Project Report

on

"Explicit and Implicit Method in Finite Difference Method"

by

Dipan Deb (16101012)

Jyotshna Bali (16101016)

Netrapal Singh (16101023)

Rahul Ranjan (16101034)

Shobhit Shrivastava (16101043)

Department Of Aerospace Engineering

IIT Kanpur

Introduction:

Most of the partial differential equations cannot be solved analytically, therefore, these equations are solved using numerical techniques with a considerable amount of error. Finite difference method is a type of numerical technique which makes use of the Taylor series expansion to convert the partial differential equation into a difference equation.

Problem Statement:

In this report, we are going to consider a 1D, unsteady, heat conduction equation with constant thermal diffusivity and zero source term.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

As the above equation is parabolic in nature with marching variable *t*, the running index for this will be denoted by n as a superscript. For the space variable the index be denoted by i as subscript. Now, by using Taylor series expansion the difference equation can be written as,

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{(T_{i+1}^n - 2T_i^n + T_{i-1}^n)}{(\Delta x)^2}$$

where we have neglected the higher order terms which is known as truncation error. The above difference equation represents the original partial differential equation. However it may be noted that it is just an approximation to our original differential equation.

Explicit Method:

With some rearrangement the difference equation can be rewritten as

$$T_i^{n+1} = T_i^n + \alpha \frac{\Delta t}{(\Delta x)^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

In the above equation we have kept the n+1 level values on the left hand side and the values at level n on the right hand side. Let us assume we know the values of T at n-level at all the grid points. Using the above equation we can then calculate the values of T at n+1 level at all grid points. So now when we have the values of T at n+1 level we can thus proceed further in the same manner to calculate the T at n+2 level and so on. One important thing to be noted here is that we also know the boundary conditions which means we know the T values at the boundary of the next level. Thus in this manner we can calculate the entire solution by marching in time.

We have one unknown corresponding to one equation which can be solved explicitly. This approach of solving a differential equation is termed as explicit method.

Implicit Method:

The above mentioned difference equation was not the only way to discretize the partial differential equation. Now, taking the average properties of the right hand side values between time levels n & n+1, the equation will be

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{(\Delta x)^2}$$

This difference equation is called Crank-Nicolson method. In the above equation we have 3 unknowns namely T_i^{n+1}, T_{i+1}^{n+1} and T_{i-1}^{n+1} at level n+1. When this equation is applied at a grid point *i*, it cannot result in a solution for T_i^{n+1}. Rather we must apply this equation at all interior grid points, thus giving us a system of algebraic equations which in need to be solved simultaneously. This approach of solving difference equation is known as implicit method.

Rearranging the above equation,

$$-CT_{i-1}^{n+1} + (1-2C)T_i^{n+1} - CT_{i+1}^{n+1} = T_i^n$$

Let us define, $C = \frac{\alpha \Delta t}{(\Delta x)^2}$

$$A = -C\ ;\ \ B = 2C - 1\ ;\ \ K_i = T_i^n$$

Thus, substituting the above in the equation, we get

$$AT_{i-1}^{n+1} - BT_i^{n+1} + AT_{i+1}^{n+1} = K_i$$

Using the equation at 7 grid points, the system of equation can be written in a matrix form as below.

$$AT_1 - BT_2 + AT_3 = K_2$$

This equation can be written as,

$$-BT_2 + AT_3 = K_2'$$

where, $KT_2 - AT_1 = K_2'$

$$AT_2 - BT_3 + AT_4 = K_3$$

$$AT_3 - BT_4 + AT_5 = K_4$$

$$AT_4 - BT_5 + AT_6 = K_5$$

$$AT_5 - BT_6 + AT_7 = K_6$$

Again writing this equation as

$$AT_5 - BT_6 = K_6'$$

where, $K_6 - AT_7 = K_6'$

In the above set of equations, $A, B, K_2', K_6',\ T_{1,}\ T_7$ are known variables.

Combining the above equations in a matrix from

$$\begin{bmatrix} -B & A & 0 & 0 & 0 \\ A & -B & A & 0 & 0 \\ 0 & A & -B & A & 0 \\ 0 & 0 & A & -B & A \\ 0 & 0 & 0 & A & -B \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} K'_2 \\ K_3 \\ K_4 \\ K_5 \\ K'_6 \end{bmatrix}$$

The coefficient matrix is a tri-diagonal matrix.

Error and Analysis of Stability:

Stability of a numerical method means in every marching step the numerical error should decrease. For the above problem let,

D = Exact solution of difference equation

N = Numerical solution from a real computer with finite accuracy

So, N = D+ε, where ε is the round-off error. There is another type of error called discretization which is the difference between analytical solution of the partial differential equation and that of difference equation. Now, putting the value of D & N we can find that the error also satisfies the difference equation i.e.

$$\frac{\varepsilon_i^{n+1} - \varepsilon_i^n}{\Delta t} = \alpha \frac{(\varepsilon_{i+1}^n - 2\varepsilon_i^n + \varepsilon_{i-1}^n)}{(\Delta x)^2}$$

Now, from the definition of stability, for the solution to be stable

$$\left|\frac{\varepsilon_i^{n+1}}{\varepsilon_i^n}\right| \le 1$$

The error is a function of space & time both. With time the error may exponentially decay or increase. But, with space the error varies like this.

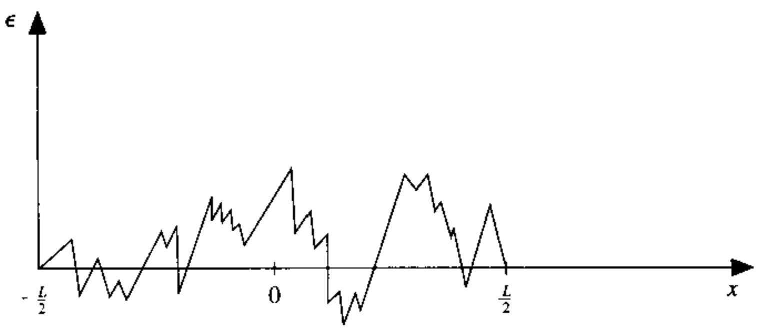


Figure 1 : Round off error v/s x

This round-off error graph is only for a given time step. at -L/2 & L/2 the error is zero because of the known boundary conditions. Now, the round-off error can be analytically written like this

$$\varepsilon(x,t) = \sum_{m=1}^{N/2} e^{at} e^{ik_m x}$$

Where, a is constant, N=no. of elements in the mesh, k_m is the no. of full sine waves at a given length.

The behaviour of each term of the series is similar to the series itself. So, we will consider one term of the series and put it in the difference equation. So, we will get,

$$\frac{e^{a(t+\Delta t)} e^{ik_m x} - e^{at} e^{ik_m x}}{\alpha \Delta t} = \frac{e^{at} e^{ik_m (x+\Delta x)} - 2e^{at} e^{ik_m x} + e^{at} e^{ik_m (x-\Delta x)}}{(\Delta x)^2}$$

Now, rearranging the equation we can show that,

$$e^{a\Delta t} = 1 - \frac{4\alpha\Delta t}{(\Delta x)^2} sin^2 \frac{k_m x}{2}$$

And, $\left|\frac{\varepsilon_i^{n+1}}{\varepsilon_i^n}\right| = e^{a\Delta t}$. For stability -1≤ $e^{a\Delta t} \leq 1$. For $e^{a\Delta t} \leq 1$ it is clearly understood because $\frac{4\alpha\Delta t}{(\Delta x)^2} sin^2 \frac{k_m x}{2}$ term is always positive. But, for -1≤ $e^{a\Delta t}$ condition to hold $\frac{\alpha\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$ should always satisfy.

This gives the stability requirement for the difference equation to solve.

Let the quantity $\frac{\alpha\Delta t}{(\Delta x)^2}$ be denoted by C.

Example:

Let a rod of length 10cm of uniform temperature 30^0C is placed in an environment where ambient temperature is 250^0C. Here temperature variation only along the length is considered which made this problem one dimensional unsteady heat conduction problem.
At first the Explicit method is used to solve the problem. For different values of C different Temperature v/s length graph are plotted.

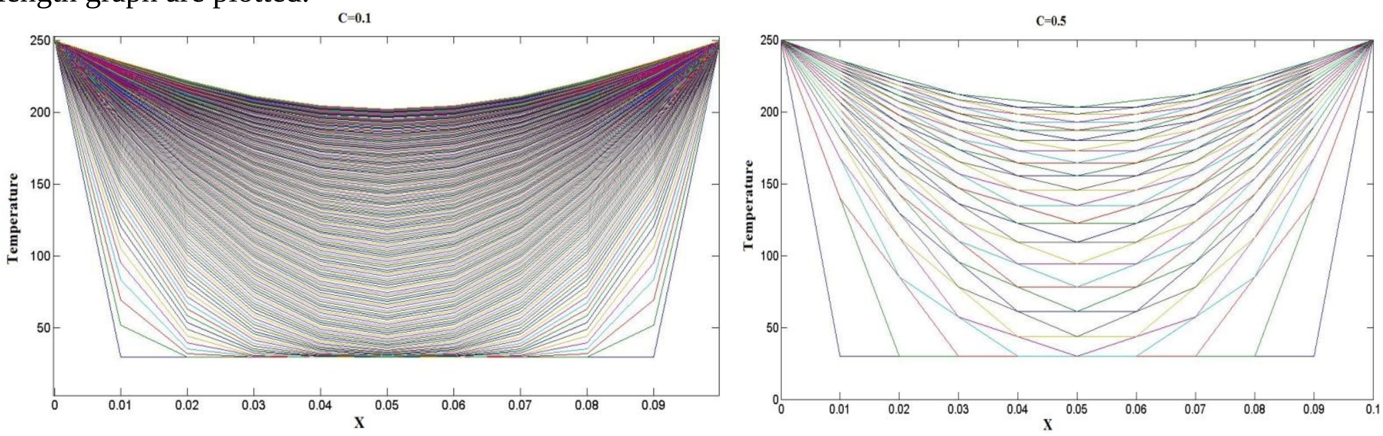


Figure 2 : Temperature v/s Length graph for the C=.1 and C=0.5

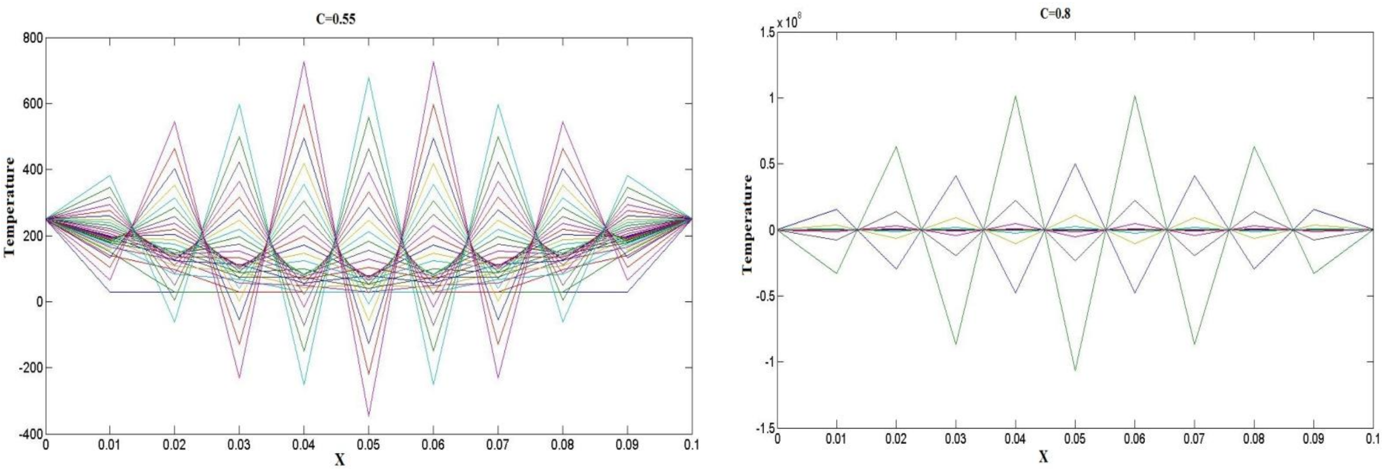
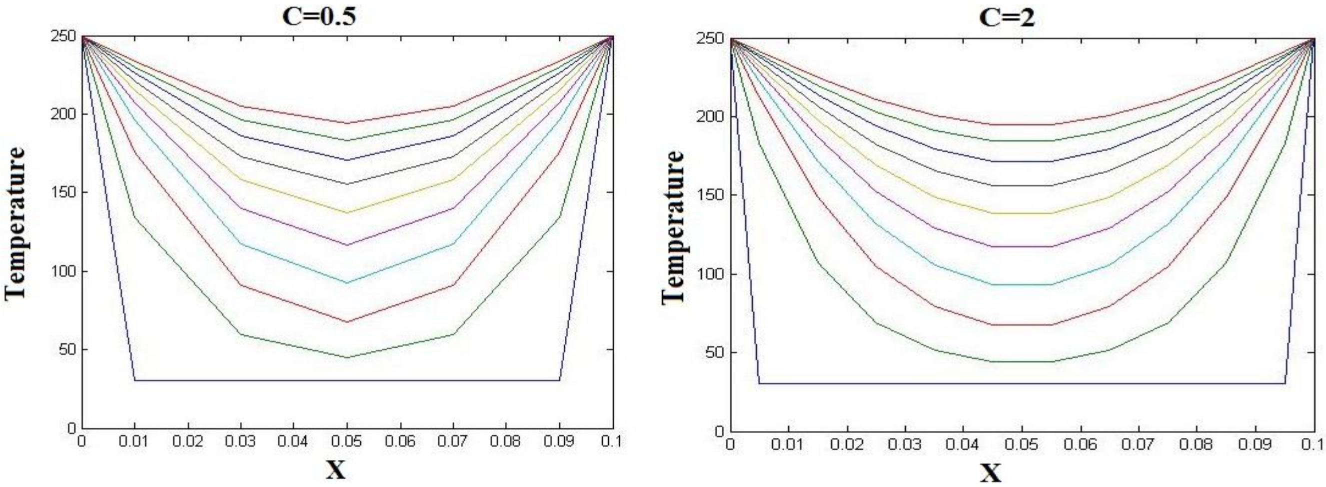


Figure 3: Temperature v/s Length graph for the C=.55 and C=0.8

The above stability analysis is for explicit method. For, implicit method solution is unconditionally stable.



Conclusion:

Explicit Schemes, allows the calculation of a flow field at an appropriate step from calculations, obtained from previous steps. Since the discretization schemes normally involve some sort of truncation error, the error will accumulate over each step. Hence explicit schemes are known to be stable for a limited step size and due to the accumulation of the error over each time step, normally over estimate the final solution. Implicit methods gain their time-step independence by introducing diffusive effects into the approximating equations.

References:

1. Computational Fluid Dynamics- The Basics with Application - J.D. Anderson
2. Numerische Methoden 1 – B.J.P. Kaus
3. Finite Difference Method – Mark Davis, Imperial College of London

Appendix:

a) Code for explicit method

```
clear all
clc
%%Initialisation
%T_inf = input('Enter the temperature of the surrounding: ');
%T1 = input('Enter the initial temperature of the plate: ');
T_inf = 250;
T1= 30;
%K = input('Enter the thermal conductivity of the slab: ');
%Rho = input('Enter the density of the material: ');
%cp = input('Enter the specific heat value of the slab: ');
%alpha = K/(Rho*cp);
%fprintf('The thermal diffusivity of the material is ');
%disp(alpha);
alpha= input('The thermal diffusivity of the material:');
L= input('Enter the length of the plate: ');
t= input('Enter the value of time till which the readings are to be taken(in seconds):
');
m= input('Enter the number of the time interval: ');
n= input('Enter the number of the interval: ');
h= L/n;
h1= t/m;
a=zeros(n+1,1);
for i=1:n+1
    if(i==1)
        a(i)=0;
    else
        a(i)= a(i-1)+h;
    end
end
b=zeros(m+1,1);
for i=1:n+1
    if(i==1)
        b(i)=0;
    else
        b(i)= a(i-1)+h1;
    end
end
g=(alpha*h1)/(h^2);

T=zeros(m+1,n+1);
for i=1:m+1
    for j=1:n+1
        if (i==1)
            if (j==1||j==n+1)
                T(i,j)= T_inf;
            else
                T(i,j)=T1;
            end
        else
            if (j==1||j==n+1)
                T(i,j)= T_inf;
            else
                T(i,j)= T(i-1,j)+ (g*(T(i-1,j+1)+ T(i-1,j-1)-(2*T(i-1,j))));
            end
        end
    end
```

```
end
disp(T);
fprintf('The value of C is ');
disp(g);
plot(a,T);
xlabel('x')
ylabel('Temperature')
set (gca,'fontsize',14);
```

b) Code for implicit method

```
clc;
clear all
tend=250;                %temp of the end plates
ti=30;                   %initial temp of the plastic sheet
% k=0.25;                   %value of heat conductivity
% ro=1300;                 %density of material
% cp=2000;                 %heat coefficient
alfa=0.00002;
dtim=10;                 %time elapsed for each calculation
timlst=90;              %time till which the readings are to be taken
gr=(timlst/dtim)+1;          %grid rows
t=0.10;                    % total thickness of plastic sheet in meters
gc=10;                   %grid numbers /coulmn
dxp=t/gc;                %grid size
dxend=(dxp/2);           %distance of starting and ending points on plastic from the
tips for first and the nth grid
alfa
2*alfa*dtim/(dxp^2)+1
alfa*dtim/(dxp^2)
temp=zeros(gr,gc+2);
temp(1,:)=ti;
temp(:,1:gc+1:gc+2)=tend;
temp
%-------------------------------------------------------------------%
for l=1:gr-1
    a=wilkinson(gc)
    b=ones(gc,1);
    for m=1:gc                            %For Inner Nodes

    a(m,m)=((-2*alfa*dtim/(dxp^2))-1);

    if (m+1)>gc                           % to avoid addition of rows in matrix
            break
    end
    a(m,m+1)=(alfa)*dtim/(dxp^2);
    a(m+1,m)=(alfa)*dtim/(dxp^2);

    end
    a
    a(1,1)=((-4*alfa*dtim/(dxp^2))-1);
     a(1,2)=(4*alfa)*dtim/3/(dxp^2);
    a(gc,gc)=a(1,1);

   a(gc,gc-1)=(alfa)*4*dtim/3/(dxp^2);

    a
   for n=1:gc
```

```
   b(n,1)=(-temp(l,n+1));
   end

      b(1,1)=(-temp(l,2))-(temp(l+1,1)*8*alfa*dtim/3/(dxp^2));
   b(gc,1)=(-temp(l,gc+1))-(temp(l+1,gc+2)*8*alfa*dtim/3/(dxp^2));

   b
    x=a\b
    for o=1:gc
    temp(l+1,o+1)=x(o,1);
    end
    temp


end
 p=[0,dxp/2:dxp:(t-dxp/2),t];
disp(p);
v=p'
plot(v,temp);
```