Problem 1 Consider the Poisson equation
\[ \Delta u = f(x, y) \quad \text{for } x \in [0, 1] \text{ and } y \in [0, 1] \]
\[ u(x, y) = 0 \quad \text{if } x = 0 \text{ or } x = 1 \text{ or } y = 0 \text{ or } y = 1. \]

With
\[ f(x, y) = \sin(\pi x) \sin(2\pi y), \]
the true solution is
\[ u(x, y) = -\frac{f(x, y)}{(5\pi^2)}. \]

Use the finite difference method with \( x_i = ih, \ i = 0, \ldots, n + 1 \) and \( y_j = jh, \ j = 0, \ldots, n + 1 \), for the values \( n = 10, 50, 100 \) and \( h = 1/(n+1) \). Compute the maximum absolute error in your approximation and produce a log-log plot with \( h \) in the abscissa and the error in the ordinate. Is this plot consistent with the expected \( O(h^2) \) convergence rate?

Notes:
- You may find it easier to write the discretization matrix with Matlab’s \texttt{kron} (in Octave replace by \texttt{spkron}).
- You may use Matlab’s backslash to solve the system.
- Your system matrix should be \( n \times n \). Think of using matrix operations to put values in lexicographic ordering:
  \[ x = \texttt{linspace}(0,1,n+2); \quad y = \texttt{linspace}(0,1,n+2); \]
  \[ [X,Y] = \texttt{ndgrid}(x(2:n+1),y(2:n+1)); \]
  \[ \texttt{utrue} = \emptyset(x,y) \ldots \% \text{some function} \]
  \[ \texttt{Utrue} = \texttt{utrue}(X,Y); \]
- The matrix \( \texttt{Utrue} \) is \( n \times n \) and such that
  \[ \texttt{Utrue}(i,j) = \texttt{utrue}(x(i+1),y(j+1)) \]
- Since the vector \( \texttt{Utrue}(:) \) contains the columns of \( \texttt{Utrue} \) concatenated, it corresponds to ordering the nodes by \( y \) and then by \( x \) as in the following example with \( n = 3 \) (which includes only the nodes that are not on the boundary)

\[
\begin{array}{ccc}
7 & 8 & 9 \\
y \uparrow & 4 & 5 & 6 \\
1 & 2 & 3 \\
\rightarrow \\
x
\end{array}
\]

where the arrows indicate the direction of increasing values of the corresponding variables.
Problem 2 Consider the parabolic PDE (heat equation)
\[ u_t = u_{xx} \quad \text{for } t > 0 \text{ and } x \in [0,1], \]
\[ u(x,0) = \eta(x) \quad \text{for } x \in [0,1], \]
\[ u(0,t) = u(1,t) = 0 \quad \text{for } t > 0, \]

Use the Crank-Nicholson method with the space discretization \( x_i = ih, \) \( i = 0,\ldots,n+1, \) \( h = 1/(n+1), \) \( n = 100 \) and time discretization \( k = 1/1000 \) to approximate the solution for the initial conditions
(a) \( \eta(x) = \sin(\pi x) \)
(b) \( \eta(x) = \sin(\pi x) + \sin(10\pi x) \)

Please include snapshots of both solutions at times \( t = 2k \) and \( t = 5k. \)

Notes:
- With these particular boundary conditions the method can be written as
  \[ U^{n+1} = (I - (k/2)A)^{-1}(I + (k/2)A)U^n \]
  where \( A \) is the usual finite difference discretization of the 1D Laplacian.
- You may use Matlab’s backslash to solve the systems at each iteration.