Project Review: *1D Wave Equation Solution MatLab*

Alexandre Ait-Ettajer, Roman Silen, Steven Spreizer

May 20, 2020

All tasks regard the 1-D Wave Equation and its solutions.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial t^2} \rightarrow u_{tt} = c^2 u_{xx}$$

Task 1

Verify that $u = cos(k(x + ct))$ is a solution to the 1-D Wave analytically and use MatLab to visualize such a solution and its evolution.

$u_{tt} = c^2 u_{xx}$

$u = \cos(k(x - ct))$

$u_t = kc \sin(k(x-ct))$

$u_{tt} = -k^2c^2 \cos(k(x-ct))$

$u_x = -k \sin(k(x-ct))$

$u_{xx} = -k^2 \cos(k(x-ct))$

$c^2 u_{xx} = -k^2c^2 \cos(k(x-ct)) = u_{tt}$ ✓

Task 2

Verify that $u = cos(k(x - ct))$ is a solution to the 1-D Wave analytically and use MatLab to visualize such a solution and its evolution.

$u = \cos(k(x+ct))$

$u_t = -kc \sin(k(x+ct))$

$u_{tt} = -k^2c^2 \cos(k(x+ct))$

$u_x = -k \sin(k(x+ct))$

$u_{xx} = -k^2 \cos(k(x+ct))$

$c^2 u_{xx} = -k^2c^2 \cos(k(x+ct)) = u_{tt}$ ✓

Task 3

Verify that $u = cos(k(x - ct)) + cos(k(x + ct))$ is a solution to the 1-D Wave analytically and use MatLab to visualize such a solution and its evolution.

$u = \cos(k(x-ct)) + \cos(k(x+ct))$

$u_t = -kc \sin(k(x-ct)) - kc \sin(k(x+ct))$

$u_{tt} = -k^2c^2 \cos(k(x-ct)) - k^2c^2 \cos(k(x+ct))$

$u_x = -k \sin(k(x-ct)) - k \sin(k(x+ct))$

$u_{xx} = -k^2 \cos(k(x-ct)) - k^2 \cos(k(x+ct))$

$c^2 u_{xx} = -k^2c^2 \cos(k(x-ct)) - k^2c^2 \cos(k(x+ct)) = u_{tt}$

Code to output a movie of the above three solutions of the 1-D wave equation. Movies stored in Git Repo.

```
% User Input of Parameters for wave equation solutions
N = input('Resolution of x-space (0,1) (N): ');
M = input('Total Time Steps (M): ');
k = input('Wave Parameter (k): ');
c = input('Wave Speed (c): ');
t_f = input('Final Time (t_f): ');
a = 0;
b = 1;

% Define x and t interval
x = linspace(a,b,(N));
dx = (b-a)/(N-1);
dt = t_f/M;
assert(x(2)-x(1) == dx);

% Loop over
for n = 1:M
   t = n*dt;
   u1 = f1(x,t,c,k);
   u2 = f2(x,t,c,k);
   u3 = f3(x,t,c,k);

   % Figure of Solution 1 (Forward Propagating Wave)
   figure(1)
   plot(x,u1);
   xlim([a b])
   ylim([-1 1])
   title('Wave Equation Solution: Backward Propagation')
   xlabel('x')
   ylabel('u')

   % Figure of Solution 2 (Backward Propagating Wave)
   figure(1)
   plot(x,u2);
   xlim([a b])
   ylim([-1 1])
   title('Wave Equation Solution: Forward Propagation')
   xlabel('x')
   ylabel('u')

   % Figure of Solution 3 (Standing Wave)
   figure(1)
   plot(x,u3);
   xlim([a b])
   ylim([-1 1])
   title('Wave Equation Solution: Standing Wave')
   xlabel('x')
   ylabel('u')

   pause(t_f/M);
end

% Solution 1 to the 1-D wave eqaution
function u = f1(x,t,c,k)
    u = 0.5*cos(2*pi*k*(x+c*t));
end

% Solution 2 to the 1-D wave eqaution
```

```
function u = f2(x,t,c,k)
    u = 0.5*cos(2*pi*k*(x-c*t));
end

% Solution 3 to the 1-D wave eqaution (Superposition of Sol. 1 and 2)
function u = f3(x,t,c,k)
    u = f1(x,t,c,k) + f2(x,t,c,k);
end
```

Listing 1: Analytical Solution for 1-D Wave Equation Visualized

Task 4

Find the solution analytically for the following Boundary Value Problem and create a MatLab code to solve this problem for any initial displacement and velocity in the domain $x \in (0, 1)$.

$$u(x = 0, t) = 0$$

$$u_x(x = 1, t) = 0$$

To solve the BVP analytically, we use the Separation of Variables method:

Let $u(x,t) = X(x)T(t)$

$u_{xx} = X''(x)T(t)$

$u_{tt} = X(x)T''(t)$

$X(x)T''(t) - c^2 X''(x)T(t) = 0$

$X(x)T''(t) = c^2 X''(x)T(t)$

$\frac{1}{c^2} \cdot \frac{T''(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda$

Now we find the corresponding eigenfunctions to the separation constant λ by solving the ODE $X''(x) + \lambda X(x) = 0$. We split this problem into 3 separate cases for λ in order to find non-trivial solutions.

$X''(x) + \lambda X(x) = 0$

$X(0) = 0$, $X(1) = 0$

$r^2 + \lambda = 0$

Case 1: $\lambda < 0$

$r = \pm\sqrt{|\lambda|}$

$X(x) = C_1 e^{\sqrt{|\lambda|}x} + C_2 e^{-\sqrt{|\lambda|}x}$

$X(0) = 0 = C_1 + C_2 \rightarrow C_1 = -C_2$

$X(1) = 0 = C_1(e^{\sqrt{|\lambda|}x} - e^{-\sqrt{|\lambda|}x}) \rightarrow C_1 = 0 \rightarrow C_2 = 0$

Case 2: $\lambda = 0$

$r = 0$

$X(x) = C_1 + C_2 x$

$X(0) = 0 = C_1$

$X(1) = 0 = C_2$

Case 3: $\lambda > 0$

$r = \pm\sqrt{\lambda}\, i$

$X(x) = C_1 \cos(\sqrt{\lambda}\, x) + C_2 \sin(\sqrt{\lambda}\, x)$

$X(0) = 0 = C_1$

$X(1) = 0 = C_2 \sin(\sqrt{\lambda})$

$0 = \sin(\sqrt{\lambda})$

$n\pi = \sqrt{\lambda} \longrightarrow \lambda = n^2\pi^2,\ n \in +\mathbb{Z} \longrightarrow$ general form: $\lambda = \frac{n^2\pi^2}{L^2},\ n \in +\mathbb{Z}$

$\boxed{X(x) = C_2 \sin(n\pi x)}$

We see here that λ must be greater than 0 for a non-trivial solution. Now we use this value for λ to solve the following ODE in the time dimension:

$$T''(t) + c^2 \lambda T(t) = 0$$
$$r^2 + c^2 \lambda = 0 \longrightarrow r = i\sqrt{\lambda}\, c = in\pi c$$
$$T(t) = D_1 \cos(n\pi c t) + D_2 \sin(n\pi c t)$$

Note that D_1 and D_2 are general coefficients. Now we take our solutions for $X(x)$ and $T(t)$ and combine them into a solution for $u(x,t)$.

$$\begin{aligned} u(x,t) &= X(x)T(t) \\ &= c_n \sin(n\pi x) \left[D_1 \cos(n\pi c t) + D_2 \sin(n\pi c t) \right] \\ &= A_n \cos(n\pi c t)\sin(n\pi x) + B_n \sin(n\pi c t)\sin(n\pi x) \end{aligned}$$

Note that $A_n = c_n D_1$ and $B_n = c_n D_2$. Now we take an "infinite superposition" for all values of n:

$$u(x,t) = \sum_{n=1}^{\infty} \left[A_n \cos(n\pi c t)\sin(n\pi x) + B_n \sin(n\pi c t)\sin(n\pi x) \right]$$

Now we use our initial displacement and initial velocity functions, $f(x)$ and $g(x)$, to determine A_n and B_n:

$$u(x,0) = f(x) = \sum_{n=1}^{\infty} A_n \cos(n\pi c t)$$

$$u_t(x,t) = \sum_{n=1}^{\infty} \left[-n\pi c \cdot A_n \sin(n\pi c t)\sin(n\pi x) + n\pi c \cdot B_n \cos(n\pi c t)\sin(n\pi x) \right]$$

$$u_t(x,0) = g(x) = \sum_{n=1}^{\infty} n\pi c \cdot B_n \sin(n\pi x)$$

Both of these give us Fourier sine series on the interval $[0, L]$, where $L = 1$. Using the Fourier sine series formulas gives us:

$$A_n = \frac{2}{L}\int_0^L f(x)\sin\left(\frac{n\pi x}{L}\right)dx = 2\int_0^1 f(x)\sin(n\pi x), \quad n \in \mathbb{N}$$

$$n\pi c \cdot B_n = \frac{2}{L}\int_0^L g(x)\sin\left(\frac{n\pi x}{L}\right)dx = 2\int_0^1 g(x)\sin(n\pi x), \quad n \in \mathbb{N}$$

So our final solution is:

$$u(x,t) = \sum_{n=1}^{\infty} \left[A_n \cos(n\pi c t)\sin(n\pi x) + B_n \sin(n\pi c t)\sin(n\pi x) \right]$$

$$A_n = 2\int_0^1 f(x)\sin(n\pi x)$$
$$B_n = \frac{2}{n\pi c}\int_0^1 g(x)\sin(n\pi x)$$

MATLAB Implementation

```
function [u,w] = oneDWaveSolver(a,b,N,dt,c,tf,f,g,l)
% 1-D Wave Equation FDM solver
%  B.C. u(0,t) = l(t), u_x(b,t) = 0
%  I.C. u(x,0) = f(x), u_t(x,0) = g(x)

x = linspace(a,b,N+1);
dx = x(2)-x(1);
sigma = c*dt/dx;
if(sigma > 1)
    error('CFL number greater than 1')
end

% Initial Condition
w = zeros(N+1,1);
for i = 1:N+1
   w(i) = f(x(i));
end
dummy = w(N);

% First Time Step
temp = zeros(N+1,1);
temp(1) = l(dt);
for i = 2:N
    temp(i) = (1-sigma^2)*w(i)+dt*g(x(i))+sigma^2/2*(w(i-1)+w(i+1));
end
temp(N+1) = (1-sigma^2)*w(i)+dt*g(x(i))+sigma^2/2*(w(i-1)+dummy);
dummy = temp(N);

w = [w,temp];

% Remaining Time Steps
j = 2;
temp = zeros(N+1,1);
while j*dt <= tf
    temp(1) = l(j*dt);
    for i = 2:N
        temp(i) = 2*w(i,j)-w(i,j-1) + sigma^2 * (w(i+1,j) - 2*w(i,j) + w(i-1,j));
    end
    temp(N+1) = 2*w(i,j)-w(i,j-1) + sigma^2 * (dummy - 2*w(i,j) + w(i-1,j));
    dummy = temp(N);
    w = [w,temp];

    j = j+1;
end

[~,n] = size(w);
u = w(:,n);

end
```

Listing 2: 1-D Wave Equation Solver for above boundary conditions

In the output, u is the final state of the wave in vector form while w is a matrix contatining the wave at all time steps.

```
a = 0;
b = 1;
```

```
N = 100;
c = 1;
dt = .01;
tf = pi;
f = @(x) sin(3*pi/2*x);
g = @(x) -sin(3*pi/2*x);
l = @(t) 0;

[u,w] = oneDWaveSolver(a,b,N,dt,c,tf,f,g,l);
figure(1)
surf(w);
xlabel('time step');
ylabel('x index');
zlabel('wave amplitude');
shading interp
pause
figure(2)
[~,m] = size(w);
for n = 1:m
    plot([1:101],w(:,n));
    xlim([0 101])
    ylim([-1.5 1.5]);
    xlabel('x index');
    ylabel('amplitude');
    if n==1
        pause
    end
    pause(.001);
end
```

Listing 3: Driver Code and Plot Generation