1. INTRODUCTION

In this project, 2-D heat transfer of a plate is analyzed with MATLAB. The method is selected as Finite Volume Method (FVM) with Tri-diagonal Matrix Algorithm. The code is first written with respect to the problem which is solved in Patankar's book and after seeing that the code is working, convection term is added to the code.

2. POBLEM DEFINITION

As seen on figure below, 2D steady heat conduction with heat source is modeled on a rectangular domain by FVM using MATLAB programing language. Grid size and size of domain and boundary conditions are input to the problem. Also, that domain have small amount of heat generation.

- South side of the domain modeled as heat-flux boundary conditions (in or out).
- East side modeled as constant temperature boundary condition.
- West side is insulated.
- North side assumed heat loss boundary condition defined with a surrounding fluid temperature.

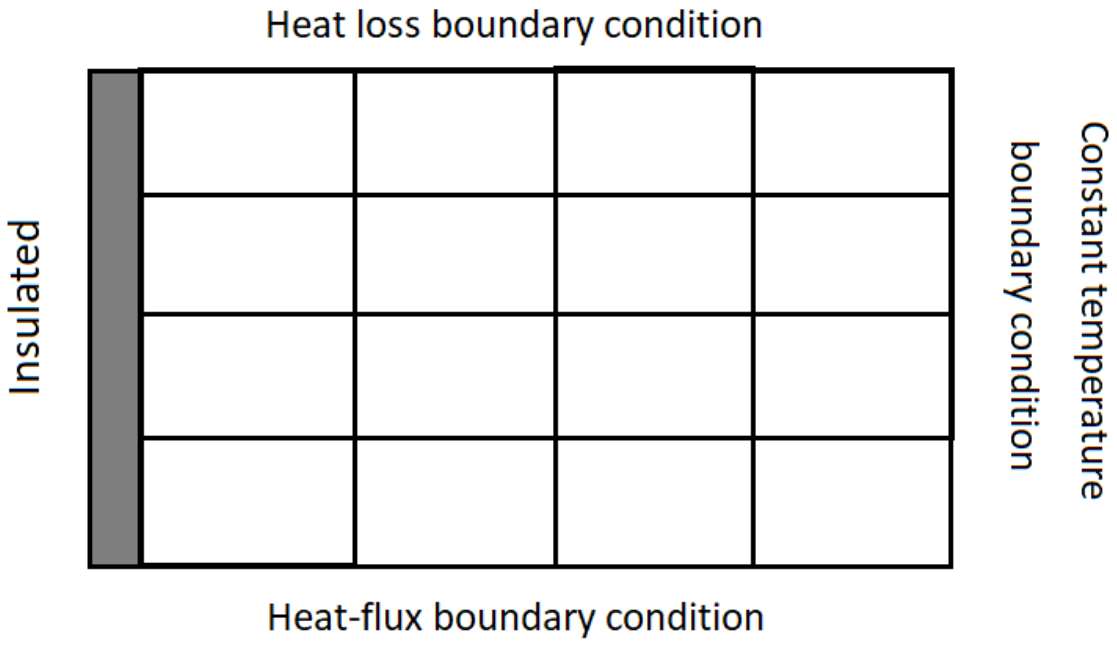

Figure 1 Problem Demonstration

3. INPUTS

Input variables can be entered by the user in MATLAB user interface “Command Window”. The meanings of the inputs are defined below.

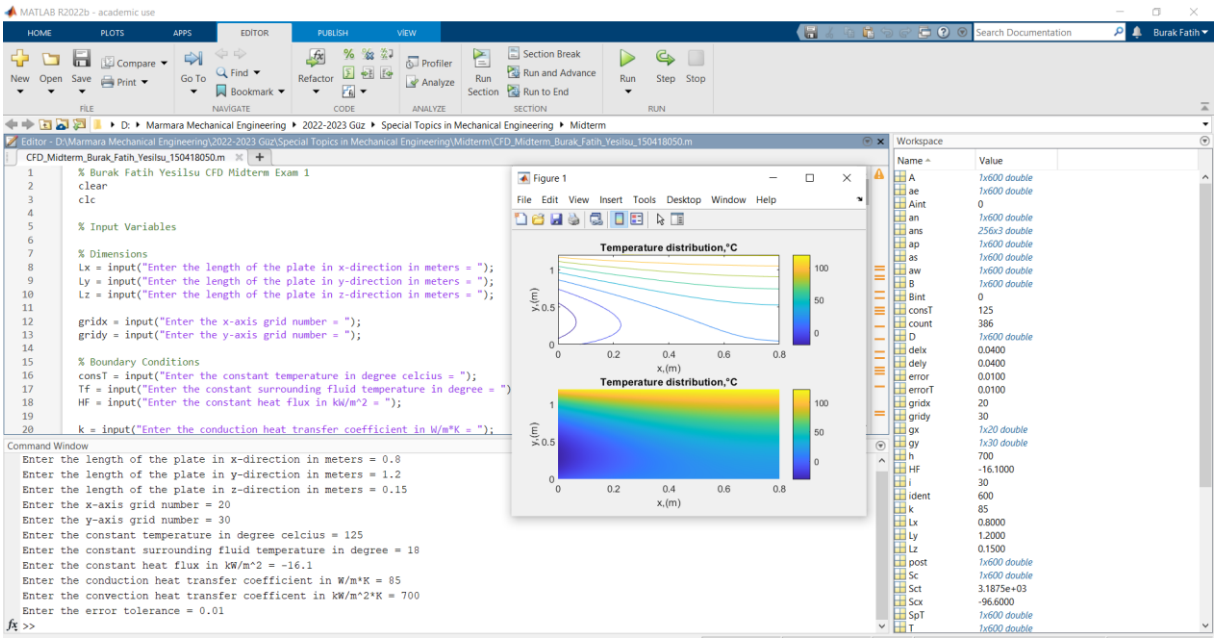

Figure 2 MATLAB Program interface

3.1. DIMENSIONS

Lx : The length of the plate in x-direction in meters.

Ly : The length of the plate in y-direction in meters.

Lz : The length of the plate in z-direction in meters (Thickness).

gridx : Number of cells in x-direction.

gridy : Number of cells in y-direction.

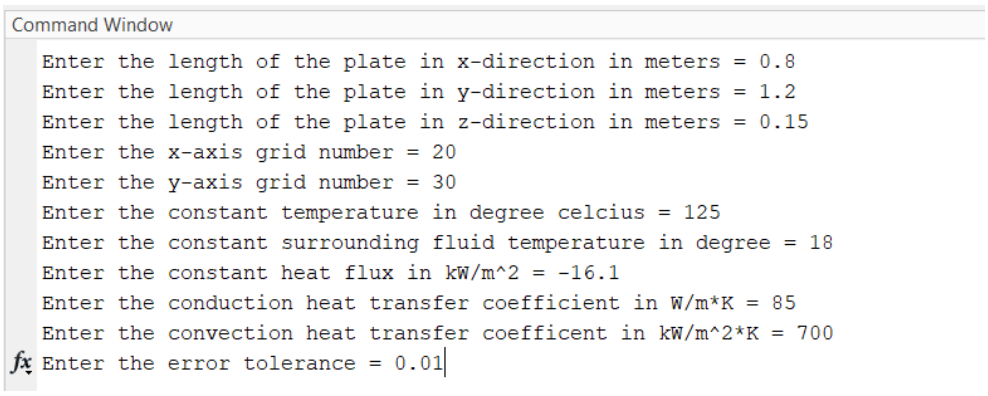

Figure 3 MATLAB Inputs in Command Window

3.2. BOUNDARY CONDITIONS

consT : Constant temperature in $°C$ (East).

Tf : Constant surrounding fluid temperature in $°C$ (North).

Insulated : Boundary condition at West side (no parameter).

HF : Constant heat flux in kW/m^2 (South).

k : Conduction heat transfer coefficient of plane in W/mK.

h : Convection heat transfer coefficient of surrounding fluid in kW/m^2K.

3.3. CALCULATION

errorT : Implies maximum error that can be tolerated.

To calculate the temperature values, the program uses finite volume method (FVM) with alternating direction implicit (ADI) method. First creates mesh grids, then using FVM with ADI sweeps from bottom to top starting from left bottom corner. When reaches to top right corner starts sweeping from left to right from left bottom corner and calculates temperature values for each cell. Program continuous this process until converge accuracy reached.

4. RESULTS

Results of the calculation are shown as temperature contour in $°C$. Color bar at the right shows the corresponding color scale for each grid's temperature.

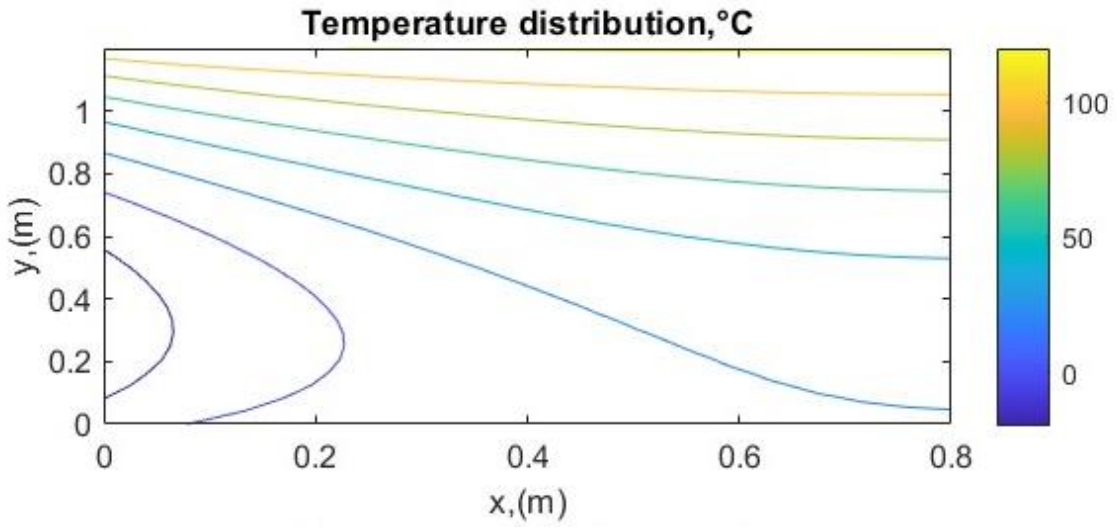

Figure 4 Temperature Distributions

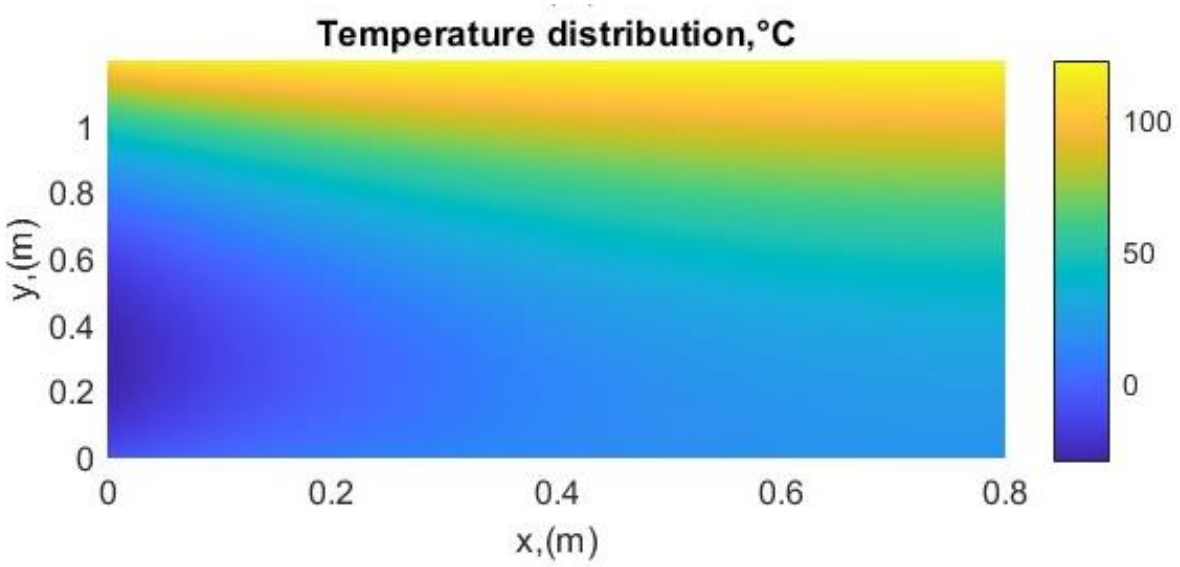

Figure 5 Temperature Distribution (Colorful format)

5. CODING OF THE PROGRAM

```
% Burak Fatih Yesilsu CFD Midterm Exam 1
clear
clc

% Input Variables

% Dimensions
Lx = input("Enter the length of the plate in x-direction in meters = ");
Ly = input("Enter the length of the plate in y-direction in meters = ");
Lz = input("Enter the length of the plate in z-direction in meters = ");

gridx = input("Enter the x-axis grid number = ");
gridy = input("Enter the y-axis grid number = ");

% Boundary Conditions
consT = input("Enter the constant temperature in degree celcius = ");
Tf = input("Enter the constant surrounding fluid temperature in degree =
  ");
HF = input("Enter the constant heat flux in kW/m^2 = ");

k = input("Enter the conduction heat transfer coefficient in W/m*K = ");
h = input("Enter the convection heat transfer coefficent in kW/m^2*K = ");

% Calculation
errorT = input("Enter the error tolerance = ");


delx = Lx / gridx;
dely = Ly / gridy;

x = gridx;
y = gridy;
xmy = round(x*y);
```

```
Tbef = zeros(1,xmy);
count = 0;
Tint = zeros(1,x*y);
error = 100;

while error > errorT
    for ident = 1:1:xmy
        if ident <= y
            SpT = zeros(1,y);
            x3 = ((k*dely*Lz)/delx);
            x2 = ((k*delx*Lz)/dely);
            Sct = 2*x2*consT;
            aw =zeros(1,y);
            ae = x3*ones(1,y);
            as = [0 x2*ones(1,y-1)];
            an = [x2*ones(1,y-1) 0];
            SpT(y) = -Sct/consT;
            SpT(1) = -h*delx*Lz;
            ap = aw+as+ae+an-SpT;
            Scx = HF*10^3*dely*Lz;
            Sc = Scx*ones(1,y-1);
            Sc(y)=Sct+Scx;
            Sc(1)=Scx+h*Tf*delx*Lz;
            Tw1 = zeros(1,y);
            Te = Tbef(1:y);
            D = aw.*Tw1+ae.*Te+Sc;
            Aint = 0;
            Bint = 0;
            A(1) = an(1)/(ap(1)- as(1)*Aint);
            B(1) = (as(1)*Bint+D(1))/(ap(1)-as(1)*Aint);
            for i = 2:1:y
                A(i) = an(i)/(ap(i)- as(i)*A(i-1));
                B(i) = (as(i)*B(i-1)+D(i))/(ap(i)- as(i)*A(i-1));
            end
            T(y+1) = 0;
            for u = y:-1:1
                T(u) = A(u)*T(u+1)+B(u);
            end
            T = T(1:y);
        elseif y < ident & ident <= xmy-y
            if rem(ident,y) == 1
                Te(ident) = Tbef(ident);
                Tw(ident) = T(ident-y);
                Sc(ident) = h*Tf*delx*Lz;
                SpT(ident) = -h*delx*Lz;
                aw(ident) = ae(ident-y);
                ae(ident) = aw(ident);
                as(ident) = 0;
                an(ident) = an(ident-y);
                ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
SpT(ident);
                D(ident) =
aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-1));
                B(ident) = (as(ident)*B(ident-1)+D(ident))/(ap(ident)-
as(ident)*A(ident-1));
            elseif rem(ident,y) > 1
                Te(ident) = Tbef(ident);
                Tw(ident) = T(ident-y);
```

```
                Sc(ident) = 0;
                SpT(ident) = 0;
                aw(ident) = ae(ident-y);
                ae(ident) = aw(ident);
                as(ident) = as(ident-y);
                an(ident) = an(ident-y);
                    ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
   SpT(ident);
                    D(ident) =
   aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                    A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-
   1));
                    B(ident) = (as(ident)*B(ident-
   1)+D(ident))/(ap(ident)- as(ident)*A(ident-1));
                elseif rem(ident,y) == 0
                    Te(ident) = Tbef(ident);
                    Tw(ident) = T(ident-y);
                    Sc(ident) = Sct;
                    SpT(ident) = SpT(ident-y);
                    aw(ident) = ae(ident-y);
                    ae(ident) = aw(ident);
                    as(ident) = as(ident-y);
                    an(ident) = an(ident-y);
                    ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
   SpT(ident);
                    D(ident) =
   aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                    A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-
   1));
                    B(ident) = (as(ident)*B(ident-
   1)+D(ident))/(ap(ident)- as(ident)*A(ident-1));
                    T(ident+1) = 0;
                    for z = ident:-1:ident-y+1
                        T(z) = A(z)*T(z+1)+B(z);
                    end
                    T = T(1:ident);
                end
            elseif (xmy-y) < ident
                if rem(ident,y) == 1
                    Te(ident) = 0;
                    Tw(ident) = T(ident-y);
                    Sc(ident) = h*Tf*delx*Lz;
                    SpT(ident) = -h*delx*Lz;
                    aw(ident) = ae(ident-y);
                    ae(ident) = 0;
                    as(ident) = 0;
                    an(ident) = an(ident-y);
                    ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
   SpT(ident);
                    D(ident) =
   aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                    A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-
   1));
                    B(ident) = (as(ident)*B(ident-
   1)+D(ident))/(ap(ident)- as(ident)*A(ident-1));
                elseif rem(ident,y) > 1
                    Te(ident) = 0;
                    Tw(ident) = T(ident-y);
                    Sc(ident) = 0;
```

```
                    SpT(ident) = 0;
                    aw(ident) = ae(ident-y);
                    ae(ident) = 0;
                    as(ident) = as(ident-y);
                    an(ident) = an(ident-y);
                    ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
SpT(ident);
                    D(ident) =
aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                    A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-
1));
                    B(ident) = (as(ident)*B(ident-
1)+D(ident))/(ap(ident)- as(ident)*A(ident-1));
                elseif rem(ident,y) == 0
                    Te(ident) = 0;
                    Tw(ident) = T(ident-y);
                    Sc(ident) = Sc(ident-y);
                    SpT(ident) = SpT(ident-y);
                    aw(ident) = ae(ident-y);
                    ae(ident) = 0;
                    as(ident) = as(ident-y);
                    an(ident) = an(ident-y);
                    ap(ident) = aw(ident)+as(ident)+an(ident)+ae(ident) -
SpT(ident);
                    D(ident) =
aw(ident)*Tw(ident)+ae(ident)*Te(ident)+Sc(ident);
                    A(ident) = an(ident)/(ap(ident)- as(ident)*A(ident-
1));
                    B(ident) = (as(ident)*B(ident-
1)+D(ident))/(ap(ident)- as(ident)*A(ident-1));
                    T(ident+1) = 0;
                    for w = ident:-1:ident-y+1
                        T(w) = A(w)*T(w+1)+B(w);
                    end
                    T = T(1:xmy);
                    Tbef(1:(xmy-y))= T((y+1):xmy);
                    count = count+1;
                    post = T;
                    error = max(abs(post - Tint));
                    Tint = post;
                end
            end
        end
    end

    gx = linspace(0,Lx,x);
    gy = linspace(0,Ly,y);
    T1 = reshape(T,y,x);

    subplot(2,1,1);, contour(gx,gy,T1);, colormap;
    title('Temperature distribution,°C'), xlabel('x,(m)'),
ylabel('y,(m)'), colorbar;
    subplot(2,1,2);,pcolor(gx,gy,T1);, shading interp
    title('Temperature distribution,°C'), xlabel('x,(m)'),
ylabel('y,(m)'), colorbar;
    grid on
    grid minor
```