Jeffrey Rodriguez 110733867
AMS 326 Test 2
3/08/2018

T2-2

Introduction

The Runge function is defined as $f(x) = \frac{1}{1+25x^2}$. Partitioning the intervals [-1,0] and [1,0], we wish to fit two curves on these points to see if they will match with the original function's curve. The first of these lines is a 4th interpolation of the left interval, and the 2nd is a quadratic curve fit onto the right interval.

Method

This process was done in Python with assistance of the packages Numpy and Matplotlib.ply. First, the points in each interval are assigned to an array for the left and right x-values. Next, two y arrays were created to contain the values of the function at each x point.
To obtain the interpolation curve, we set $P(x_i) = y_i$, with $P(x_i)$ being a polynomial curve, and $0 \leq i \leq 4$. We write $P(x_i) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$. Then with 5 x values and 5 y values, we can write this in matrix form as the equation:

$$\begin{pmatrix} x_0^4 & x_0^3 & x_0^2 & x_0 & 1 \\ x_1^4 & x_1^3 & x_1^2 & x_1 & 1 \\ x_2^4 & x_2^3 & x_2^2 & x_2 & 1 \\ x_3^4 & x_3^3 & x_3^2 & x_3 & 1 \\ x_4^4 & x_4^3 & x_4^2 & x_4 & 1 \end{pmatrix} \begin{pmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Here, the $x_i,\ y_i$ values correspond to those of the interval [-1,0]. We can then solve for a by setting $a = X^{-1}Y$.
To obtain a quadratic curve, we set $Y_i = b_0 + b_1x_i + b_2x_i^2 + \epsilon_i$, where $x_i,\ y_i$ now corresponds to the right interval. We can then express this as $\sum_{i=0}^{4} \epsilon_i^2 = \sum_{i=0}^{4}(Y_i - (b_0 + b_1x_i + b_2x_i^2))^2$. To solve for the b values, we take the partial derivatives of ϵ^2 with respect to each b_i, set them equal to 0. With three equations and three unknown constants to solve for, we can express this as the following matrix:

$$\begin{pmatrix} 5 & \sum_{i=0}^{4} x_i & \sum_{i=0}^{4} x_i^2 \\ \sum_{i=0}^{4} x_i & \sum_{i=0}^{4} x_i^2 & \sum_{i=0}^{4} x_i^3 \\ \sum_{i=0}^{4} x_i^2 & \sum_{i=0}^{4} x_i^3 & \sum_{i=0}^{4} x_i^4 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{4} y_i \\ \sum_{i=0}^{4} y_ix_i \\ \sum_{i=0}^{4} y_ix_i^2 \end{pmatrix}$$

As with the interpolation polynomial matrix, we invert this above matrix and multiply it by the y vector to get the regression coefficients.
The matrix inversions and multiplication computations were performed via Python's numpy package.

Results

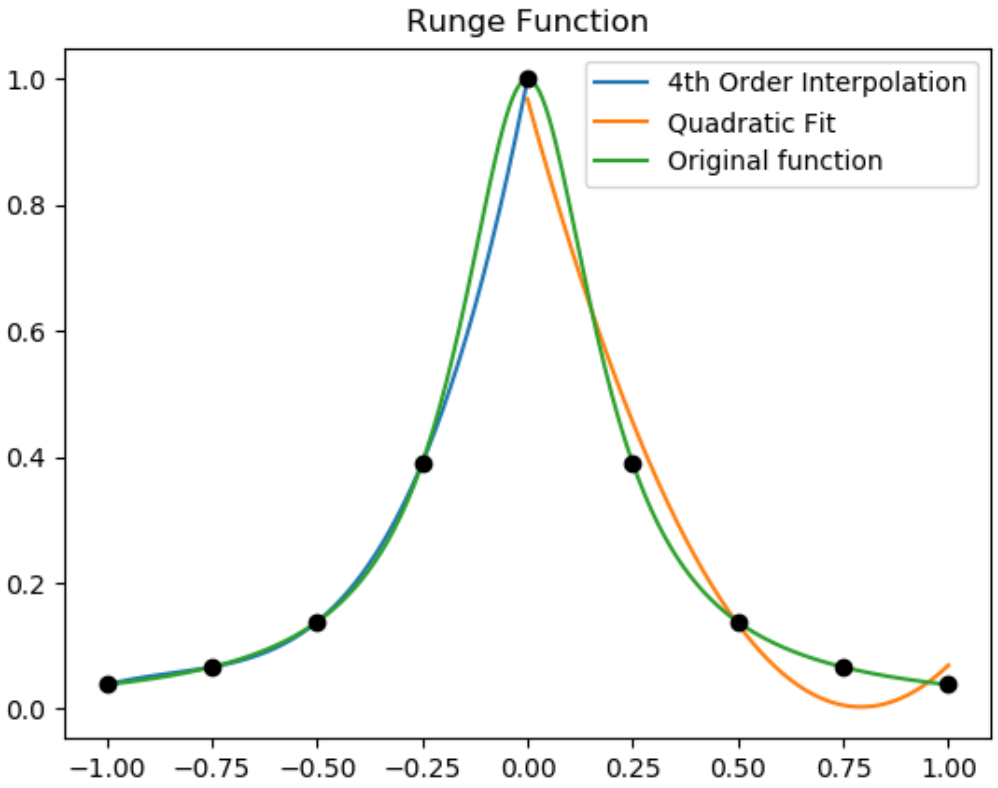
We find the interpolation curve for the left points to be defined by the function

$$f_l(x) = 0.42146101x^4 + 2.51668596x^3 + 4.56267112x^2 + 3.42898463x + 1$$

The quadratic curve for the right points is defined by the function

$$f_r(x) = 1.53648807x^2 - 2.43526039x + 0.96805247$$

Plotting these two lines, as well as the line for the function itself gives the following plot.



Conclusion

We see that the curves of the interpolation and original function are initially very close together, until after x is in the range of (-0.25,0] after which the interpolated curve becomes more straight as it approaches 1. The quadratic curve does not follow the original function as closely as the interpolated plot does, and also changes in direction after the original function. Due to the low amount of data, the code runs fast and can be seen in the attached file '326T2.py'.