
```

function [x, t, w] = waveEquation1D(c, f, g, l, r, xdom, tdom, M, N)
% Solves the 1D wave equation in a string using the Finite Difference Method
% c: wave speed in the string
% f: function (of x) describing the initial state of the string
% g: function (of x) describing the initial rate of change of the vertical
    displacement of the string
% l: function (of time) describing the change in temperature at the left
    boundary of the bar
% r: function (of time) describing the change in temperature at the right
    boundary of the bar
% xdom: 1x2 array giving the locations of the boundaries
% tdom: 1x2 array giving the time interval of interest
% M: number of grid points along the string (i.e. along x-coordinate)
% N: number of grid points along time coordinate
% Discretize domain in time and space
x = linspace(xdom(1), xdom(2), M)';
t = linspace(tdom(1), tdom(2), N)';
%Check that initial and boundary conditions are compatible
TOL = 10^(-9);
if abs(f(x(1)) - l(t(1))) > TOL
    f(x(1))
    l(t(1))
    error('Initial condition does not agree with left boundary condition')
elseif abs(f(x(M)) - r(t(1))) > TOL
    f(x(M))
    r(t(1))
    error('Initial condition does not agree with right boundary condition')
end
% Define step sizes in space and time
h = (xdom(2) - xdom(1))/(M-1);
k = (tdom(2) - tdom(1))/(N-1);
sigma = c*k/h;
if sigma > 1.0
    warning ('Scheme is unstable for chosen discretization parameters')
end
% Initialize displacement matrix
w = zeros(N, M);
%Set the initial condition
w(1,:) = f(x);
gvec = g(x);
%Set left and right boundary conditions
w(:,1) = l(t);
w(:,M) = r(t);
%Construct the matrix A
m = M - 2; %number of inner grid points along the bar
s2 = sigma*sigma;
A = diag((2 - 2*s2)*ones(m,1));
for i=1:m-1
    A(i,i+1) = s2;
    A(i+1,i) = s2;
end
s0 = zeros(m,1);

```

```
sj = zeros(m,1);
% compute displacement at inner nodes for the first time step
w0 = w(1,2:m+1)';
s0(1) = s2*w(1,1);
s0(m) = s2*w(1,M);
w(2,2:m+1) = 0.5*A*w0 + k*gvec(2:m+1,1) + 0.5*s0;
% compute displacement at inner nodes for the each remaining time step
for j=2:N-1

    wj = w(j,2:m+1)';
    wj1 = w(j-1,2:m+1)';

    sj(1) = s2*w(j,1);
    sj(m) = s2*w(j,M);

    w(j+1,2:m+1) = A*wj - wj1 + sj;
end
end

Not enough input arguments.

Error in waveEquation1D (line 13)
x = linspace(xdom(1), xdom(2), M)';
```

Published with MATLAB® R2022a