Table of Contents

1D One Way Wave Equation 1
Initial Given values 1
Calculating Numerical and Analytical Solution 1
Initial Condition 2
FTBS Scheme 2
Analytical Solution 2
Accuracy Test 3
Plot 3
Discussion 4

1D One Way Wave Equation

```
% Solving 1D One Way Wave Equation using FTCS Scheme
% Author: Pradeep Singh
% Date: 10/30/2017

clc
close all
```

Initial Given values

```
% Domain values for x and t
x_min = -1;
x_max = 1;
t_min = 0;
t_max = 1.2;
c = 1;      % c = 1 (given value)

k =10;  % Number of times we double our grid size
        % for accuracy test.
m = 2;  % Resolution
n = 2;  % Calculate accuracy at this given point

% Store data in store matrix to calculate the accuracy.
% First row will hold analytical solution and the
% Second row will hold numerical solution (FTBS Scheme).

store = zeros(2, k);
```

Calculating Numerical and Analytical Solution

```
% Loop k times
for j=1:k

    if(j > 1)
        m = 2*m; %doubling the resolution
```

```
end

% Number of grid points
x_grid = m;
t_grid = m;

% Step size in space and time direction
dx = (x_max - x_min)/(x_grid-1); %step size in space
dt = (t_max - t_min)/(t_grid-1); %step size in time
```

Initial Condition

```
% Initial Condition
% u(x,0) = u0(x) = sin(2*pi*x)
syms x
f(x) = sin(2*x*pi); %function to calculate IC
Initial_cond = zeros(x_grid, 1); %vector holding IC in first row

% CFL condition
CFL = c*dt/dx; % c =1 (given)
```

FTBS Scheme

```
% FTBS coefficient Matrix
U = (1-CFL)*eye(x_grid) + CFL*diag(ones(x_grid-1, 1), -1);
U(1, x_grid) = CFL;  % Boundary value for U

% Cal values for 1st row using IC
Initial_cond(1, 1) = f(x_min);
for i=1:x_grid-1
   Initial_cond(i+1, 1) = f(i*dx+x_min);
end

% Marching the solution for t_grid times.
% Previous values are stored in U_old and current value is
% calculated and stored in U_new.
U_old = Initial_cond(:, 1);
for i=1:t_grid
    U_new = U * U_old;
    U_old = U_new;
end
```

Analytical Solution

```
% Cal the Analytic Sol at t =1.2
analytical = zeros(1, x_grid);
t = 1.2;
analytical(1, 1) = f(x_min-t);

% Calculating analytical solution using function f
% and looping x_Grid-1 times.
```

```
    for i=1:x_grid-1
        analytical(1, i+1) = f((i*dx+x_min)-t);
    end

    % Storing data in store matrix for calculating accuracy
    store(1,j) = analytical(1, n+(i-1));
    store(2,j) = U_new(n+(i-1), 1);

end
```

Accuracy Test

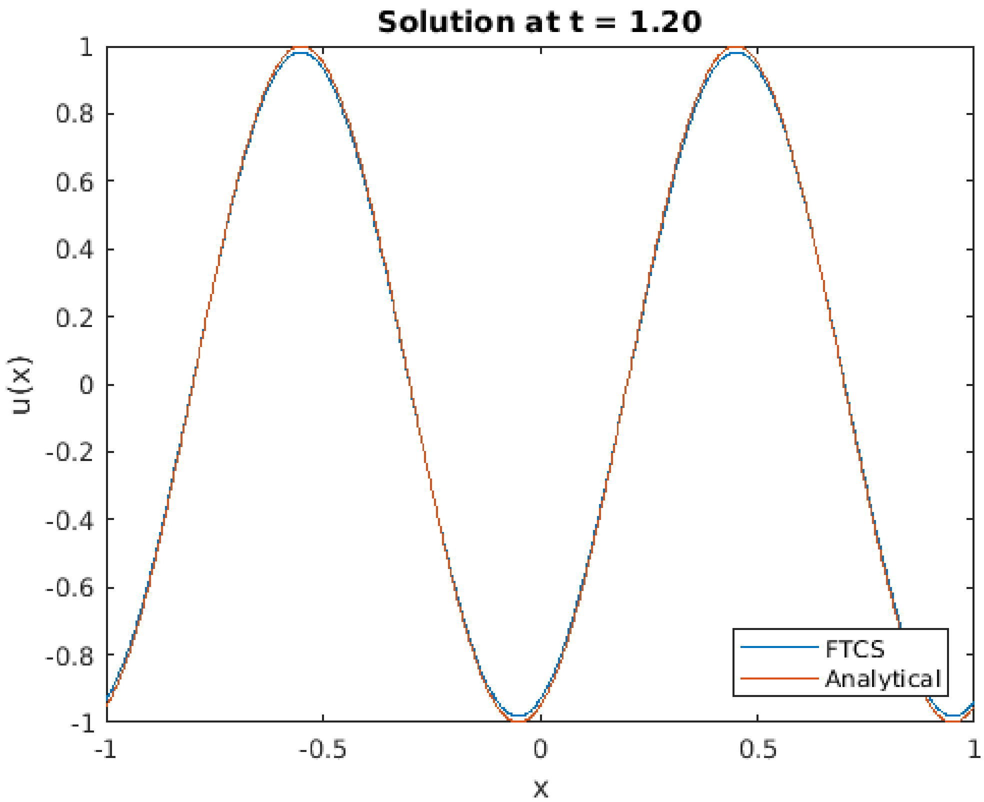
```
% Computing the order accuracy
accuracy = zeros(1, k-1);
for i=1:(k-1)
    accuracy(i) = log2(abs(((store(1,i)-store(2,i)))/(store(1, i+1)-
store(2, i+1))));
end
```

Plot

```
% Plot the Analytical and FTCS solution
x = x_min:dx:x_max;  %grid values on x-axis

plot(x, U_new');     % Numerical Solution
hold on
plot(x, analytical); % Analytical Solution
hold off

axis([-1 1 -1 1])
str = sprintf('Solution at t = %.2f', 1.2);
title(str)
xlabel('x')
ylabel('u(x)')
legend('FTCS','Analytical', 'Location', 'Southeast');
```



Discussion

```
% As we increas k, our grid size decrease and we get much finer and
 accurate plots.
% Also, with increase in k, we get accuracy close to 1.

% Eg: With k = 10, our order of accuracy is .99, which is very close
 to 1.
```

Published with MATLAB® R2017a