In this homework we find the solution to
\[ \frac{d^2 u(x)}{dx^2} = \sin(\pi x)e^{-100x^2}, \quad -1 \leq x \leq 1, \] (1)
with some different boundary conditions using some different methods.

1. Legendre Galerkin

We first consider the homogenous boundary conditions \( u(-1) = u(1) = 0 \). In the Galerkin method we seek a solution
\[ u_N(x) = \sum_{i=0}^{N} c_i \phi_i(x), \]
where \( \phi_i(x) \) is in the space \( B_N \) composed of polynomials of degree at most \( N \) that satisfies the boundary conditions. To find the coefficients \( c_i \) we ask that the residual
\[ R_N(x) = \frac{d^2 u_N(x)}{dx^2} - \sin(\pi x)e^{-100x^2}, \]
is orthogonal to \( B_N \) (this is the Galerkin procedure) in some inner product. In the Legendre Galerkin method we choose the basis functions to be
\[ \phi_i(x) = P_{i+1} - P_{i-1}, \quad i \geq 1, \]
where \( P_i(x) \) is the Legendre polynomials (eq. (4.5.55) in DQB, note that these can be evaluated in Matlab using \texttt{legendre(n,X)}). Check that \( \phi_i(x) \) satisfies the boundary conditions. With this choice the expansion becomes (\( \phi_{N-1} \) is of degree \( N \))
\[ u_N(x) = \sum_{i=1}^{N-1} c_i \phi_i(x). \]
You should find the coefficients \( c_i \) by solving the system of equations
\[ \int_{-1}^{1} R_N(x)\phi_k(x)dx = 0, \quad k = 1, \ldots, N - 1. \]
Precisely, first find expressions for the elements of the matrix
\[ S_{i,k} = \int_{-1}^{1} \frac{d^2 \phi_i(x)}{dx^2} \phi_k(x)dx, \]
by using the properties of the Legendre polynomials. Hint: Integrate by parts and use that the recursion for the derivative of \( P_l \)
\[ P_l'(x) = P_{l-2}'(x) + (2l - 1) P_{l-1}(x), \quad P_0'(x) = 0, \quad P_1'(x) = 1, \]
can be reformulated into
\[ P'_{n+1}(z) = (2n + 1)P_n + (2(n - 2) + 1)P_{n-2} + (2(n - 4) + 1)P_{n-4} + \ldots \]

Then, use some sufficiently accurate numerical quadrature to compute the integrals
\[ \bar{F}_k = \int_{-1}^{1} \sin(\pi x)e^{-100x^2}\phi_k(x)dx, \quad k = 1, \ldots, N - 1. \]

Solve the system \( S\bar{c} = \bar{F} \) numerically and demonstrate self-convergence, that is, use the computed solution with the largest \( N \) as exact solution and plot the errors for the other (smaller) \( N \) as a function of \( N \). What is the rate of convergence?

2. **Chebyshev Tau method**

Now consider Laplace’s equation \( u_{xx} = 0, -1 \leq x \leq 1 \), with the more complicated boundary conditions
\[
\begin{align*}
    a_1 u(-1) + b_1 \frac{du(-1)}{dx} &= c_1, \\
    a_2 u(1) + b_2 \frac{du(1)}{dx} &= c_2.
\end{align*}
\]

In the Tau method we still seek a solution in \( B_N \) but we ask that the residual is orthogonal to \( P_{N-k} \) where \( k \) is the number of boundary conditions. The \( k \) additional equations are obtained by plugging in the approximation into the boundary conditions.

This time find solutions in terms of the Chebyshev polynomials
\[ u_N(x) = \sum_{i=0}^{N} c_i T_n(x), \]

by solving the \((N + 1) \times (N + 1)\) system composed of the equations
\[
\begin{align*}
    a_1 u_N(-1) + b_1 \frac{du_N(-1)}{dx} &= c_1, \\
    a_2 u_N(1) + b_2 \frac{du_N(1)}{dx} &= c_2,
\end{align*}
\]

and the \( N-1 \) equations asking for \( R_N \) to be \( w(x) = 1/\sqrt{1-x^2} \) orthogonal to \( T_k, k = 0, \ldots, N - 2, \)
\[ \int_{-1}^{1} \frac{d^2u_N(x)}{dx^2}T_k(x) \frac{1}{\sqrt{1-x^2}}dx = 0, \quad k = 0, N - 2. \]

Here try (don’t get stuck on this) to compute the integral by hand using the properties of the Chebyshev polynomials or use numerical quadrature based on Chebyshev polynomials as illustrated in the following code:
\[ x_{\text{int}} = -\cos((0:N_{\text{int}})*\pi/N_{\text{int}})'; \]
\[ w_{\text{int}} = \text{ones}(1,N_{\text{int}}+1)*\pi/N_{\text{int}}; \]
\[ w_{\text{int}}(1) = w_{\text{int}}(1)/2; \]
\[ w_{\text{int}}(N_{\text{int}}+1) = w_{\text{int}}(N_{\text{int}}+1)/2; \]
\[ f = \exp(x_{\text{int}}); \]
\[ I_C = w_{\text{int}}*f; \]

The above code finds an approximate value for

\[ \int_{-1}^{1} e^{x} \frac{1}{\sqrt{1-x^2}} dx. \]

Choose a non trivial boundary condition and again perform self convergence. Compare the results you got with the above quadrature with the one you get after using the composite midpoint rule

\[ h = 2/N_{\text{int}}; \]
\[ x_{\text{int}} = -1+ h*(0.5+0:(N_{\text{int}}-1))'; \]
\[ f = \exp(x_{\text{int}}); \]
\[ w = 1./\text{sqrt}(1-x_{\text{int}}.^2); \]
\[ I_M = h*\text{sum}(f.*w); \]