```
% ME 303 - Zhao Pan
% Programmers: Shaan B, Zubair H, Mirza M, Dharmik R, Milind K
% Date: 2nd April, 2025

% Comparison of Analytical and Numerical (FTCS) Solutions
N = 50; % Fourier terms (for analytical)
Nx = 100; % Number of spatial points
x = linspace(0, 1, Nx+1); % Create spacial vector with 100 points
delta_x = x(2) - x(1); % Spatial grid spacing (Δx)
dt = 0.5 * delta_x^2 / 2; % Time grid spacing (meets stability condition)
r = 2 * dt / delta_x^2; % FTCS scheme coefficient (used in update formula)

% Array of times at which solutions will be evaluated
t_vals = [0.001, 0.01, 0.1, 10];

% Initialize solution matrices for each time value
u_analytical=zeros(length(t_vals),Nx+1); % Stores real solution at each t
u_numerical=zeros(length(t_vals),Nx+1); % Stores FTCS solution at each t

% Compute Analytical Solution
for k = 1:length(t_vals) % Lööp over each specified time value

    t = t_vals(k); % Current time

    for n = 1:N % Sum N Fourier series terms

        % Fourier coefficient for each mode n
        Cn = (4 * (-1)^(n+1)) / (n * pi);

        % Add contribution of mode n
        u_analytical(k,:) = u_analytical(k,:) + ...
            Cn * sin(n * pi * x) * exp(-2 * (n * pi)^2 * t);

    end

    u_analytical(k,:) = u_analytical(k,:) + 2 * x; % Add steady-state part

end

% Initial condition
u = cos(pi * x);  % u(x,0)

% Compute Numerical FTCS Solution
for k = 1:length(t_vals) % Loop over each desired time

    u_temp = u; % Start with the initial condition
    Nt = round(t_vals(k) / dt); % Time steps to reach current time

    % Time-stepping loop
    for step = 1:Nt
```

```matlab
        u_new = u_temp; % Initialize new u for current time step

        for i = 2:Nx

            % Update all interior points using FTCS formula
            u_new(i) = u_temp(i) + r * (u_temp(i+1) - ...
                2*u_temp(i) + u_temp(i-1));

        end

        u_new(1) = 0; % Left boundary condition: u(0,t) = 0
        u_new(end) = 2; % Right boundary condition: u(1,t) = 2
        u_temp = u_new; % Update solution for next time step

    end

    u_numerical(k,:) = u_temp; % Store final solution at time t_vals(k)

end

% Plot comparison (overlay)
figure; % open new figure

for k = 1:4 % Plot for each time in t_vals

    subplot(2,2,k); % 2x2 grid of subplots

    % Analytical in blue solid line
    plot(x, u_analytical(k,:), 'b-', 'LineWidth', 2);

    hold on; % keep multiple plots

    % Numerical in red dashed line
    plot(x, u_numerical(k,:), 'r--', 'LineWidth', 2);

    title(['Comparison at t = ', num2str(t_vals(k))]); % Add title
    xlabel('x'); % X-axis label
    ylabel('u(x,t)'); % Y-axis label
    legend('Analytical', 'Numerical'); % Legend to identify lines
    grid on; % Add the grid for clarity

end
```