Project 4 Write-Up

I will use the terms Phase 1, 2, and 3 in this write up. Phase 1 will be the Newton polynomial interpolation, Phase 2 will be the step function, and Phase 3 will be the Application.

In Phase 1, I was given the function polyval\_lagrange.m and the script lagrange\_test.m to visually see how the LaGrange series was coded in MATLAB. I then created the script newton\_test.m which was used to show a test over 2 other functions that I wrote which together determined the Newton interpolated polynomial for a given set of data. These two other functions are called newton\_eval.m and newton\_construct.m. I was asked to interpolate the polynomial for data points which were given by the function e^(x^2) from x = 1 to -1 with 11 points in between (intervals of .2). I initially printed out the x values with their corresponding f(x) values for this function for all x values. The f(x) values varied between 1 and e. I then set this distance equal to dx, and a z vector equal to the midpoints of all the x values (therefore giving it a length of n-1). Then I found the b values from newton\_construct.m by passing it the x and f(x) vectors. Then inside of a loop which looped from I = 1 to n-1 I called newton\_eval.m with the parameters of the b vector, x vector, length of x, and the particular z midpoint value that is being interpolated, and set the result to p. Before the loop repeats, I put the value p inside of a vector pz which will hold all of the interpolated function values for the z values. I then display all of this information out to the screen, along with an error calculator which is created by taking the absolute value of pz and then subtracting fz (true function values evaluated at all points of z) from it. The resulting error for all z values was very small (no greater than 1.26865 \* 10^-5).

In Phase 2, I evaluated 2 interpolated polynomials via the LaGrange method and 2 cubic splines of the step function and compared all 4 of them with the true step function, graphically. I initially created x and y vectors both with a length of 201 values ranging from -1 to 1 (interval of .01). Then for the 2 interpolated polynomials, I created vectors x2 and fx, which were evaluated at 8 points on the step function, and x3 and fx2, which were evaluated at 12 points on the step function. I found the pz vectors for both the 8 and 12 point vectors and evaluated them both for 201 points via the LaGrange method just like I did in Phase 1, but without the use of midpoints. I then plotted these two graphs on top of the true step function in a 2x2 matrix plot of subplots. These two graphs make up the top two graphs. Then I did the same exact thing, but instead of the LaGrange method, I used cubic splines (MATLAB spline command) to interpolate the step function at 8 and 12 points. I called the spline function for both pairs of vectors and used the exact same parameters as I did with the LaGrange method, except instead of looping through all the original x values I just took the loop out and put in the entire x vector, and then I set pp and pp2 equal to the spline function values. I then graphed these two interpolated functions on top of the step function in the same 2x2 plot, and these two graphs make up the bottom two graphs.

Since the step function has a giant gap in f(x) values as x approaches 0, all 4 of the interpolated polynomials have a much larger error with regard to continuous functions and their possible interpolations. There are several points on the plot which do not give accurate results that are in sync with the true step function, and this is because the step function is discontinuous. The discontinuous-ness of the step function is the reason why both the LaGrange interpolation and the spline interpolation create functions which fluctuate above and below the values of the step function.

In Phase 3, I used both LaGrange interpolations and spline interpolations to answer certain questions based on 3 data files that I was given: dallas.txt (average temperatures in Dallas for each month, with January repeated), historical.txt (yearly average global temperate over the last 20,000 years), and global.txt (average global temperatures from 1880 through 2009), which I had to change to gglobal.txt because “global” is a keyword in MATLAB. For all of the questions, I iterated through the length of the file and set an x vector equal to all of the x values of the data and an fx vector equal to all of the y values of the data. For all of the parameters of all of my function calls, I used the applicable vector x, the vector fx, and the point at which I am asking for the interpolated value.

For part (i), when using the LaGrange interpolation, I got that the average temperature on my birthday in Dallas is 84.5380 degrees Fahrenheit. When I used spline interpolation, I got 84.5563 degrees Fahrenheit. These were very similar, but the other questions had very different answers between the two interpolations.

For part (ii), when using the LaGrange interpolation, I got that the average global temperature in the year Julius Caesar was assassinated (44 BC) was 98,572 degrees Fahrenheit. When I used spline interpolation, I got 54.1636 degrees Fahrenheit.

For part (iii), when using the LaGrange interpolation, I got that the average global temperature around the time the mastodon went extinct was 4.8844 \* 10^39 degrees Fahrenheit. When I used spline interpolation, I got 57.2808 degrees Fahrenheit.

For part (iv-a), when using the LaGrange interpolation, I got that the average global temperature in the year 2015 would be 1.5336 \* 10^44 degrees Fahrenheit. When I used spline interpolation, I got 173.3472 degrees Fahrenheit. For part (iv-b), when using the LaGrange interpolation, I got that the average global temperature in the year 2025 would be -1.9052 \* 10^52 degrees Fahrenheit. When I used spline interpolation, I got 1603.1 degrees Fahrenheit. For part (iv-c), when using the LaGrange interpolation, I got that the average global temperature in the year 2050 would be -1.2286 \* 10^75 degrees Fahrenheit. When I used spline interpolation, I got 22,604 degrees Fahrenheit.