NOTE: For each homework assignment observe the following guidelines:

- Include a cover page.
- Always clearly label all plots (title, x-label, y-label, and legend).
- Use the `subplot` command from MATLAB when comparing 2 or more plots to make comparisons easier and to save paper.

Part 1: Non-Uniform Grid

Consider the non-uniform grid:

1. Derive a finite difference approximation to $u''(x_2)$ that is accurate as possible for smooth functions $u(x)$, based on the four values $U_1 = u(x_1), \ldots, U_4 = u(x_4)$. Give an expression for the dominant term in the error.

In the next two questions, you will try to determine an order of accuracy for your method:

2. To get a better feel for how the error behaves as the grid gets finer, take 500 values of $H$, where $H$ spans 2 or 3 orders of magnitude, and for each value of $H$, randomly generate three numbers, $h_1, h_2,$ and $h_3$, where each $h_i \in [0, H]$. For each value $H$, compute your approximation to $u''(x_2)$ using the randomly generated $h_i \in [0, H]$. Plot the error against $H$ on a log-log plot to get a scatter plot of the behavior as $H \to 0$. NOTE: in MATLAB the command `x = H*rand(1)` will produce a single random number in the range $[0,H]$.

3. Estimate the order of accuracy by doing a least squares fit of the form

$$\log(E(H)) = K + p \log(H)$$

to determine $K$ and $p$ based on the 500 data points. Recall that this can be done by solving the following linear system in the least squares sense:

$$\begin{bmatrix}
1 & \log(H_1) \\
1 & \log(H_2) \\
\vdots & \vdots \\
1 & \log(H_{500})
\end{bmatrix} \begin{bmatrix} K \\ p \end{bmatrix} = \begin{bmatrix} \log(E(H_1)) \\
\log(E(H_2)) \\
\vdots \\
\log(E(H_{500}))\end{bmatrix}.$$

NOTE: “In the least-squares sense” means that one should solve the rectangular system $Ax = b$, by solving the (square) normal equation: $A^T A x = A^T b$. 

1
Part 2: Mixed Conditions
Consider the following 2-point BVP:

\[ u'' + u = f(x), \quad \text{on} \quad 0 \leq x \leq 10 \]
\[ u'(0) - u(0) = 0, \quad u'(10) + u(10) = 0. \]

4. Construct a second-order accurate finite-difference method for this BVP. Write your method as a linear system of the form \( A\vec{u} = \vec{f} \).

5. Construct the exact solution to this BVP with \( f(x) = -e^x \).

6. Verify that your method is second order accurate by solving the BVP with \( f(x) = -e^x \) at four different grid spacings \( h \).

HINT 1: Use the \texttt{spdiags} command in \texttt{MATLAB} to create the sparse matrix \( A \) – this will save storage and allow \texttt{MATLAB} to use a fast solver. Modify the following commands, which generate a tri-diagonal matrix with a \([1,-2,1]\) structure, to model your specific BVP:

\[
\begin{align*}
e &= \text{ones}(n,1); \\
A &= \text{spdiags}([e -2*e e], -1:1, n, n);
\end{align*}
\]

HINT 2: Use the command \( \vec{u} = A\backslash \vec{f} \) to invert your system – this will use a fast solver if you created your matrix with \texttt{spdiags}.

Part 3: Periodic Boundary Conditions
Consider the following 2-point BVP:

\[ -u'' + u = f(x), \quad \text{on} \quad 0 \leq x \leq 1 \]
\[ u(0) = u(1) \quad u'(0) = u'(1) \quad (\text{these are periodic BCs}). \]

7. Construct a fourth-order accurate finite-difference method for this BVP based on the fourth-order central finite difference.

Write your method as a linear system of the form \( A\vec{u} = \vec{f} \).

8. Construct the exact solution to this BVP with \( f(x) = \sin(4\pi x) \).

9. Verify that your method is fourth-order accurate by solving the BVP with \( f(x) = \sin(4\pi x) \) at four different grid spacings \( h \).

10. Prove that your method is consistent with truncation error \( \|\vec{\tau}\| = O(h^4) \) and \( L_2 \)-stable, thereby proving that your method converges at \( O(h^4) \) in the \( L_2 \)-norm.

HINT: what is the smallest eigenvalue of the discrete version of \(-u''\) ? How does \(+u\) modify this?