Computational Fluid Dynamics

Assignment Catalogue

AM5630 Assignment 2

Aakash R -ME14B149

February 19, 2017

Contents

1 Problem statement

Given a rod of Length **L** ,with boundary conditions,initial conditions as follows:

1.1 Boundary condition:

$$T(0,t) = 0°C \tag{1}$$
$$T(L,t) = 1°C \tag{2}$$

1.2 Initial condition:

$$T(x,0) = 0°C \tag{3}$$

Compute the temperature for t = 0s to 20s for various values of $\Delta T = 0.1s, 0.01s, 0.001s$.

2 Governing Equations

2.1 PDE

$$\frac{\delta T}{\delta t} = \alpha \frac{\delta^2 T}{\delta x^2} \tag{4}$$

2.2 Finite difference formulation using BTCS scheme (Implicit)

$$-\gamma T_{i-1}^{n+1} + T_i^{n+1}(2\gamma + 1) - \gamma T_{i+1}^{n+1} = T_i^n \tag{5}$$

3 Pseudo Code

1. Initialize the variables $\alpha, \Delta t, T, \Delta x, N_x, L$.
   (Note here T is a matrix with N_x columns ,and $20/(\Delta t) + 1 = N_y$ rows)

2. For n = 2 to N_y .

   (a) Solve the equation below for T^{n+1} using the **TDMA** algorithm.

   $$AT^{n+1} = T^n$$

   Where,

   $$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -\gamma & 2\gamma+1 & -\gamma & 0 & \cdots & 0 \\ 0 & -\gamma & 2\gamma+1 & -\gamma & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

   The **TDMA** consists of converting the Tridiagonal matrix to an upper triangular matrix , then solving the system of equations by back substitution.
   TDMA Algorithm:

   i. For i = 2:$(N_x - 1)$

      A. $A[i,:] = A[i,:] - A[i-1,:]\frac{A[i,i-1]}{A[i,i]}$

      B. $T[n-1,i] = T[n-1,i] - T[n-1,i-1]\frac{A[i,i-1]}{A[i,i]}$

   ii. Back substitution.

3. End

4 Results

The values of the variables used are $Length = 10m, \alpha = 0.5\frac{m^2}{s}$.
The below are the Computational time taken by the schemes to run the program for various grid points(N_x), Δt.

4.1 FTCS explicit scheme

$N_x \downarrow \vert \Delta \rightarrow t$	0.001 s	0.01 s	0.1 s
11	0.0156 s	0 s	0 s
21	0.0313 s	0 s	0 s
31	0.0313 s	0 s	0 s
41	0.0625 s	0 s	0 s
51	0.0156 s	0 s	0 s

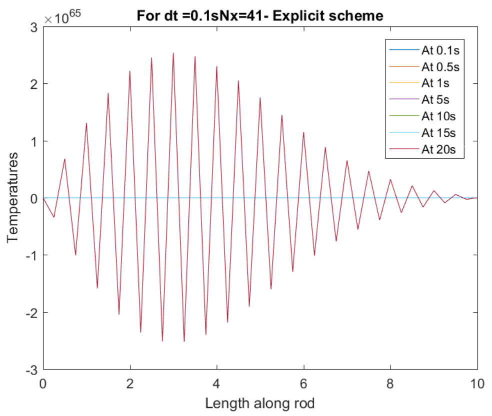
4.2 BTCS implicit scheme

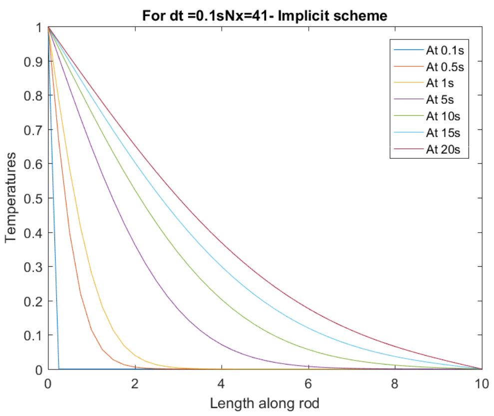
$N_x \downarrow \vert \Delta \rightarrow t$	0.001 s	0.01 s	0.1 s
11	0.5156 s	0.0625 s	0.0469 s
21	0.6875 s	0.0938 s	0.0313 s
31	1.0625 s	0.1250 s	0.0156 s
41	1.4844 s	0.1875 s	0.0938 s
51	1.8906 s	0.2188 s	0.0313 s

5 Graphs

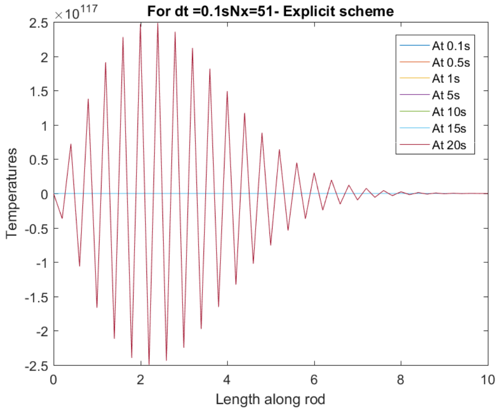
Below are **sample Graphs** showing the Unconditional stability of the Implicit schemes.

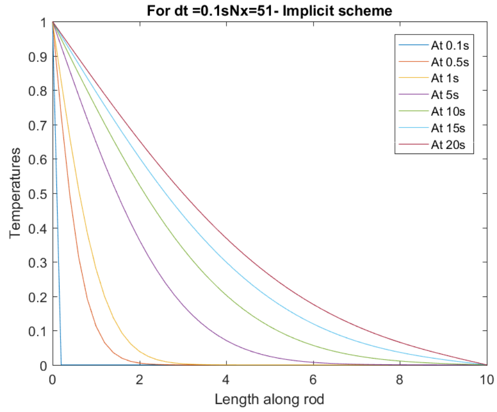
5.1 Nx=41 ,dt= 0.1s





5.2 Nx=51 ,dt= 0.1s





6 MATLAB Code

6.1 Contents - ME14B149_Input.m

- Input Variables
- Running both schemes , comparing time

Input Variables

```
close ;
clear ;
clc;


L = 10 ; % Length of rod in meters
t = 25;  % Max time of observation in seconds
alpha = 0.5; % SI units


Nx = 11:10:101;  % No of grid points in space
dt = [0.001,0.01,0.1];% Time differential in seconds.
```

Running both schemes , comparing time

```
for i=1:1:5
     for j=1:3
   e(i,j)= Assignment1( L,t,alpha,Nx(i),dt(j)); % Explicit time
   f(i,j)= Assignment2( L,t,alpha,Nx(i),dt(j)); % Implicit time
     end
end
```

6.2 Contents - Assignment2.m

- CFD Assignment 2 -Intro (BTCS scheme)
- Variable initialization -1
- BTCS scheme
- Plotting data for t = 0.1,0.5,1,5,10,15,20 s

```
function [ TotalTime ] = Assignment2(L,t,alpha,Nx,dt )
```

CFD Assignment 2 -Intro (BTCS scheme)

One dimentional unsteady heat conduction equation

```
close all;
```

Variable initialization -1

```
 dx = (L/(Nx-1)); % Distance differential in m

 m = round(t/dt); % No of grid points in time
 T = zeros(m,Nx); % Grid generation ,Initial condition

 T(:,1) = 1;  %Boundary condition
 T(:,Nx) = 0; %Boundary condition
```

BTCS scheme

dT/dt = alpha d2T/d2x Tn+1(i) = Tn(i) + alpha*dt(Tn+1(i+1)-2*Tn+1(i)+Tn+1(i-1))/dx2 Tn+1(i)-gamma*Tn+1(i+1)+2*gamma*Tn+1(i)-gamma*Tn+1(i-1)= Tn(i) -gamma*Tn+1(i-1)+Tn+1(i)(2*gamma+1) + -gamma*Tn+1(i+1) = Tn(i)

```
  t = cputime; % Calculating Time
 for n = 2:m
     M = zeros(Nx);
     M(1,1) =1/dt;
     M(Nx,Nx) =1/dt;

      % Matrix Construction
      for i = 2:(Nx-1)
      M(i,(i-1):(i+1)) =  [-alpha/(dx^2),1/dt+2*alpha/(dx^2),-alpha/(dx^2)];
      end
      % A = M\(T(n-1,:)'/dt); %CHECK 1

      % Upper triangular Matrix Conversion
      M = [M,(T(n-1,:)'/dt)];
      for i = 2:(Nx-1)
      M(i,:) = M(i,:) - M(i-1,:)*(M(i,i-1)/M(i-1,i-1));
      end

      %B = M(:,1:(end-1))\M(:,end);%CHECK 2


      % Solving
```

```
      X = M(:,1:(end-1));
      T(n,1)=1;
      T(n,Nx)=0;
      for i = (Nx-1):-1:(2)
      T(n,i) = ( M(i,end) - X(i,i+1)*T(n,i+1))/M(i,i);
      end

      %D(:,:,n)=M; %CHECK 3
 end
 % T(n,:)- B' %CHECK 4

 TotalTime = cputime - t; % Computational time
```

Plotting data for t = 0.1,0.5,1,5,10,15,20 s

```
     plot(0:dx:L , T(0.1/dt,:),0:dx:L , T(0.5/dt,:),0:dx:L , T(1/dt,:),0:dx:L , T(5/dt,:),0:dx
     xlabel('Length along rod')
     ylabel('Temperatures')
     legend('At 0.1s','At 0.5s','At 1s','At 5s','At 10s','At 15s','At 20s');
     s1 = num2str(dt);
     s2 = 'For dt =' ;
     s4= num2str(Nx);
     s3 = strcat(s2,s1,'s','Nx=',s4,'- Implicit scheme');
     title(s3);
      %pause;
print(strcat(s3,'.jpg'),'-dpng')

end
```

6.3 Contents - Assignment1.m

- CFD Assignment1 -Intro (FTCS scheme)
- Variable initialization -1
- CSFT scheme
- Plotting data for t = 0.1,0.5,1,5,10,15,20 s

```
function [ TotalTime ] = Assignment1( L,t,alpha,Nx,dt)
```

CFD Assignment1 -Intro (FTCS scheme)

One dimentional unsteady heat conduction equation

```
 close all;
```

Variable initialization -1

```
  dx = (L/(Nx-1)); % Distance differential in m

 m = round(t/dt+1); % No of grid points in time
 T = zeros(m,Nx); % Grid generation ,Initial condition

 T(:,1) = 1;  %Boundary condition
 T(:,Nx) = 0; %Boundary condition
```

CSFT scheme

dT/dt = alpha d2T/d2x Tn+1(i) = Tn(i) + alpha*dt(Tn(i+1)+2*Tn(i)+Tn(i-1))/dx2

```
 t = cputime;
 for n = 2:m
     for i = 2:(Nx-1)
      T(n,i) =  T(n-1,i) + alpha*dt*(T(n-1,i+1)-2*T(n-1,i)+T(n-1,i-1))/dx/dx;

     end
 end
 TotalTime = cputime - t;
```

Plotting data for t = 0.1,0.5,1,5,10,15,20 s

```
    plot(0:dx:L , T(0.1/dt+1,:),0:dx:L , T(0.5/dt+1,:),0:dx:L , T(1/dt+1,:),0:dx:L , T(5/dt+1,
     xlabel('Length along rod')
     ylabel('Temperatures')
     legend('At 0.1s','At 0.5s','At 1s','At 5s','At 10s','At 15s','At 20s');
     s1 = num2str(dt);
     s2 = 'For dt =' ;
     s4= num2str(Nx);
     s3 = strcat(s2,s1,'s','Nx=',s4,'- Explicit scheme');
     title(s3);
% pause


print(strcat(s3,'.jpg'),'-dpng')

end
```

7 Appendix B - Code Link

For complete set of the graphs please check the results folder in the below link.
Matlab Code,Complete set of graphs for all combinations of $N_x, \Delta t$ results :
https://github.com/RAAKASH/Intro-to-CFD-/tree/master/Assignment%202