APPM 5610 - Homework 9

Eappen Nelluvelil

April 13, 2022

1. Implement the Crank-Nicolson scheme for the heat equation

$$
\begin{aligned}
\phi_t &= \left[a(x)\,\phi_x\right]_x + f(x,t), \quad t>0, \quad x\in(0,1),\\
\phi(x,0) &= \phi_0(x), \quad x\in(0,1)\\
\phi(0,t) &= 0, \quad t>0,\\
\phi(1,t) &= 0, \quad t>0.
\end{aligned}
$$

   Verify the scheme on several examples.

   We consider the following examples, with $h_x = h_t = \frac{1}{50}$ and $t\in[0,1]$:

   (a) $a(x) = \frac{1}{10}$, $f(x,t)=0$, $\phi_0(x) = \sin(\pi x)$

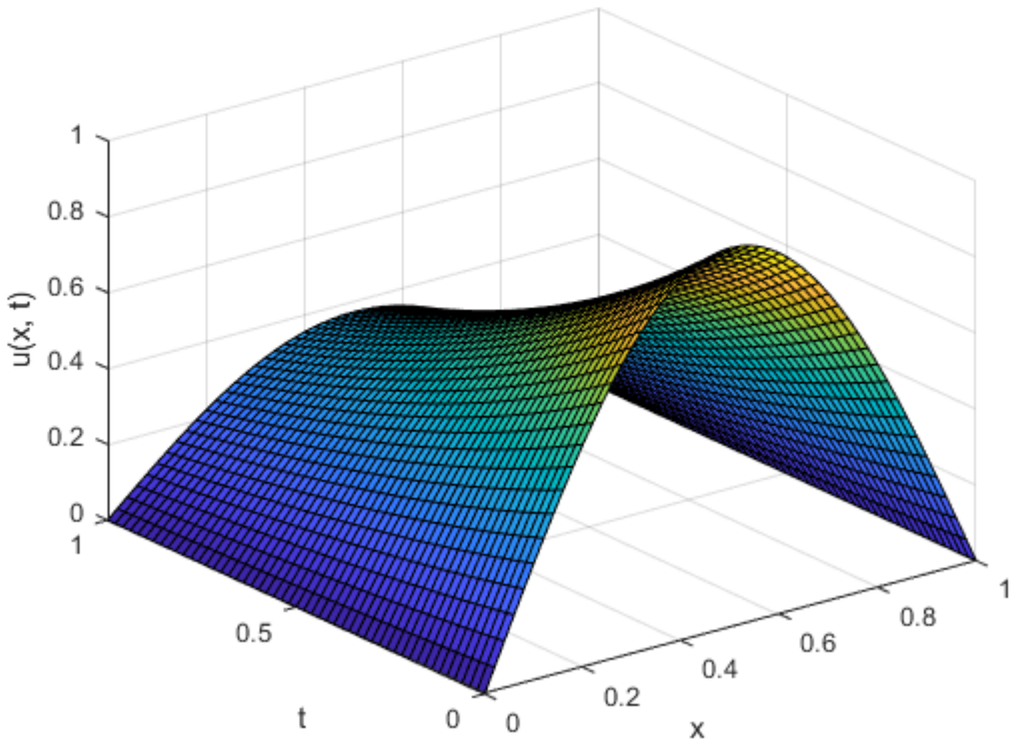

Figure 1: Numerical solution corresponding to $a(x) = \frac{1}{10}$, $f(x,t)=0$, $\phi_0(x) = \sin(\pi x)$ for $x\in[0,1]$ and $t\in[0,1]$

(b) $a(x) = \frac{1}{5}$, $f(x,t) = e^{-t}$, $\phi_0(x) = x(1-x)$

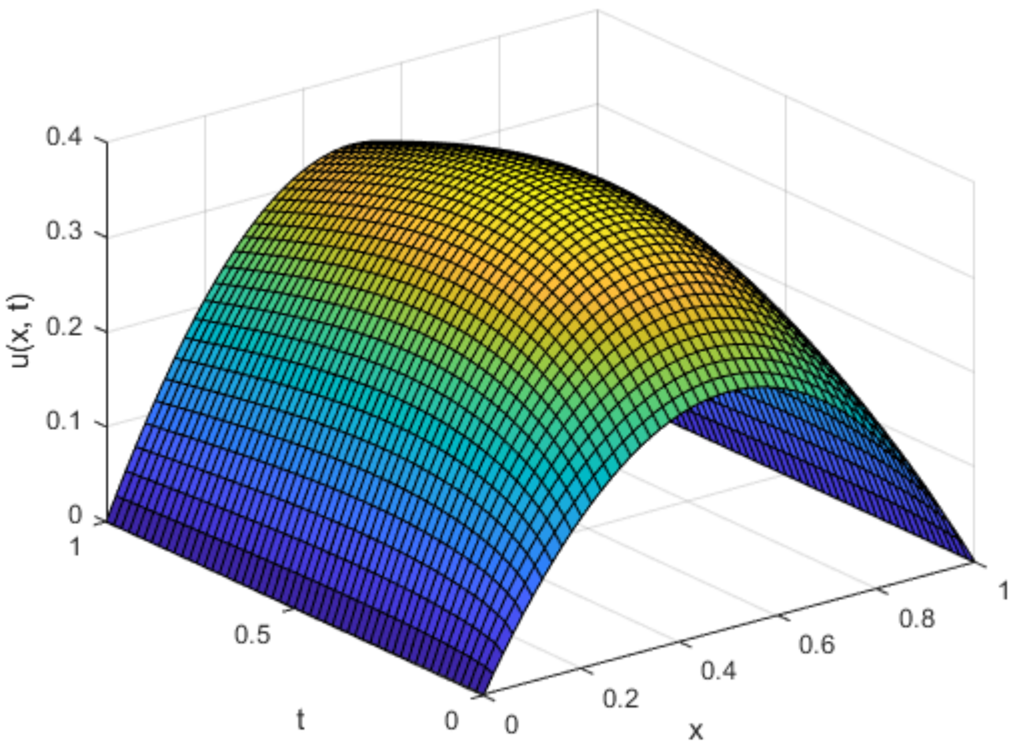

Figure 2: Numerical solution corresponding to $a(x) = \frac{1}{5}$, $f(x,t) = e^{-t}$, $\phi_0(x) = x(1-x)$ for $x \in [0,1]$ and $t \in [0,1]$

2. Implement the explicit second-order central difference scheme for the wave equation

$$\begin{aligned} \phi_{tt} &= [a(x)\phi_x]_x + f(x,t), \quad t > 0, \\ \phi(x,0) &= \phi_0(x), \\ \phi_t(x,0) &= \phi_1(x), \end{aligned}$$

where all functions are periodic in x with the period 1. Verify the scheme on several examples using an appropriate choice (explain) of step sizes h_t and h_x. Also run the scheme with the step sizes violating the stability criterion. Describe the numerical effect.

We consider the following examples, with $h_x = h_t = \frac{1}{50}$ and $t \in [0,1]$

(a) $a(x) = \frac{1}{4}$, $\phi_0(x) = \sin(2\pi x)$, $\phi_1(x,) = \cos(2\pi x)$, and $f(x,t) = -1$

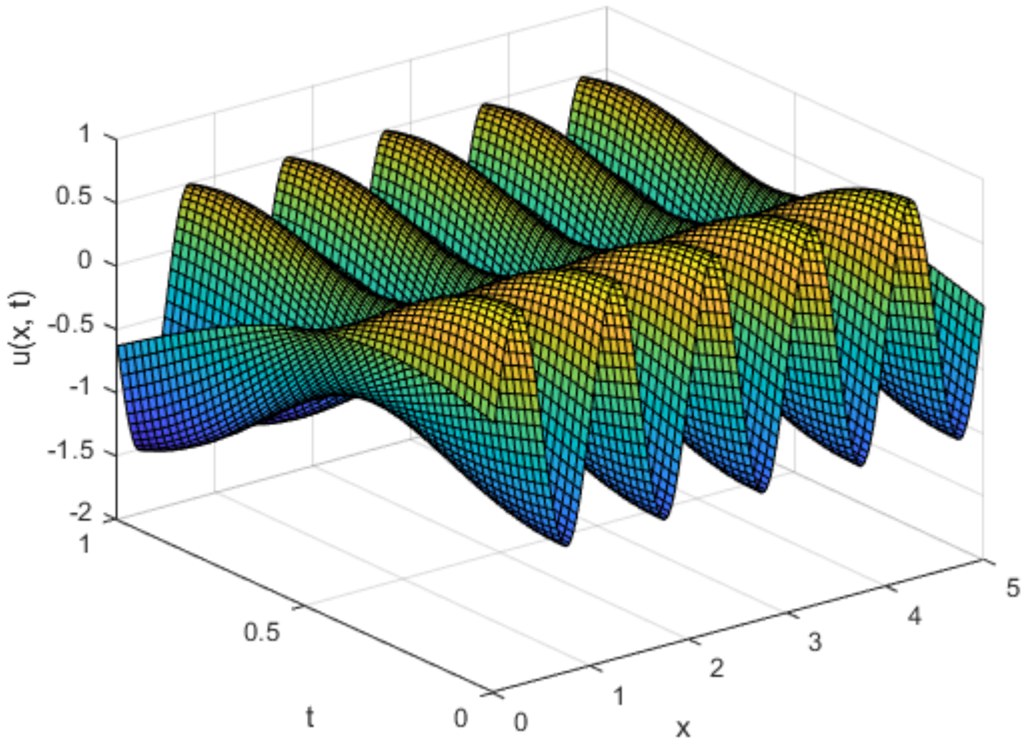

Figure 3: Numerical solution corresponding to $a\left(x\right)=\frac{1}{4}$, $\phi_0\left(x\right)=\sin\left(2\pi x\right)$, $\phi_1\left(x,\right)=\cos\left(2\pi x\right)$, and $f\left(x,t\right)=-1$ for $x\in[0,5]$ and $t\in[0,1]$

(b) $a\left(x\right)=\frac{1}{3}$, $\phi_0\left(x\right)=\sin\left(2\pi x\right)$, $\phi_1\left(x,\right)=\cos\left(2\pi x\right)$, and $f\left(x,t\right)=1$

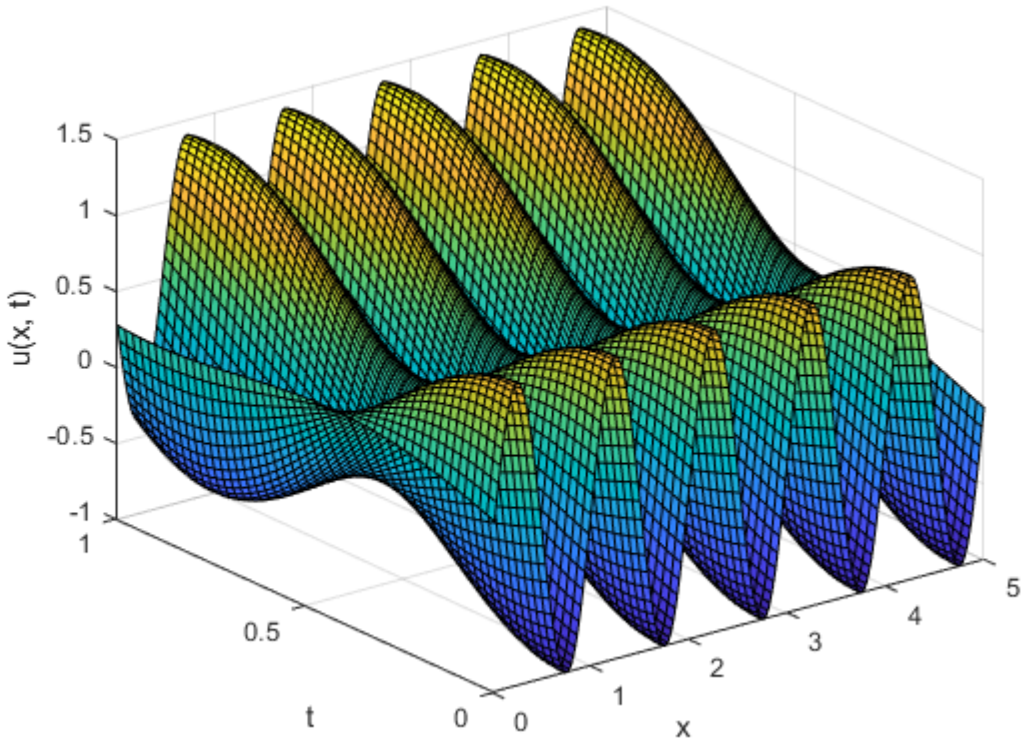

Figure 4: Numerical solution corresponding to $a(x) = \frac{1}{3}$, $\phi_0(x) = \sin(2\pi x)$, $\phi_1(x,) = \cos(2\pi x)$, and $f(x,t) = 1$ for $x \in [0, 5]$ and $t \in [0, 1]$

In both cases, $|a| < 1$ and $h_t = h_x$, which does not violate the CFL condition. If we were to pick h_t to be larger than h_x, then we see numerical instability in the solutions. For example, the surface below corresponds to the wave equation with the conditions given in (b), except that $h_x = \frac{1}{100}$ and $h_t = \frac{1}{10}$. From the plot, we see that the numerical solution rapidly blows up as $t \to 1$:

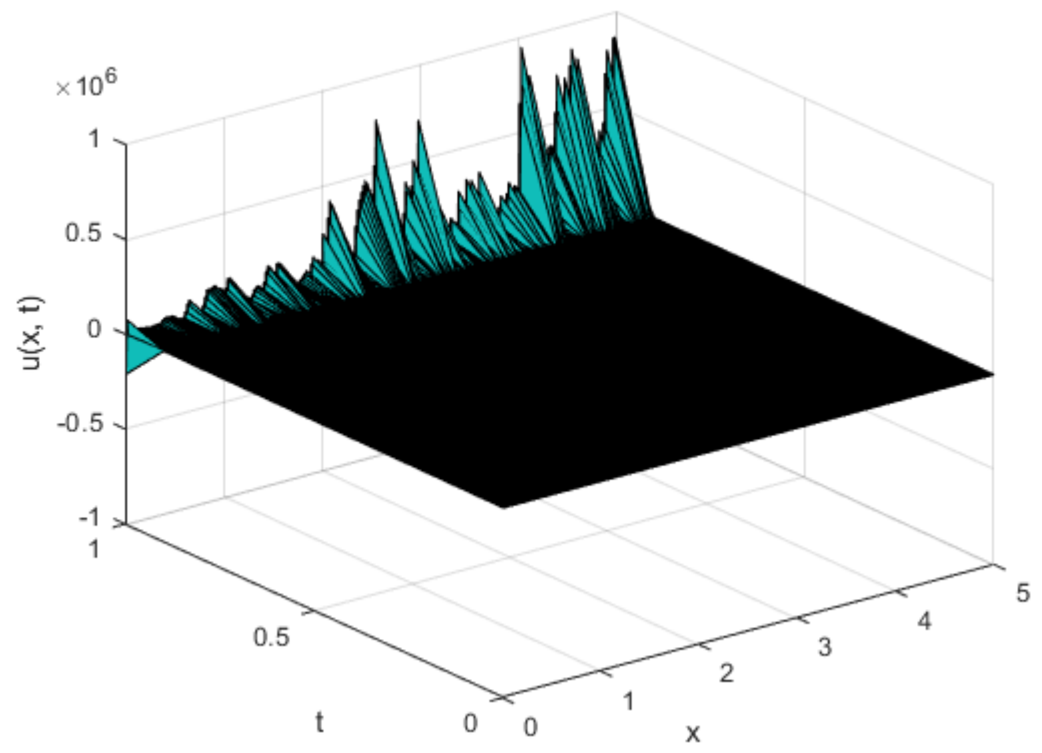

Figure 5: Effects of violating the CFL condition ($h_t = \frac{1}{10}$ and $h_x = \frac{1}{100}$) for the wave equation

3. In both problems, it is suggested to use graphical display of solutions (at various moments in time). Also, as a way of understanding ill-posted problems, reverse the direction of time in the heat equation and the Crank-Nicolson scheme and observe the effects of such time reversal. Also, by changing the sign of the right hand side of the wave equation, try to solve an ill-posed IVP for the resulting elliptic equation and observe the numerical effects.

   If we were to reverse the direction of time in the heat equation and solve the resulting backwards heat equation via the Crank-Nicolson scheme, we see that the numerical solution blows up as $t \to 1$, as expected.

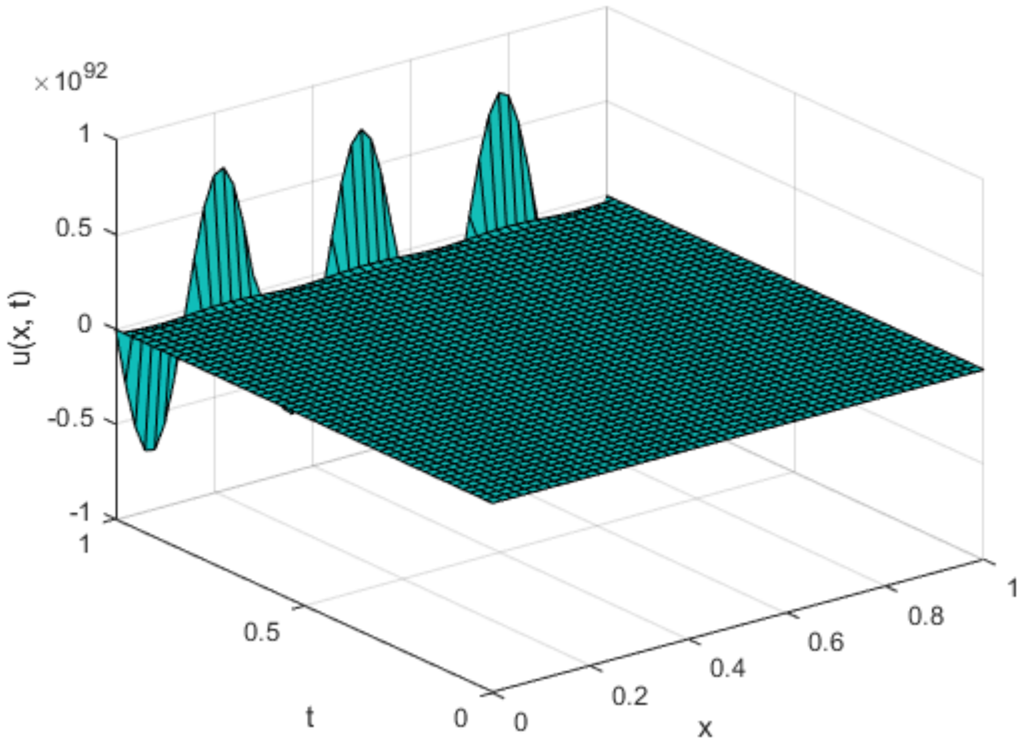

Figure 6: Effects of solving the backwards heat equation via the Crank-Nicolson scheme

If we were to change the sign on the right-hand side of the wave equation and solve the resulting system via second-order central finite differences, we see that the numerical solution blows up as $t \to 1$, which is also expected.

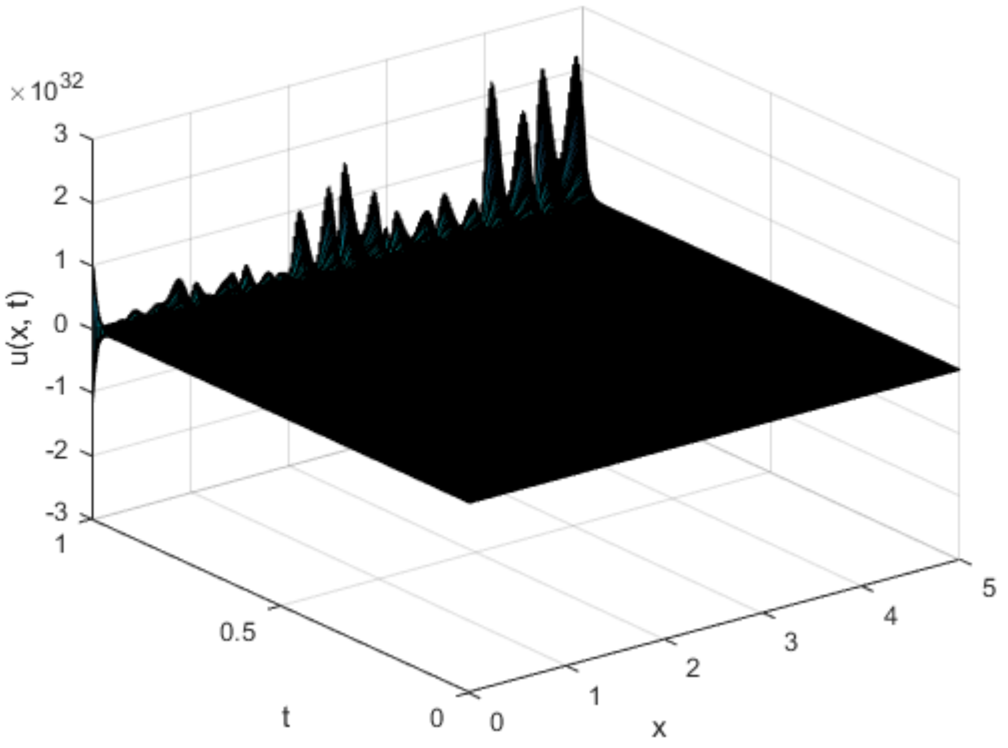

Figure 7: Effects of solving the wave equation with a negative right hand side. Here, $a\left(x\right)=\frac{1}{3}$, $\phi_0\left(x\right)=\sin\left(2\pi x\right)$, $\phi_1\left(x,\right)=\cos\left(2\pi x\right)$, and $f\left(x,t\right)=1$ for $x\in\left[0,5\right]$ and $t\in\left[0,1\right]$.

```
clc;
clear;
close;

x_0 = 0;
x_f = 1;
h_x = 1/50;

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

% Coefficient function that appears in front of u_x
% a   = @(x) (1/10)*ones(size(x));
a   = @(x) (1/5)*ones(size(x));

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

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

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

% Append the homogenous Dirichlet boundary conditions to the solutions
u_next = [zeros(1, length(t)); u_next; zeros(1, length(t))];
x = [x_0; x; x_f];

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

function [u_next, x, t] = cn(a, f, u_0, x_0, x_f, h_x, t_0, t_f, h_t)
    % Discretize spatial grid
    x = (x_0:h_x:x_f)'; % Here, x = [x_0, x_1, ..., x_{N}]
    x = x(2:(end-1)); % Now, x = [x_1, x_2, ..., x_{N-1}]

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

    % Create matrix to store numerical solution at each time step
    u_next = zeros(length(x), length(t));

    % Get solution at time t = t_0
    u_next(:, 1) = u_0(x);

    A_j = (1/2)*(h_t/(h_x^2))*a(x - h_x);
    C_j = (1/2)*(h_t/(h_x^2))*a(x + h_x);
    B_j = 1 + A_j + C_j;

    % Uncomment to see the instability in the backwards heat equation
    % B_j_unstable = 1 - (A_j + C_j);

    mat = diag(A_j(2:end), -1) - diag(B_j, 0) + diag(C_j(1:(end-1)), 1);

    % Matrices for backwards heat equation
    % RHS_mat_unstable = diag(A_j(2:end), -1) + diag(B_j_unstable, 0) +
 diag(C_j(1:(end-1)), 1);

    % Apply CN scheme at each time step
    for n = 1:(length(t) - 1)
        t_n = t(n);
        D_j = mat*u_next(:, n) + 2*u_next(:, n) + (h_t/2)*(f(x, t_n) + f(x,
 t_n + h_t));
        u_next(:, n + 1) = -mat\D_j;

        % Solve for the backward heat equation
        % D_j_unstable = -mat*u_next(:, n) + 2*u_next(:, n) - (h_t/2)*(f(x,
 t_n) + f(x, t_n + h_t));
        % u_next(:, n + 1) = RHS_mat_unstable\D_j_unstable;
    end
end
```

Published with MATLAB® R2022a

```
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