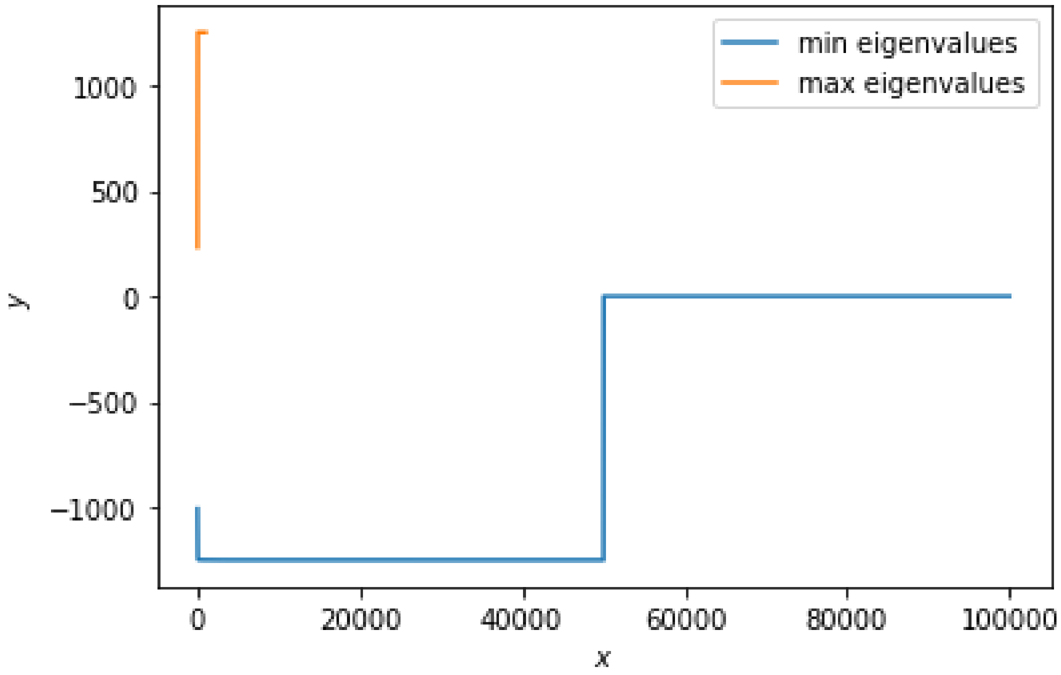
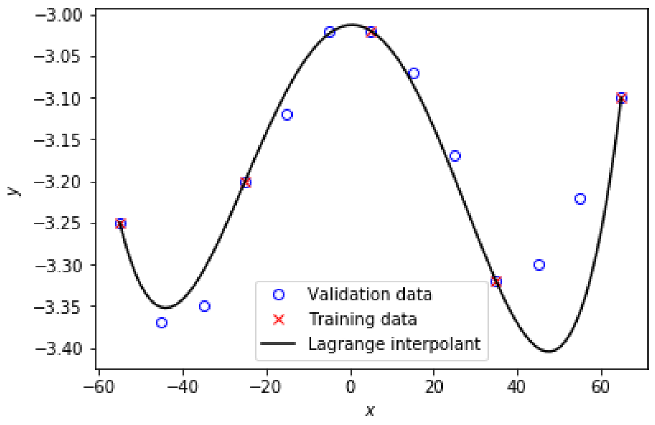
Elom Dumenyo
Assignment 2
ENM 360 – Perdikaris

1. The power method was used to approximate first the maximum eigenvalue (λ_{max}) of a positive definite symmetric matrix A, and then the minimum eigenvalue (λ_{min}) of that matrix. First, 1000 power iterations were computed to converge on λ_{max}. Then a delta shift was performed on $B = A - \lambda_{max}$, power iterations were computed on B to find λ_{min-B}, and a reverse shift had to performed (resulting in $\lambda_{min} = \lambda_{max} + \lambda_{min-B}$). While converging on the max eigenvalue happened in under 1000 iterations, the correct value for the min eigenvalue was not converged to even after 100,000 iterations – this may be because the min eigenvalue is so small.

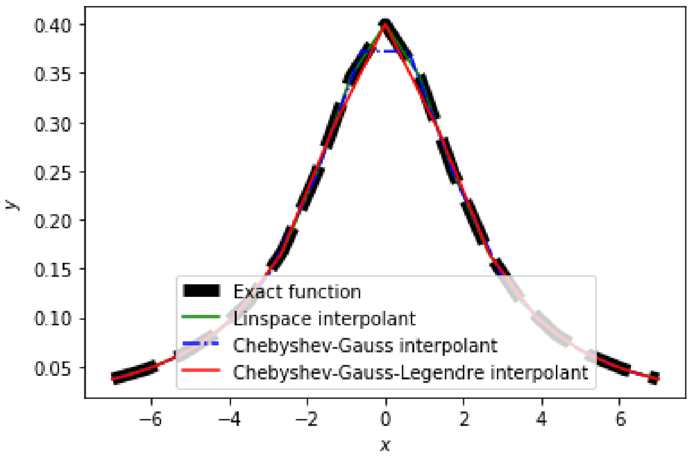


2. Using the training data and using the calculations from class, I fit a Lagrange polynomial curve to the training data. Subsequently, I plotted both the validation data and training data:

   The fit is satisfactory for the validation points in center of the graph, but it is poor for the points further from $x = 0$. This is Runge's phenomenon, caused due to the overfitting of the Lagrange polynomial to the training data.

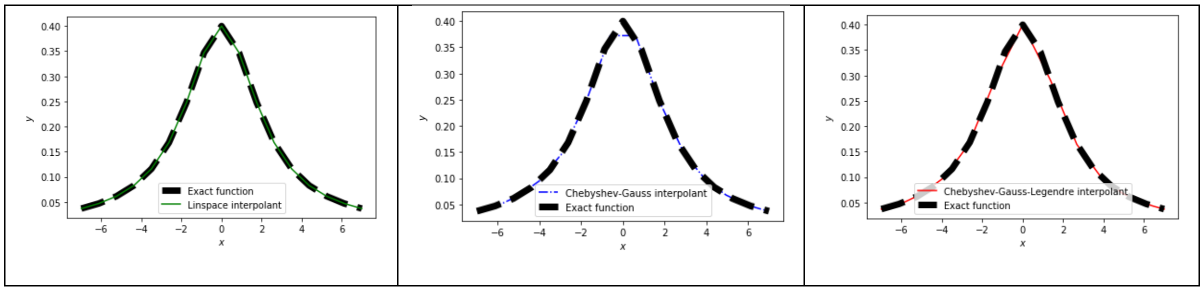


3. Uses the Lagrange interpolation to approximate the function $f(x) = 2/(5 + x^2)$ in the interval $x \in [-7,7]$. This interpolation is performed with linearly spaced nodes, then with Chebyshev-Gauss-Legendre nodes, and the with Chebyshev-Gauss nodes: the results of each are plotted (along with the graph of f).



   The error of each method is subsequently calculated, and the program prints out the results, revealing the most accurate interpolation method.

   The best way to use this is to compare the plot of the exact function to the plot of each interpolation method --- individually. Upon doing so, it becomes clear that **the regular, linearly spaced method** is the best.



4. Using the central finite differences approach to approximate the derivative of $ifunc$, I have found the convergence follows the sqrt function, meaning that increasing N creates

much better derivative approximations at first, and after some time, the effect of increasing the number of equidistant points is minimal. The plot of this is below.

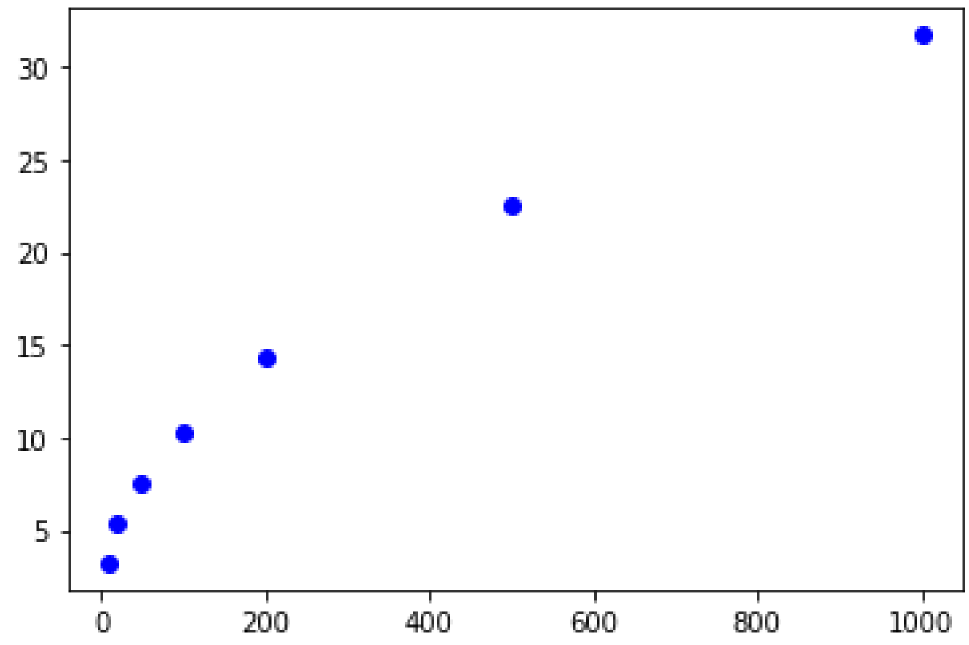
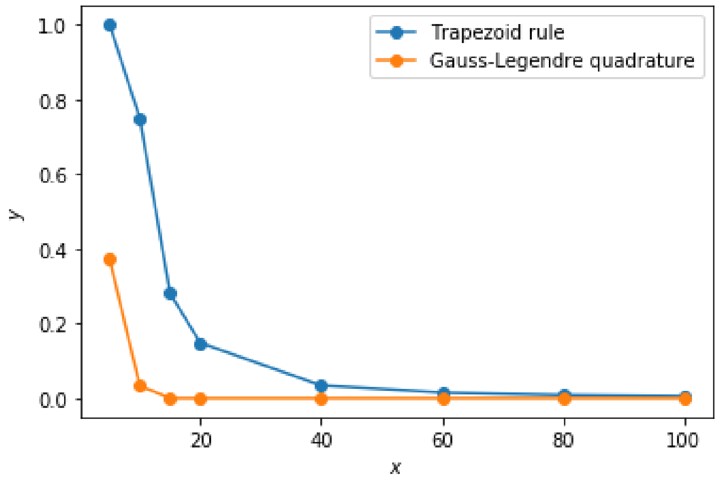


Figure 1: The convergence of the error as the number of nodes (N) increased

5. Both the Gauss-Legendre quadrature & trapezoid The Gauss-Legendre quadrature converges quickly – after N=20 nodes. The trapezoid approximation is much slower, however, it doesn't converge until much later – ~100 nodes in.


Trapezoid rule
Gauss-Legendre quadrature
y
x