```
clc;
clear;
close;

x_0 = 0;
x_f = 5;
h_x = 1/100;

t_0 = 0;
t_f = 1;
h_t = h_x;

% Coefficient function that appears in front of u_x
% a     = @(x) (1/4)*ones(size(x));
a     = @(x) (1/3)*ones(size(x));

% Initial condition
u_0   = @(x) sin(2*2*asin(1)*x);

% Initial condition for u_t
u_t_0 = @(x) cos(2*2*asin(1)*x);

% Forcing function
% f     = @(x, t) -ones(size(x));
f     = @(x, t) ones(size(x));

[u_next, x, t] = wave_solver(a, f, u_0, u_t_0, x_0, x_f, ...
                             h_x, t_0, t_f, h_t);

% [X, T] = meshgrid(x, t);
% surf(X, T, u_next');
% zlabel("u(x, t)");
% xlabel("x");
% ylabel("t");

function [u_next, x, t] = wave_solver(a, f, u_0, u_t_0, x_0, x_f, ...
                                      h_x, t_0, t_f, h_t)

    % Discretize spatial grid
    x = (x_0:h_x:x_f)'; x = x(2:end);

    % Discretize time grid
    t = (t_0:h_t:t_f)';

    % Create matrix of numerical solutions at each time step
    u_next = zeros(length(x), length(t));

    % Create A matrix
    x_minus = x - h_x;
    x_plus  = x + h_x; x_plus(end) = x_minus(1);

    a_minus = a(x_minus);
```

```
    a_plus  = a(x_plus);
    a_tilde = a_minus + a_plus;

    A = diag(-a_tilde) + diag(a_plus(1:(end-1)), 1) ...
                       + diag(a_plus(1:(end-1)), -1);
    A(1, end) = a_minus(1);
    A(end, 1) = a_minus(1);

    B = eye(size(A)) + (1/2)*(h_t/h_x)^(2)*A;

    % Solve the wave equation with negative right-hand side
    % B_sign_change = eye(size(A)) - (1/2)*(h_t/h_x)^(2)*A;

    % Apply second-order central finite difference scheme
    u_next(:, 1) = u_0(x);

    for n = 1:(length(t) - 1)
        t_n = t(n);

        % At the first time step, we handle the need for the solution
        % before the initial time
        if n == 1
            u_next(:, n + 1) = B*u_next(:, 1) + h_t*u_t_0(x) + (h_t)^(2)*f(x,
 t_n);

            % Solve the wave equation with negative right-hand side
            % u_next(:, n + 1) = B_sign_change*u_next(:, 1) + h_t*u_t_0(x) +
 (h_t)^(2)*f(x, t_n);
        else
            u_next(:, n + 1) = 2*B*u_next(:, n) - u_next(:, n - 1) +
 (h_t)^(2)*f(x, t_n);

            % Solve the wave equation with negative right-hand side
            % u_next(:, n + 1) = 2*B_sign_change*u_next(:, n) - u_next(:, n -
 1) + (h_t)^(2)*f(x, t_n);
        end
    end
end
```

Published with MATLAB® R2022a