DEPARTMENT OF MECHANICAL ENGINEERING AT IIT TIRUPATI
ME 5101 COMPUTATIONAL FLUID DYNAMICS

Homework Assignment 3, Due before 9 AM on Mon, 18 Feb 2019 (Submit to Jivrakh)
(Late Submissions will not be accepted)

<u>(Please print out this sheet and attach to the front of your completed Assignment)</u>

Name (in full, upper case, write legibly) Aakash

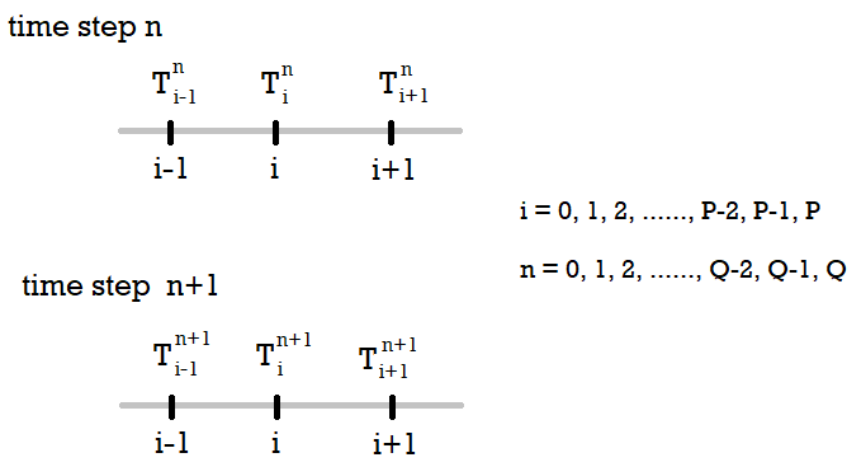
Roll Number ME16B001

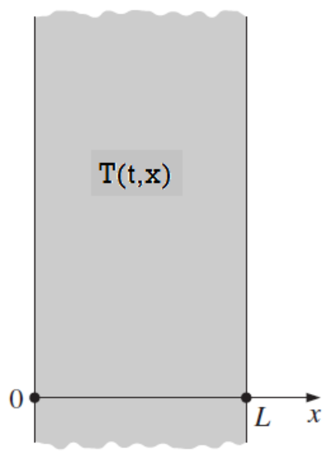
Instructions

1. Use A4 size paper only with proper margins
2. Answer all questions in detail
3. Graphs should be in pencil, or must be made using software, printed out & attached
4. Leave enough space between lines (Double spacing preferred)
5. Label all plots, sketches and graphs properly. Include correct units.
6. Sloppy work will attract severe penalty. Marks may be deducted from overall score.
7. **Plagiarism** will lead to FAILURE (U GRADE) in this course

(Do not get disappointed on seeing the size of this assignment. Once you are ready with your program, it is easy to get the graphs!)

Consider a large planar wall of width (or thickness) L = 0.24 m and uniform cross sectional area A = 8 m^2 (this area is perpendicular to the x-axis). The conductivity, density and specific heat of the material of the slab are k = 0.6 W/m-K, ρ = 3000 kg/m^3 and C = 0.2 J/kg-K respectively. There is no internal heat generation in the wall. Initially at t = 0, the entire wall is at a uniform temperature of 25ºC. Immediately after t = 0, the Temperatures of the left (x = 0) and right (x = L) ends of the wall are increased to 100ºC and 75ºC respectively. You are required to analyze the heat transfer in the wall and to obtain the time-evolution of the temperature profile in the wall using numerical techniques. Temperature profile means T(x); i.e, Temperature as a function of x.


time step n
T^n_{i-1} T^n_i T^n_{i+1}
i-1 i i+1
i = 0, 1, 2, ..., P-2, P-1, P
n = 0, 1, 2, ..., Q-2, Q-1, Q
time step n+1
T^{n+1}_{i-1} T^{n+1}_i T^{n+1}_{i+1}
i-1 i i+1



1. What is the governing equation for this problem? List the boundary and/or initial conditions.

2. Derive the governing equation (you may refer to some standard text)

3. Consider the stencil shown above for three nodes i-1, i and i+1 at two different time steps n and n+1. Discretize the governing equation using central differencing for space, and (a) explicit differencing for time (b) implicit differencing for time and (c) Crank Nicolson technique for time. For all three cases, use a unfiorm grid with $x_{i+1} - x_i = h$ for all values of i and $t^{n+1} - t^n = s$ for all values of n. Use $r = \alpha s/h^2$.

4. WAP to find the temperature profile in the wall at any given instant of time, $t = t_{interim}$. Your program should accept some input and produce the necessary output. **<u>Print out your code and attach it</u>**!!
Input: $t_{interim}$; method; h; s {Here the variable method is an integer which can take values 1, 2 or 3; method = 1 implies explicit; method = 2 implies implicit and method = 3 implies CN}.
Output: A vector $[T_0, T_1, T_2, \ldots, T_{P-2}, T_{P-1}, T_P]$ at $t = t_{interim}$ using the value of "method" which was given as input.

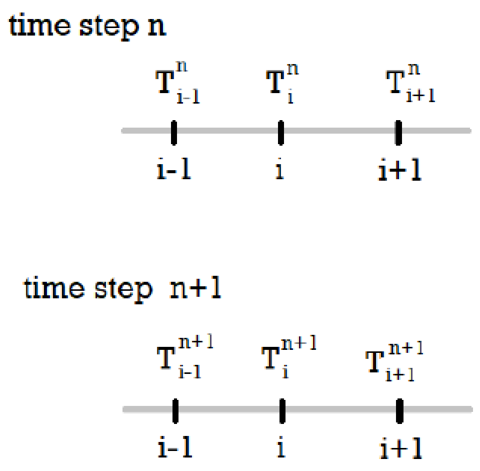
5. Find T(x) for the following values using your code. Plot the resulting curve, T vs x for all the cases in 18 different graphs. (Label axes properly with units). Please arrange your 18 graphs in 3 pages. Each page should contain 6 graphs. Page 1 should have graphs for cases (a-f); page 2 for (g-l) and page 3 for (m-2). In each page, arrange graphs in a matrix of 2 columns and 3 rows. Calculate the value of r for each case and list it on that respective graph. The label for each graph should be s = _____ , method = _____ , and r = ____. Comment on your results. See if you are obtaining realistic values for temperature in all the 18 cases listed below.

   a. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.010 s , method = 1 (explicit)
   b. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.025 s , method = 1 (explicit)
   c. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.040 s , method = 1 (explicit)
   d. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.060 s , method = 1 (explicit)
   e. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.075 s , method = 1 (explicit)
   f. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.120 s , method = 1 (explicit)
   g. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.010 s , method = 2 (implicit)
   h. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.025 s , method = 2 (implicit)
   i. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.040 s , method = 2 (implicit)
   j. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.060 s , method = 2 (implicit)
   k. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.075 s , method = 2 (implicit)
   l. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.120 s , method = 2 (implicit)
   m. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.010 s , method = 3 (CN)
   n. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.025 s , method = 3 (CN)
   o. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.040 s , method = 3 (CN)
   p. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.060 s , method = 3 (CN)
   q. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.075 s , method = 3 (CN)
   r. $t_{interim}$ = 6 s , h = 0.01 m , s = 0.120 s , method = 3 (CN)

6. Now modiy your program to find the time-evolution of temperature in the wall. For this purpose, the input will be $\{t_m\}_{m=0}^{10}$; method; h; s. Output should be T vs x for all values of $t = t_1, t_2, t_3, \ldots\ldots t_9, t_{10}$. Plot all the curves in a single graph. (Label axes properly with units). Also derive the steady-state solution and plot it in the same graph for reference. Does the wall reach steady state? If so, can you find out? For this problem, use h = 0.012 m , s = 0.05 s , method = 1 (explicit) and $t_0 = 0$; $t_1 = 1$ s ; $t_2 = 2$ s ; $t_3 = 5$ s ; $t_4 = 10$ s ; $t_5 = 20$ s ; $t_6 = 40$ s ; $t_7 = 50$ s ; $t_8 = 65$ s ; $t_9 = 80$ s ; $t_{10} = 100$ s.

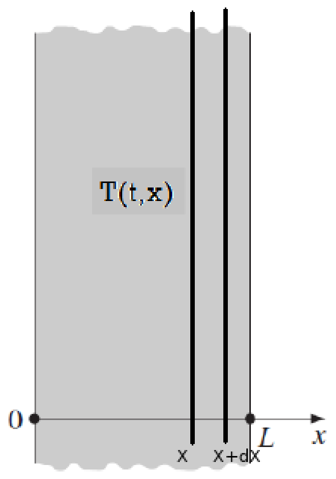
Aakash-ME16B001

1-D Transient Heat Conduction


time step n
T^n_{i-1} T^n_i T^n_{i+1}
i-1 i i+1
time step n+1
T^{n+1}_{i-1} T^{n+1}_i T^{n+1}_{i+1}
i-1 i i+1

i = 0, 1, 2, ..., P-2, P-1, P

n = 0, 1, 2, ..., Q-2, Q-1, Q



Given data - Thickness,$L = 0.24m$; Area,$A = 8m^2$; Conductivity,$k = 0.6W/m-k$; Density, $\rho = 3000Kg/m^3$; Specific heat,$C = 0.2KJ/Kg-K$ At $t = 0$ $T(t=0) = 25^\circ C$ Immediately after t = 0 $T(x=0) = 100^\circ C, T(x=L) = 75^\circ C$

Governing Equation

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t}$$

where,

$$T \text{ is the temperature in } {}^\circ C$$
$$x \text{ is the distance in meters}$$
$$t \text{ is the time in seconds}$$

α is the thermal diffusivity

$$\alpha = \frac{\kappa}{\rho C}$$

Derivation of the Governing equation

Consider the element of width δx as shown in the digram

$$\text{change of heat energy from the segment in time } \Delta T = \text{heat in from left boundary} + \text{heat inpu}$$

$$C \times \rho A \Delta x\,[T(x,t+\Delta t) - T(x,t)] = \Delta t\left(-kA\frac{dT}{dx}\right)_x - \Delta t\left(-kA\frac{dT}{dx}\right)_{x+\Delta x}$$

$$\frac{T(x,t+\Delta t) - T(x,t)}{\Delta t} = \frac{k}{\rho C}\frac{\left(\frac{dT}{dx}\right)_{x+\Delta x} - \left(\frac{dT}{dx}\right)_x}{\Delta x}$$

In the limit $\Delta x \to 0$ and $\Delta t \to 0$

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

Discretization

1. Explicit Scheme

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \left[\frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}\right]$$

Using $r = \frac{\alpha s}{h^2}$

$$T_i^{n+1} - T_i^n = r\left[T_{i+1}^n - 2T_i^n + T_{i-1}^n\right]$$

1. Implicit Scheme
$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \left[\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta x^2}\right]$$
$$T_i^{n+1} - T_i^n = r\left[T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}\right]$$
2. Crank Nicholson Scheme
$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \left[\frac{T_{i+1}^{n+1} + T_{i+1}^n - 2T_i^{n+1} - 2T_i^n + T_{i-1}^{n+1} + T_{i-1}^n}{2\Delta x^2}\right]$$
$$T_i^{n+1} - T_i^n = \frac{r}{2}\left[T_{i+1}^{n+1} + T_{i+1}^n - 2T_i^{n+1} - 2T_i^n + T_{i-1}^{n+1} + T_{i-1}^n\right]$$

Program

For explicit scheme

$$T_i^{n+1} = rT_{i-1}^n + (1-2r)T_i^n + rT_{i+1}^n$$

For n=0

$$T_i^0 = 25 \text{ for all } 1 \le i \le p$$
$$T_i^1 = rT_{i-1}^0 + (1-2r)T_i^0 + rT_{i+1}^0$$

1. For i=0
$$T_0^1 = rT_{-1}^0 + (1-2r)T_0^0 + rT_1^0$$
Assuming that temperature at node -1 is equal to that of node 0 in the program.
2. For i=1
$$T_1^1 = rT_0^0 + (1-2r)T_1^0 + rT_2^0$$
3. For i=2
$$T_2^1 = rT_1^0 + (1-2r)T_2^0 + rT_3^0$$

For n=1

$$T_i^2 = rT_{i-1}^1 + (1-2r)T_i^1 + rT_{i+1}^1$$

1. For i=0
$$T_0^2 = rT_{-1}^1 + (1-2r)T_0^1 + rT_1^1$$
2. For i=1
$$T_1^2 = rT_0^1 + (1-2r)T_1^1 + rT_2^1$$
3. For i=2
$$T_2^2 = rT_1^1 + (1-2r)T_2^1 + rT_3^1$$

In [4]:

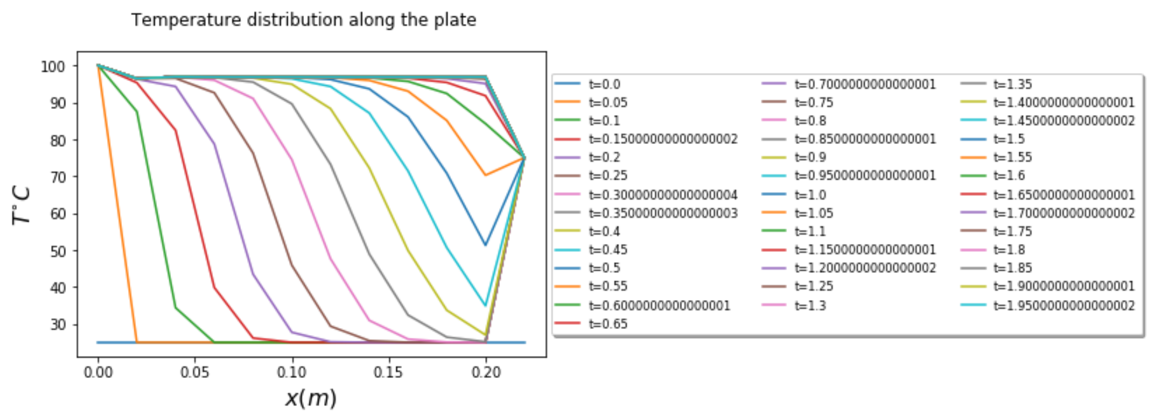
```
from sympy import *
import matplotlib.pyplot as plt
import numpy as np
from numpy import *
from array import *
from scipy.sparse import *

#  (r,c) => (n,i) \\ (q,p)  || (time,space)
L=0.24
alpha=0.001
def Temp(scheme,tinter,s,h):
    T=[[]]
    q=tinter/s
    p=L/h
    r=(alpha*s)/(h*h)

    for i in range(int(p)):
        T[0].append(25)
    # explcit scheme
    if scheme==1:
        for n in range(1,int(q)):
            T.append([])
            for i in range(0,int(p-1)):
                if i==0:
                    # asssumption that temperature at node i=-1 is equal to left
end
                    T[n].append( r*T[n-1][i] + (1-2*r)*T[n-1][i] + r*T[n-1][i+1]
)
                if i==int(p-1):
                    # asssumption that temperature at node i=21 is equal to righ
tmost end
                    T[n].append(r*T[n-1][i-1] + (1-2*r)*T[n-1][i] + r*T[n-1][i]
)
                else:
                    T[n].append(r*T[n-1][i-1] + (1-2*r)*T[n-1][i] + r*T[n-1][i+1
] )
            # boundary conditions
            T[n][0]=100
            T[n][int(p-1)]=75
    # if scheme==2:
        # TODO: implemtnt implicit scheme
    # if scheme==3:
        # TODO: implemtnt CN scheme
    return(T)

#  inputs for calling the function (scheme,time interim,time step, space step)
method=1
tin=2
s=0.05
h=0.02
# print(Temp(method,tin,s,h))
A=Temp(method,tin,s,h)
l=[]
for i in range(int(L/h)):
    l.append(i*h)
for i in range(int(tin/s)):
    plt.plot(l,A[i],label="t="+str(i*s))

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5),ncol=3, fancybox=True, sha
dow=True,fontsize='small')
plt.xlabel(r'$x (m)$', fontsize=16)
plt.ylabel(r'$ T^{\circ}C $', fontsize=16)
plt.suptitle('Temperature distribution along the plate')
plt.show()
```



In [ ]:

```
```