Problem 1

Apologies in advance, I didn't figure out a convenient way to write the power laws on the plots themselves. So instead:

1. Runge Error power law is about n^3
2. $|x|$ error power law is about n^5
3. Step error power law is about n^4

Problem 2

1. $f(x) = x^8$

   Since $f(x)$ is a simple monomial of degree 8, our use of 5 quadrature points guarantees that we will get an exact solution for any number n of subdomains, since Gauss Quadrature is exact for polynomials up to order $2N + 1$, which is 11 even for $n = 1$. Therefore, there is no power law dependency.

2. $f(x) = |x - 1/\sqrt{2}|^3$

   The power law for this is about h^4. Thinking about possible polynomial representations of this function, this would need comparatively few basis functions to provide a decent approximation, but significantly more than x^8.

3. $f(x) = H(x - 1/\sqrt{2})$

   The power law for this is about h^2 (splitting hairs here). This is because this function is highly nonlinear,

4. $f(x) = 1/\sqrt{x}$ The power law for this is about $\sqrt{h}$. The error convergence is slow largely due to the singularity, which is difficult to capture with polynomial basis functions.

Problem 3

1. Eigenvalues and eigenvectors of $\mathbf{T}_\alpha$

   I ran out of time to figure this out properly other than investigating test cases on MATLAB. Sorry!

2. Positive Definite

   Since $\mathbf{T}_\alpha$ is symmetric, it will be positive definite if all of its eigenvalues are real and positive. From part (1) above, it's clear that the eigenvalues will all be positive if *alpha* satisfies the following inequality:

   $$\alpha > 2\cos\left(\frac{j\pi}{n+1}\right)$$

   In any finite case, $\alpha = 2$ will always result in a positive definite matrix.

3. Convergence Factors and Iteration Ratio

   (a) The convergence factor of any iteration scheme is defined as $\rho(\mathbf{G}) = \max(|\lambda_i|)$. In this case, write

   $$\mathbf{G} \equiv \begin{bmatrix} \frac{1}{2} & & & \\ & \frac{1}{2} & & \\ & & \ddots & \\ & & & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & -1 & & & \\ -1 & 0 & -1 & & \\ & & \ddots & & \\ & & & -1 & 0 \end{bmatrix} = \frac{1}{2}\mathbf{T_0}$$

   Giving that the convergence factor of the Jacobi scheme explicitly is

   $$\rho(G)_J = |\cos\left(\frac{\pi}{n+1}\right)|$$

   Since the convergence matrix's largest magnitude eigenvalue will be the one associated with $j = 1$.

   (b) Observe from our convergence requirement:

   $$\begin{aligned} \|\mathbf{e}^n\|_2 &\leq \rho^n \|\mathbf{e}^0\|_2 \\ \implies n &\approx \frac{\log\epsilon}{\log\rho} \\ \implies \rho^n &\approx \epsilon \end{aligned}$$

We expect that $2n_{GS} = n_J$ from class, thus we have that

$$\rho(G_{GS}) \approx \rho(G_J)^2 = |\cos\left(\frac{\pi}{n+1}\right)|^2$$

(c) Based on discussions in class, we expect the ratio of number of Jacobi iterations to Gauss-Seidel iterations to be half for a given level of convergence.

4. Implementation

(a) I found

(b) The convergence rates found here align with what I found earlier–Gauss-Seidel is twice as fast as Jacobi, and Jacobi is incredibly slow since I found earlier that $\rho(G_J) \approx 0.9995$.

(c) I expect the convergence rate for an arbitrary vector will be the same.