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**]

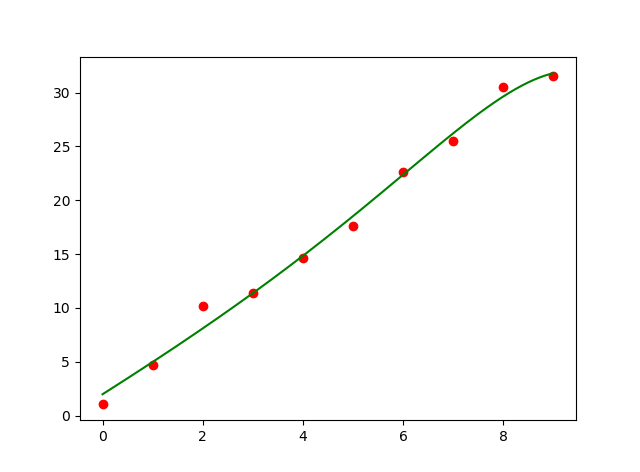
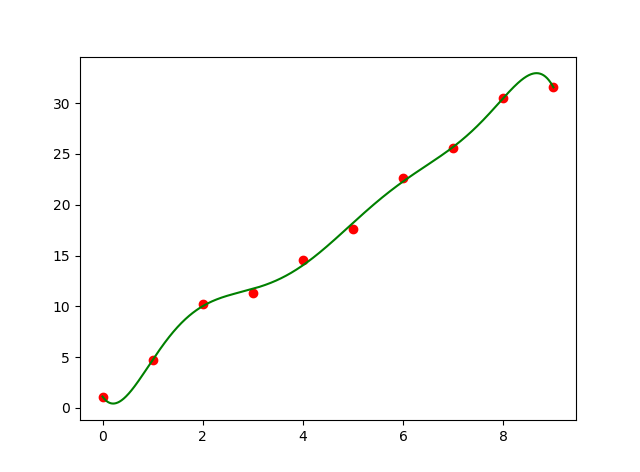
Our program use the **csvreader.cpp** in the **main.cpp** to read the given input data sets and call the functions of the system [Newton interpolation, Spline interpolation, Linear and Polynomial regression] then produce CSV’s with the result usually the output Equation coefficients.

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**

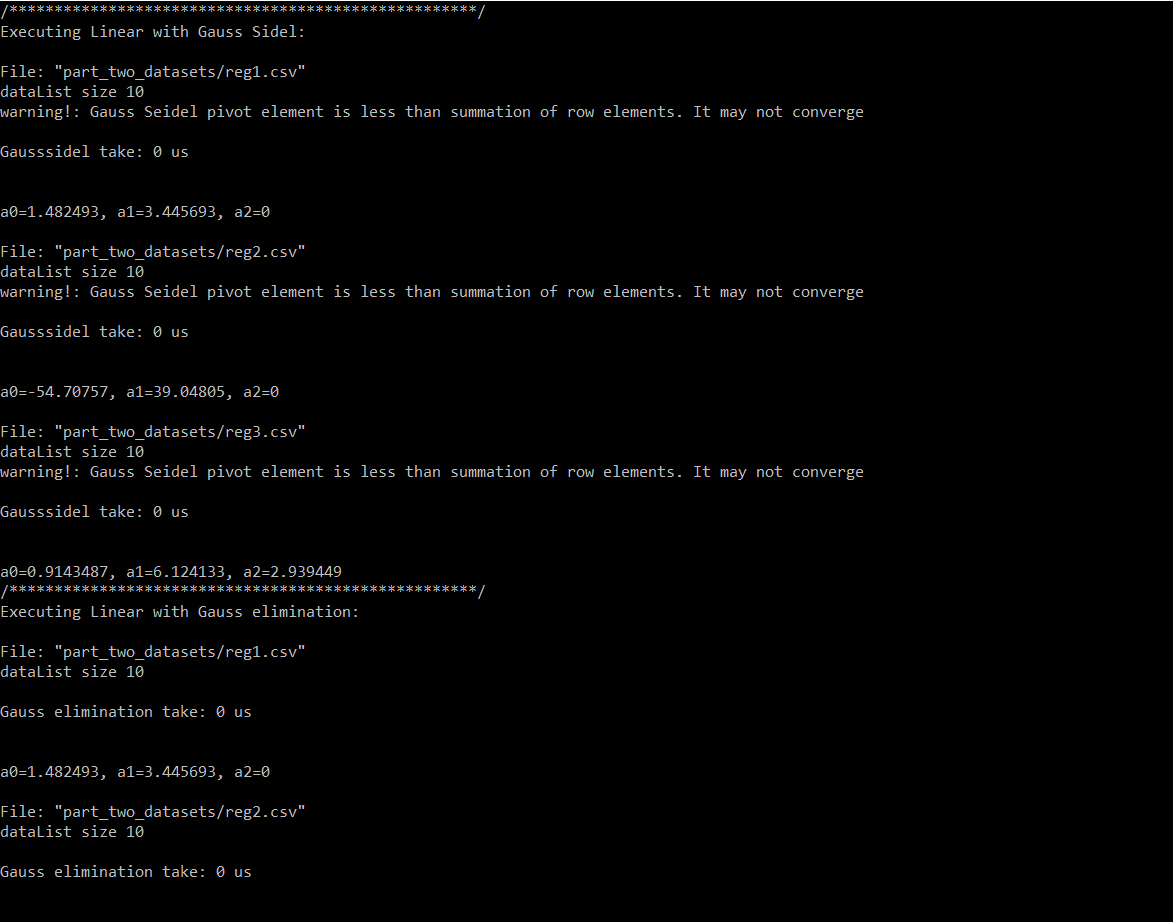
Elimination: always converge unless the system is singular or have multiple infinite number of solutions.

Seidel: It have a condition that diagonal elements must not be equal zero and that the elements in the diagonal must have a value bigger that the sum of the rest of the row. That condition is not 100% accurate as it might converge even if this is not true but what is sure is that it will converge faster if this condition is true.

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.

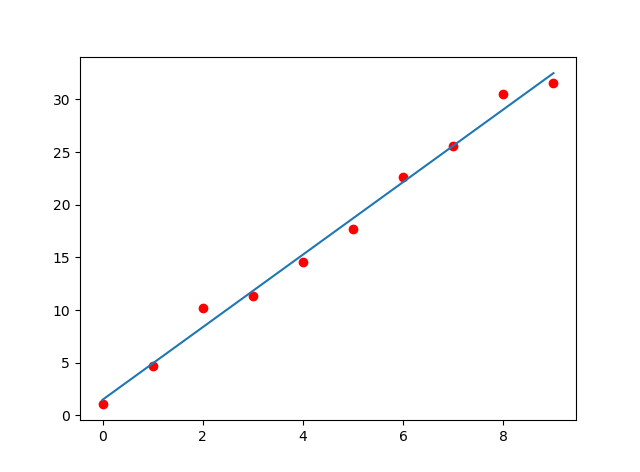


# Part 2

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:

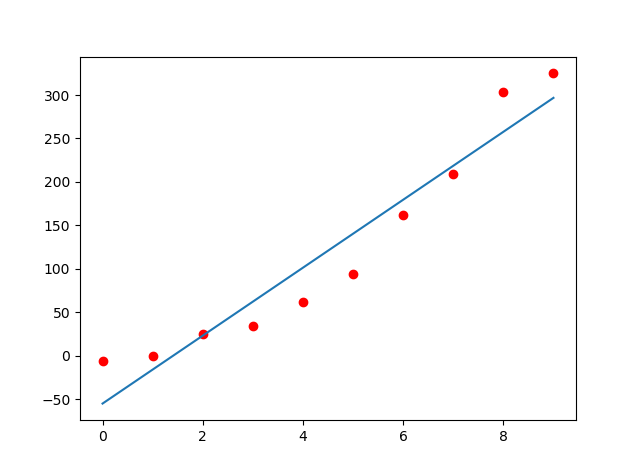
## Reg1 Linear using Gauss elimination:

## Reg1 Linear using Gauss Seidel:

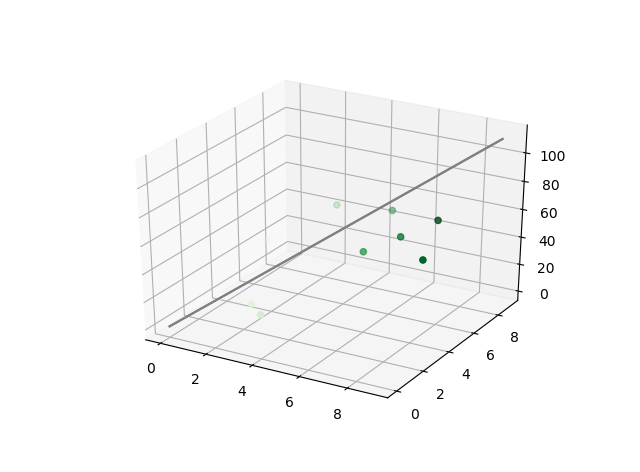
## Reg2 Linear using Gauss elimination:

## Reg2 Linear using Gauss Seidel:

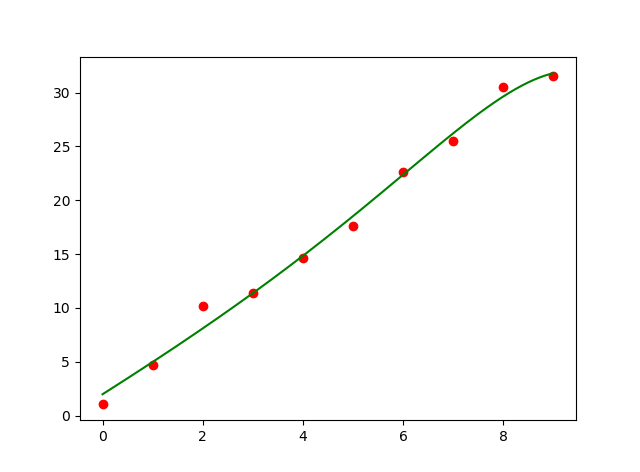
## Reg3 Linear using Gauss elimination:

## Reg3 Linear using Gauss Seidel:

## Reg1 polynomial using Gauss elimination:

## Reg1 polynomial using Gauss Seidel:

## Reg2 polynomial using Gauss elimination:

## Reg2 polynomial using Gauss Seidel:

