

---

```

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*