1. Let $x_i = i\Delta x$. Use Taylor expansion to show that
$$u_x(x_i) \approx \frac{3u(x_i) - 4u(x_{i-1}) + u(x_{i-2})}{2\Delta x} + O(\Delta x^2)$$

2. Let $x_i = i\Delta x$. Use u_i, u_{i-1}, u_{i-2} and u_{i-3} to derive an approximation of $u_{xx}(x_i)$ so that the leading error is $O(\Delta x^2)$.

3. Consider the θ-scheme
$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \theta \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} + (1-\theta)\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}.$$
Use Fourier analysis to prove the following statements.
   (a) If $\frac{1}{2} \leq \theta \leq 1$, the scheme is unconditionally stable.
   (b) If $0 \leq \theta < \frac{1}{2}$, the scheme is stable when
$$r = \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2(1-2\theta)}. \tag{1}$$
   (c) Discuss the influence of θ on the overall efficiency and accuracy.

4. Prove that the scheme
$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{2}$$
is not stable for any $\Delta t > 0$.

5. Coding problem: accuracy test with a Manufactured solution. Feel free to use Matlab/Python/Julia or other programming language (except for Java). If you submit homework with C, $C++$ or Fortran, please include a make file and/or very clear instructions on how to compile and run your code.

   Consider the problem:
$$u_t(x,t) = u_{xx}(x,t) + f(x,t), x \in (0,\pi), \tag{3a}$$
$$u(x,0) = \sin(x), u(0,t) = u(\pi,t) = 0. \tag{3b}$$

(a) Find $f(x,t)$ so that the exact solution is $u(x,t) = \sin(x)\cos(t)$.
(Hint: substituting $u(x,t)$ into $f(x,t) = u_t(x,t) - u_{xx}(x,t)$.)

(b) Define the mesh $0 = x_0 < x_1 < \cdots < x_N = \pi$ with $x_j = j\Delta x$ and $\Delta x = \frac{\pi}{N}$. Implement a code to solve (3) with the explicit scheme

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} + f_j^n \quad f_j^n = f(j\Delta x, n\Delta t). \tag{4}$$

Set the final time as $T = 1$. Choose the time step size Δt as $M = \mathrm{ceil}(\frac{T}{0.5\Delta x^2})$ and $\Delta t = T/M$. Here, $\mathrm{ceil}(A)$ means the smallest integer value greater than or equal to A.
Make an error plot with y-axis to be $\max_j |u_j^M - u(x_j, T)|$ and x-axis to be Δx under the log-log scale with $N = 20, 40, 80, 160$ to show the error scales as $O(\Delta x^2)$.

(c) Implement a code to solve equation (3) with the implicit scheme

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} + f_j^n \quad f_j^n = f(j\Delta x, n\Delta t). \tag{5}$$

Set the final time as $T = 1$. Choose the time step size Δt as $M = \mathrm{ceil}(\frac{T}{\Delta x})$ and $\Delta t = T/M$.
Make an error plot with y-axis to be $\max_j |u_j^M - u(x_j, T)|$ and x-axis to be Δx with $N = 20, 40, 80, 160$ to show the error scales as $O(\Delta x)$.