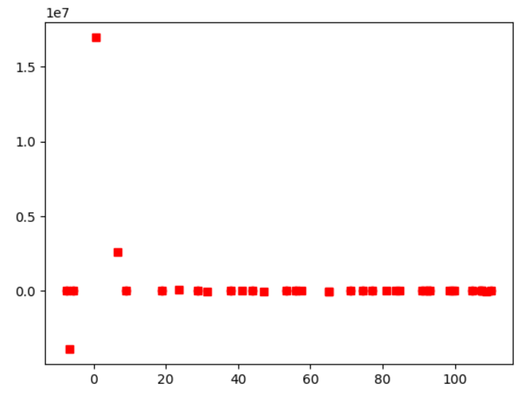
SP3 Spline Interpolation using Gauss Seidel **does not converge**.

SP4 Newton Interpolation:

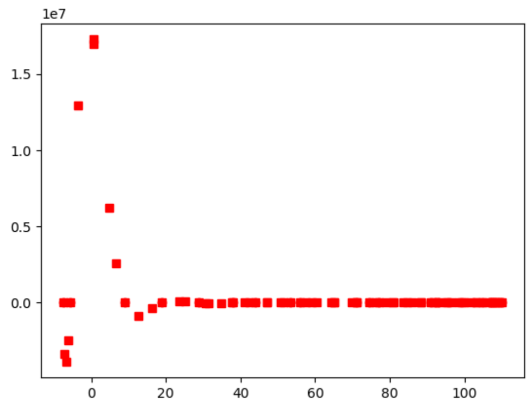
Curve



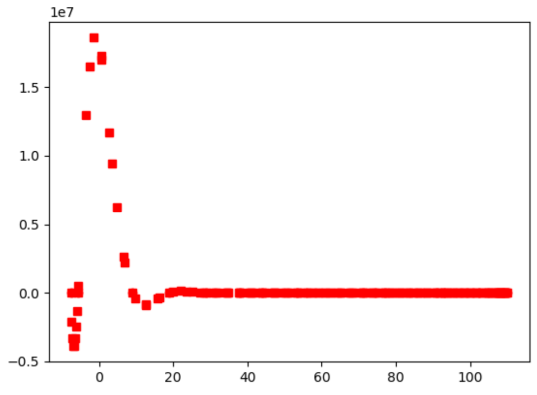
2X Points



4X Points

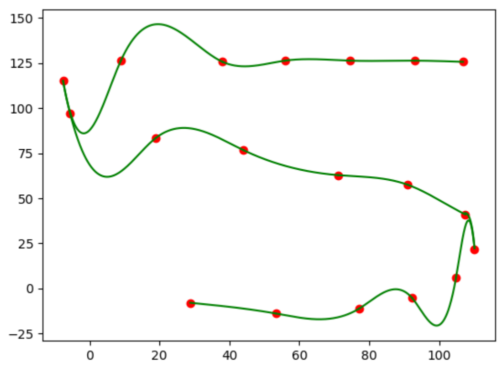


8X Points

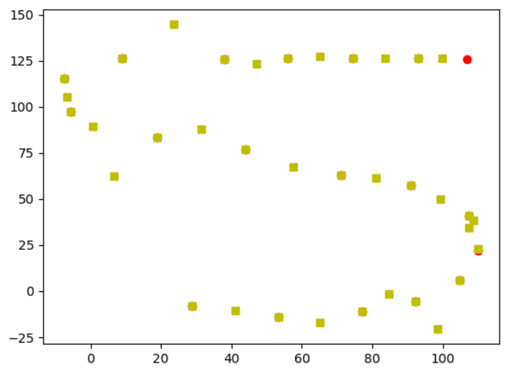


SP4 Spline Interpolation using Gauss Elemenation:

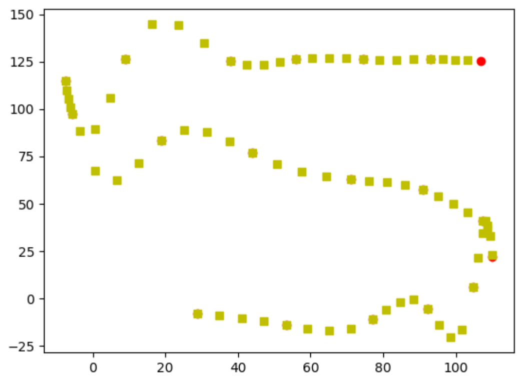
Curve



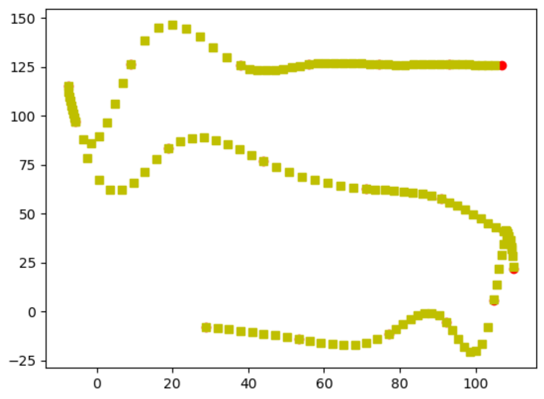
2X Points



4X Points



8X Points



SP4 Spline Interpolation using Gauss Seidel **does not converge**.