Instructions. Identify the group of 2 classmates with whom you will complete this assignment. Then, for each of the homework problems below,

1. Please have the group write up one complete solution in LaTeX.
2. Document the contributions of each member of the group in solving this problem.
3. Include sufficient written discussion to demonstrate your understanding of the problem.

Please email the PDF and mfiles of your homework solutions by **April 18** before 11:45 pm.

Homework Problems.

1. Consider the Lax-Wendroff method

$$U_j^{n+1} = U_j^n - \frac{1}{2}\nu(U_{j+1}^n - U_{j-1}^n) + \frac{1}{2}\nu^2(U_{j-1}^n - 2U_j^n + U_{j+1}^n).$$

   In Matlab, we might be tempted to implement this method with periodic boundary conditions on the grid $x_i = ih$ for $i = 1, 2, \cdots, m+1$ with $h = (b-a)/(m+1)$ within a time-stepping loop as

```
U(1) = U(1) - 0.5*nu*(U(2) - U(m+1))...
    +0.5*nu^2*(U(m+1) - 2.0*U(1) + U(2));

U(2:m) = U(2:m) - 0.5*nu*(U(3:m+1) - U(1:m-1))...
    +0.5*nu^2*(U(1:m-1) - 2.0*U(2:m) + U(3:m+1));

U(m+1) = U(m+1) - 0.5*nu*(U(1) - U(m))...
    +0.5*nu^2*(U(m) - 2.0*U(m+1) + U(1));
```

   Note that periodic boundary conditions mean that $U_0 = U_{m+1}$, so we do not solve for U_0 in the equations above. However, this implementation is incorrect. Can you find the bug?

2. Consider the numerical scheme:

$$r_i^{n+1} = \frac{1}{2}\left(r_{i+1}^n + r_{i-1}^n\right) + \frac{k}{2h}\left(s_{i+1}^n - s_{i-1}^n\right)$$
$$s_i^{n+1} = \frac{1}{2}\left(s_{i+1}^n + s_{i-1}^n\right) + \frac{k}{2h}\left(r_{i+1}^n - r_{i-1}^n\right)$$

   (a) With which system of PDEs is this scheme consistent?

(b) Find the modified equations for this scheme.

(c) Using von Neumann stability analysis find conditions on k and h sufficient for this scheme to be stable.

3. The m-file `advection_LW_pbc.m` implements the Lax-Wendroff method for the advection equation on $0 \leq x \leq 1$ with periodic boundary conditions.

   (a) Observe how this method behaves with $m + 1 = 50, 100, 200$ grid points. Change the final time to `tfinal = 0.1` and use the m-files `error_table.m` and `error_loglog.m` to verify second order accuracy.

   (b) Modify the m-file to create a version `advection_up_pbc.m` implementing the upwind method and verify that this is first order accurate.

   (c) Keep m fixed and observe what happens with `advection_up_pbc.m` if the time step k is reduced, e.g. try $k = 0.4h$, $k = 0.2h$ and $k = 0.1h$. When a convergent method is applied to an ODE we expect better accuracy as the time step is reduced and we can view the upwind method as an ODE solver applied to a Method of Lines system. However, you should observe decreased accuracy as $k \to 0$ with h fixed. Explain this apparent paradox. (Hint: What ODE system are we solving with more accuracy? You might also consider the modified equation given by (10.44) in LeVeque).