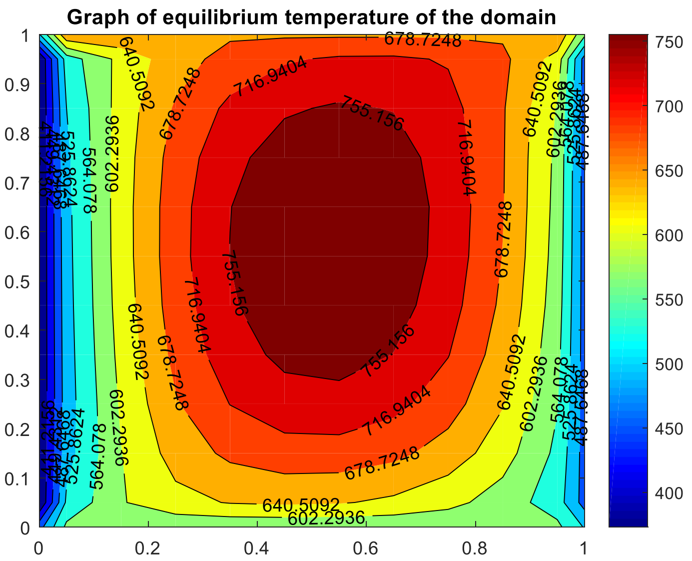
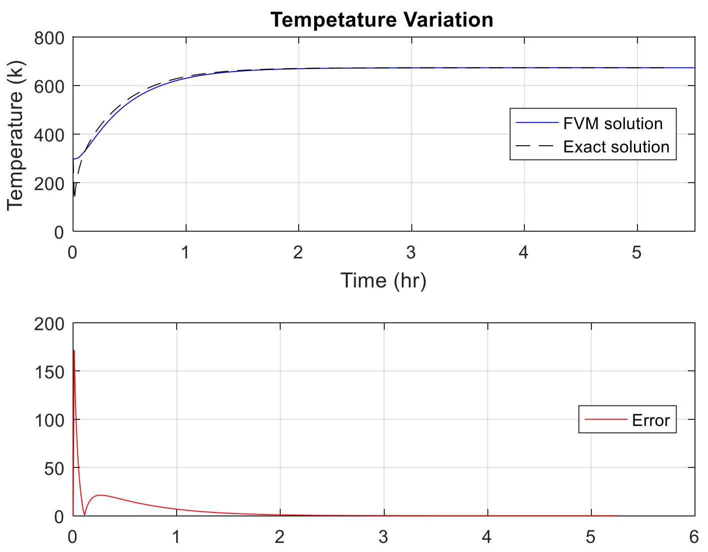
3/17/2018

ASSIGNMENT 2

COMPUTATIONAL FLUID DYNAMICS AND HEAT TRANSFER





173109003 | Sanit Prashant Bhatkar

Computational Fluid Dynamics and Heat Transfer (ME 415), Autumn 2018

Assignment 2: Transient Heat Conduction

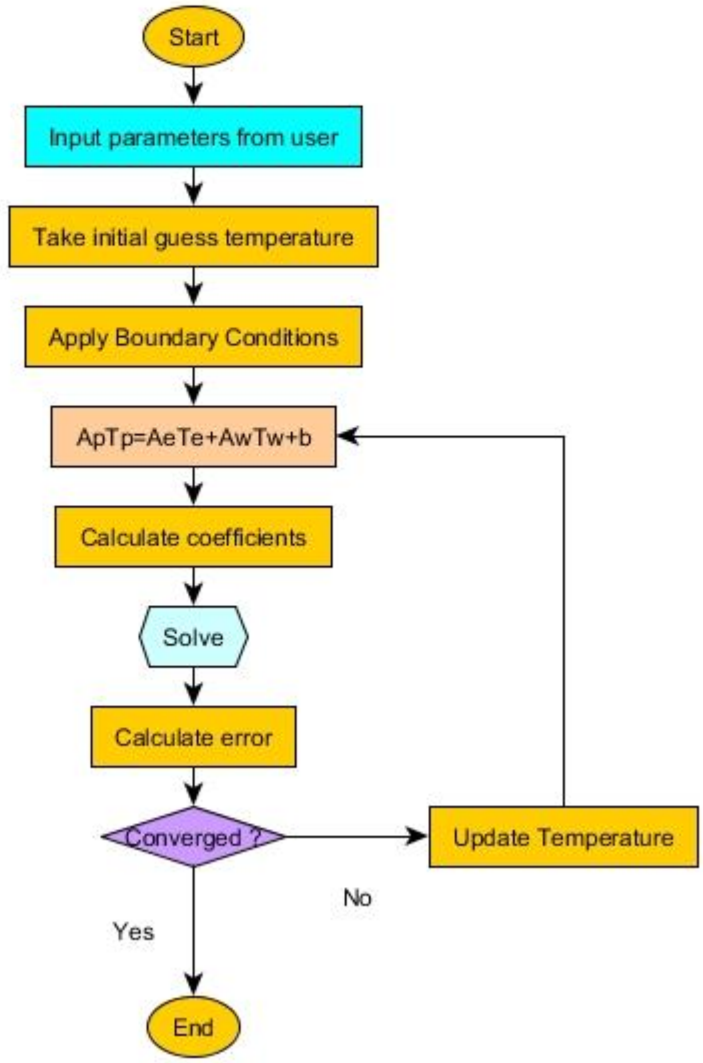
Name: Sanit Prashant Bhatkar

Roll No: 173109003

Problem 1:

One-dimensional stainless steel bar of width 2 m and a constant thermal diffusivity of 1 m2/hr. The bar is initially (t=0) at a uniform temperature of 25 ^{o}C. At t > 0, the surface temperatures of the left (x = 0) and right (x = L) faces are suddenly increased to 400 ^{o}C, and maintained at this temperature thereafter. There are no sources within the wall. Determine the time at which the middle length temperature reaches 400 °C. Plot the temperature distribution at middle length from 0 to 1 hr in steps of 0.1 hr. Plot the time-temperature graph along x direction. Solve the problem with the fully explicit method using FVM. Choose appropriate grid size taking into account of the stability criterion as discussed in class.

Algorithm



Grid details and the implemented boundary condition

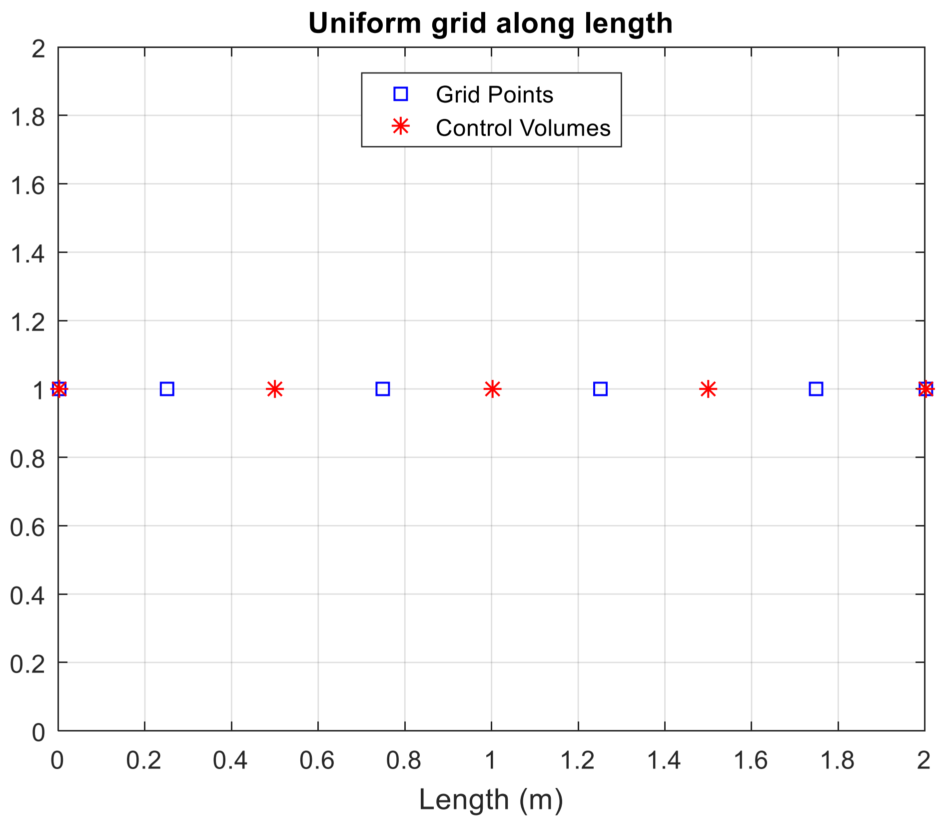


Fig 1. Generated grid for the given Cartesian geometry

Boundary Conditions

- On the left boundary uniform temperature 673K is applied
- On the right boundary uniform temperature 673K is applied

Governing Equation

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

Integrating Governing equation over the control volume

$$\int_t^{t+\Delta t} \int_w^e \frac{\partial T}{\partial t} \cdot dt \cdot dx = \alpha \int_t^{t+\Delta t} \int_w^e \frac{\partial^2 T}{\partial x^2} \cdot dt \cdot dx$$

Explicit scheme used for evaluation of the temperature.

$$(T_P^n - T_P^o) \cdot \Delta x = \alpha [\frac{(T_E^o - T_P^o)}{\delta xe} + \frac{(T_P^o - T_W^o)}{\delta xw}] \cdot \Delta t$$

Final Discretized form

$$T_P^n = \left(\frac{\alpha\Delta t}{\Delta x \cdot \delta xe}\right) T_E^o + \left(1 - \frac{\alpha\Delta t}{\Delta x \cdot \delta xe} - \frac{\alpha\Delta t}{\Delta x \cdot \delta xw}\right) T_P^o + \left(\frac{\alpha\Delta t}{\Delta x \cdot \delta xw}\right) T_W^o$$

Stability Criteria

$$\frac{\alpha\Delta t}{\Delta x}\left(\frac{1}{\delta xe} + \frac{1}{\delta xw}\right) \leq 1$$

For the first point

$$\delta xe = \Delta x$$
$$\delta xw = \Delta x/2$$

For all other points

$$\delta xe = \Delta x$$
$$\delta xw = \Delta x$$

By taking the critical value amongst the two will give us

$$\frac{\alpha\Delta t}{\Delta x^2} \leq \frac{1}{3} \qquad \text{… Eq. (1)}$$

The designed code takes care of the stability based on the Eq.1. The correct value for Δt and Δx will be displayed as output. Code will reiterate till the user gives the correct input as per the stability criteria.

Code of the problem

```
%==============================================================%
% Assignment 2 CHDHT ME 415
% Problem Number 1. 1D Transient Problem
% Progarm uses FVM to solve the problem of 1D conduction
% Heat generation is not present
% Designed only for rectangular co-ordinate system
%
% AUTHOR:
% Sanit P. Bhatkar (173109003@iitb.ac.in)
% Roll No: 173109003
% Place: IIT BOMBAY.
%==============================================================%

%% Input of Variables %%

clc
clearvars
fprintf('\n============ Unsteady Heat conduction in solid ============');
fprintf('\n\n----------------------- INPUT -----------------------\n');

%flag is used for correct stability input
flag=1;
error=1;
% l=2;
% alp=1;
% dt=0.01;
% dl=0.5;

eps=0.001;
l=input('\nLength of the domain in m: ');
alp=input('Thermal diffusivity m^2/hr: ');
dt=input('\nInput unit time step in hr: ');
dl=input('Input unit grid size in m: ');

%% Stability Criteria %%


%CFL criteria is used for stability of the scheme

fprintf('\n---------------- Stability Criteria ----------------\n');

while flag > 0

if dt/((dl)^2)<= 1/(3*alp)
    %Only when correct input is given, the loop will stop
    fprintf('\nScheme is stable for dt= %d and dx= %d\n',dt,dl);
    flag=0;
else

    fprintf('\nScheme is unstable for dt= %d and dx= %d\n',dt,dl);
    %Stable value of dt
    dtt=((dl)^2)/(3*alp);
    fprintf('\nFor stability choose dt less that dt= %d for dx= %d\n',dtt,dl);
    fprintf('or');

    %Stable value of dx
    dxx=(3*alp*dt)^0.5;
    fprintf('\nFor stability choose dx greater than dx= %d for dt= %d\n',dxx,dt);

    %flag is used for correct stability input
    dt=input('\nInput unit time step: ');
```

```
    dl=input('Input unit grid size: ');

end

end

%% Grid Formation %%

%============================FINITE VOLUME METHOD==========================

%STEP 1: Divide domain into finite sized subdomain called control volumes

%Our domain is uniform
%But boundary and the first point grid space is different
%That is why separate calculation is done
%sx represents the number of points in a grid in x direction
%NOTE: dl has to be some multiple of length

sx=(l/dl)+2;
x(1,1)=0;
x(sx,1)=l;
x(2,1)=x(1,1)+(dl/2);
for m=3:sx-1
    x(m,1)=x(m-1,1)+dl;
end


%STEP 2: Integrate governing equation over boundaries of control volume

%STEP 3: Profile assumption

%Piecewise linear profile is assumed for integral
%We will get equation of type
%ApnTpn=ApoTpo+AwoTwo+AeoTeo
%Assuming isotropic material and uniform control volume


%% Boundary Conditions %%

% tbw=400+273;
% tbe=400+273;
tbe=input('\nInput west boundary condition in degree C: ');
tbw=input('Input west boundary condition in degree C: ');
tbe=tbe+273;
tbw=tbw+273;

%Initial Condition
% Tini=25+273;
Tini=input('Input initial condition in degree C: ');
Tini=Tini+273;

%% Actual Solution %%

%st represents the time step number

st=1;
T(st,1:sx)=Tini;
T_steady(st,1:sx)=tbw;

while error>eps
%As the final time is not known, loop is run till steady state is reached
```

```
st=st+1;
T(st,1)=tbw;
T(st,sx)=tbe;

for j=2:sx-1

    dxe=x(j+1)-x(j);
    dxw=x(j)-x(j-1);
    r=(alp*dt)/dl;

    ae=r/dxe;
    aw=r/dxw;
    ap=1-ae-aw;

    %Worked out formula
    T(st,j)=aw*T(st-1,j-1)+ap*T(st-1,j)+ae*T(st-1,j+1);


end

 %Absolute error definition is used
 %for checking the steady state condition
 error=abs(T(st,round(sx/2))-tbw);

end

%% Analytic Solution

te=1;
error=1;
Te(1,1:sx)=Tini;

while error>eps

%te represents the time step number
te=te+1;
Te(te,1)=tbw;
Te(te,sx)=tbe;

for j=2:sx-1

sum=0;

%Analytic solution
for m=1:1000
    p=exp(-((m*pi/l)^2)*alp*te*dt);
    p=p*((1-(-1)^m)/(m*pi));
    p=p*sin(pi*x(j)*m/l);
    sum=sum+p;
    Te(te,j)=tbw+2*(Tini-tbw)*sum;
end


end

 %Absolute error definition is used
 %for checking the steady state condition
 error=abs(Te(te,round(sx/2))-tbw);


end
```

```
%% Output

%Kelvin to degree C converion
T=T-273;
Te=Te-273;
%============================== OUTPUT =====================================

fprintf('\n\n---------------------- OUTPUT -----------------------\n\n');

end_time=(st-1)*dt;
fprintf('Time required for attending steady state by FVM is %dhr',end_time);

end_exact=(te-1)*dt;
fprintf('\nTime required for attending steady state by exact solution is
%dhr',end_exact);


figure (1)
%Grid point plotting
%Blue squares represent the actual points
y=ones(1,sx);
plot(x,y,'bs');
title('Uniform grid along length')
grid on
hold on

%Control Volume plotting
yy=ones(1,sx-1);
xx(1,1)=x(1,1);
xx(sx-1,1)=x(sx,1);
for i=2:sx-2
    xx(i)=(x(i+1)+x(i))/2;
end
plot(xx,yy,'r*');
legend('Grid Points','Control Volumes','Location','North');
xlabel('Length (m)');
hold off


figure (2)

subplot(2,1,1)
%FVM solution plot
t=0:dt:end_time;
[n n]=size(t);

plot(t,T(1:n,round(sx/2)),'b')
title('Tempetature Variation in degree C')
hold on

%Exact solution plot
tt=0:dt:end_exact;
[n n]=size(tt);
plot(tt,Te(1:n,round(sx/2)),'--k')

hold on
grid on
xlabel('Time (hr)');
ylabel('Temperature in degree C');
xlim([0,end_time])
hold off
legend('FVM solution','Exact solution','Location','East');
```

```
subplot(2,1,2)
%Error calculation
dummy=T(1:n,1:sx);
error=abs(Te-dummy);
plot(tt,error(1:n,round(sx/2)),'r')
legend('Error','Location','East');
grid on


[g k]=size(T);
figure(3)
for i=1:(1/(10*dt)):g
    plot(x,T(i,1:k));
    hold on
end
hold off
grid on
title(' Time-Temperature graph along x-direction');
xlabel(' Length m ');
ylabel(' Temperature in degree C ');




%_______________________________________________________________________________
```

Results and Discussion

A. Time required to reach steady state

============ Unsteady Heat conduction in solid ============

---------------------- INPUT -----------------------

Length of the domain in m: 2

Thermal diffusivity m^2/hr: 1

Input unit time step in hr: 0.1

Input unit grid size in m: 0.5

---------------- Stability Criteria ----------------

Scheme is unstable for dt= 1.000000e-01 and dx= 5.000000e-01

For stability choose dt less that dt= 8.333333e-02 for dx= 5.000000e-01

or

For stability choose dx greater than dx= 5.477226e-01 for dt= 1.000000e-01

Input unit time step: 0.01

Input unit grid size: 0.5

Scheme is stable for dt= 1.000000e-02 and dx= 5.000000e-01

Input west boundary condition in degree C: 400

Input west boundary condition in degree C: 400

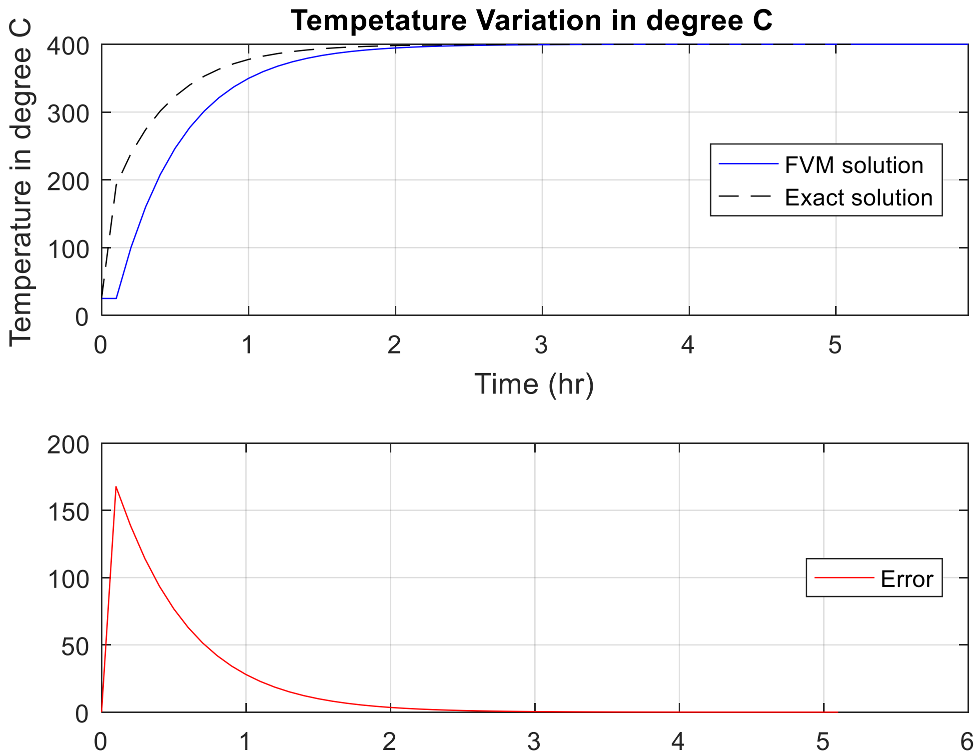
Input initial condition in degree C: 25

---------------------- OUTPUT -----------------------

Time required for attending steady state by FVM is 5.510000e+00hr

Time required for attending steady state by exact solution is 5.260000e+00hr

B. Temperature plot at middle length with Δt = 0.1hr



============ Unsteady Heat conduction in solid ============

---------------------- INPUT -----------------------

Length of the domain in m: 2

Thermal diffusivity m^2/hr: 1

Input unit time step in hr: 0.1

Input unit grid size in m: 1

---------------- Stability Criteria ----------------

Scheme is stable for dt= 1.000000e-01 and dx= 1

Input west boundary condition in degree C: 400

Input west boundary condition in degree C: 400

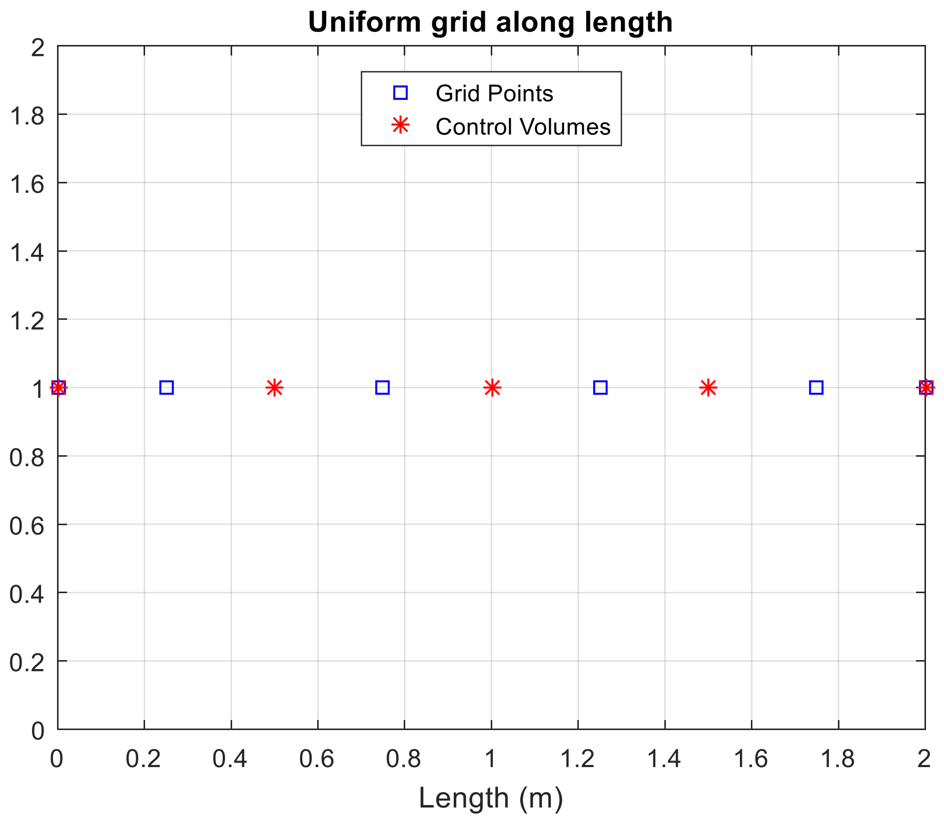
Input initial condition in degree C: 25

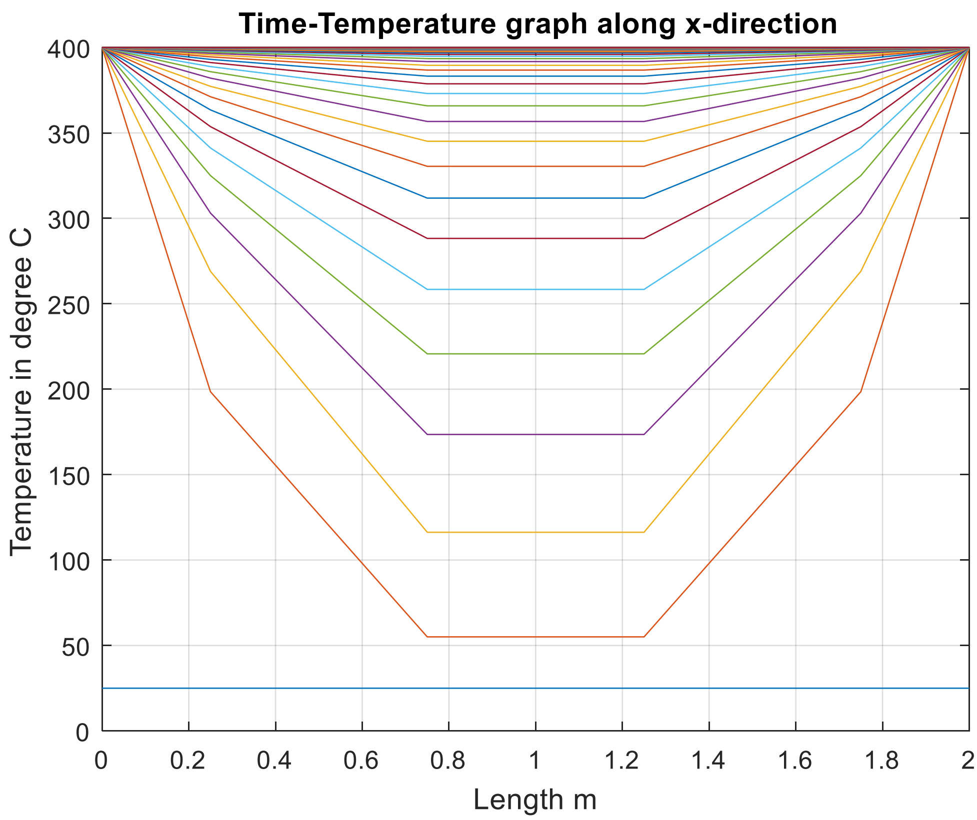
---------------------- OUTPUT ----------------------

Time required for attending steady state by FVM is 5.900000e+00hr

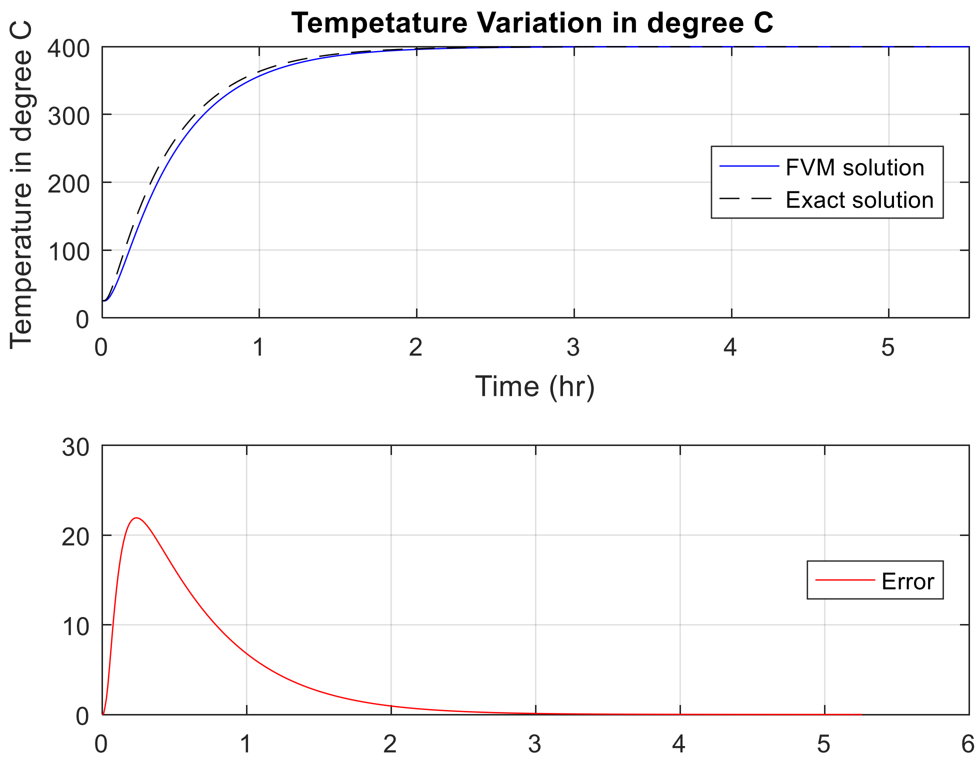
Time required for attending steady state by exact solution is 5.100000e+00hr

C. Time-Temperature graph along x direction





Time-Temperature graph along x-direction
Temperature in degree C
400
350
300
250
200
150
100
50
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Length m



Tempetature Variation in degree C
Temperature in degree C
400
300
200
100
0
0
1
2
3
4
5
Time (hr)
FVM solution
Exact solution
30
20
10
0
0
1
2
3
4
5
6
Error

============ Unsteady Heat conduction in solid ============

---------------------- INPUT -----------------------

Length of the domain in m: 2

Thermal diffusivity m^2/hr: 1

Input unit time step in hr: 0.01

Input unit grid size in m: 0.5

---------------- Stability Criteria ----------------

Scheme is stable for dt= 1.000000e-02 and dx= 5.000000e-01

Input west boundary condition in degree C: 400

Input west boundary condition in degree C: 400

Input initial condition in degree C: 25

---------------------- OUTPUT -----------------------

Time required for attending steady state by FVM is 5.510000e+00hr

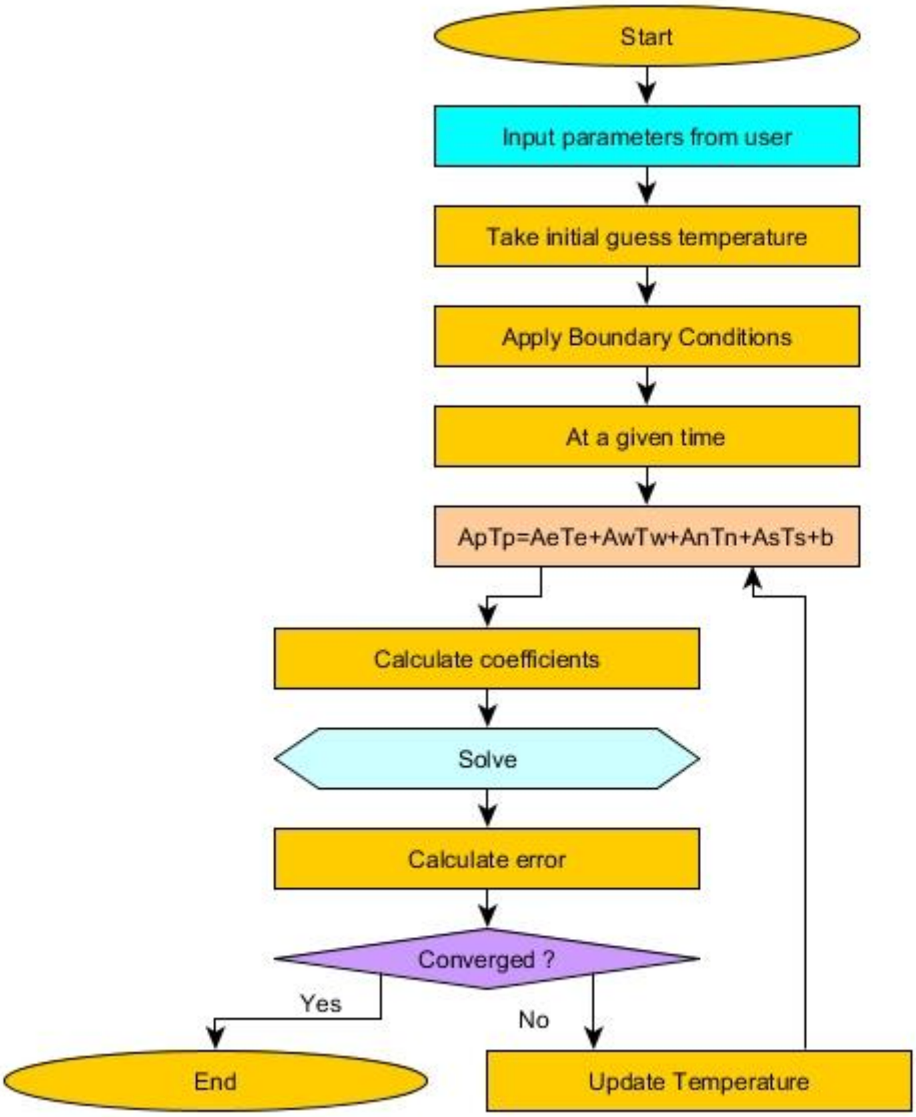
Time required for attending steady state by exact solution is 5.260000e+00hr

----

Problem 2:

2-D Computational Heat Conduction (CHC) on a uniform grid, with implicit method Consider 2D conduction in a square shaped (L1=1m and L2=1m) long stainless-steel (density ρ: 7750 kg/m3, specific-heat Cp: 500 J/Kg K, thermal-conductivity k: 16.2 W/m-K) plate. The plate is initially at a uniform temperature of 300 ^{o}C and is suddenly subjected to a constant temperature of Twb = 100^{o}C on the west boundary, Tsb = 200^{o}C on the south boundary, Teb = 300^{o}C on the east boundary, and Tnb = 400^{o}C on north boundary. Present a CFD application of the code for a volumetric heat generation of 0 and 50 kW/m^3. Consider maximum number of grid points as imax × jmax=12×12 and the steady state convergence tolerance as $\varepsilon st=10^{-4}$. Plot the steady state temperature profiles with and without volumetric heat generation.

Algorithm



Grid details and the implemented boundary condition

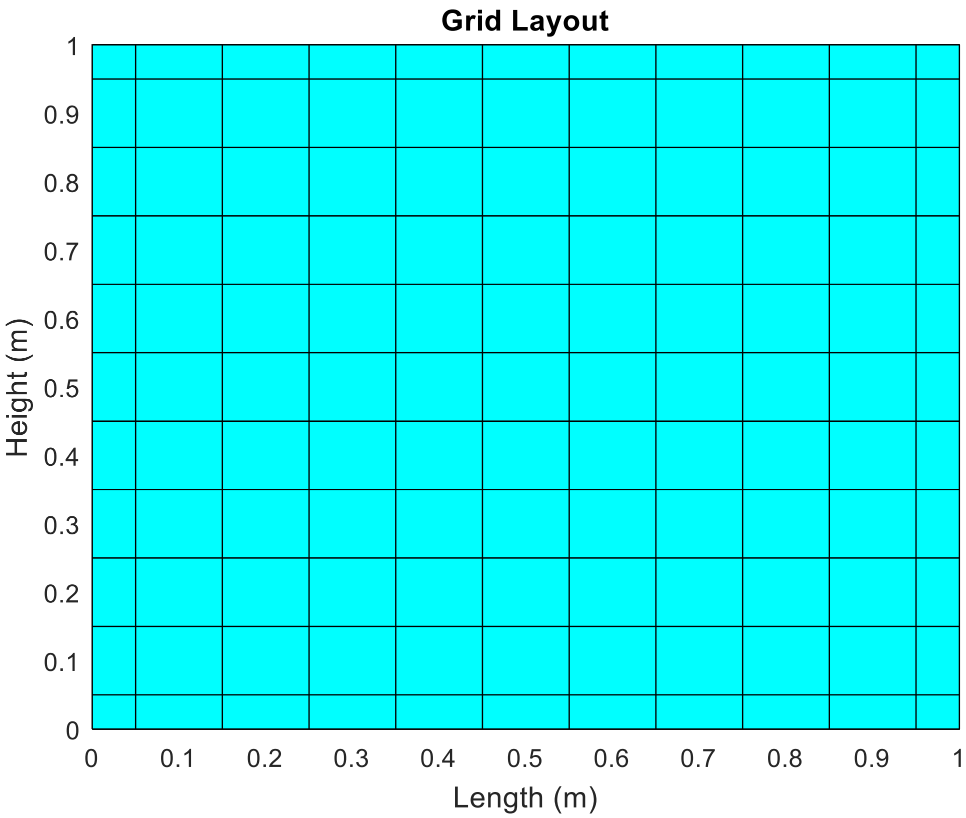


Fig 2. Generated uniform grid for given Cartesian geometry

Grid Specifications

A uniform grid was generated. Black lines are passing through the actual points on the domain. Control volume surfaces lie at half the unit distance from them.

Boundary Conditions

- On the left boundary uniform temperature 100^oC is applied
- On the right boundary uniform temperature 300^oC is applied
- On the top boundary uniform temperature 400^oC is applied
- On the bottom boundary uniform temperature 200^oC is applied

Governing Equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + q_g^{'''} = \frac{1}{\alpha}\frac{\partial T}{\partial t}$$

Integrating Governing equation over the control volume

$$\int_t^{t+\Delta t}\int_s^n\int_w^e \frac{1}{\alpha}\frac{\partial T}{\partial t}dxdydt = \int_t^{t+\Delta t}\int_s^n\int_w^e \frac{\partial^2 T}{\partial x^2}dxdydt + \int_t^{t+\Delta t}\int_s^n\int_w^e \frac{\partial^2 T}{\partial x^2}dxdy\,dt$$

Implicit scheme used for evaluation of the temperature.

$$(T_P^n - T_P^o)\cdot\frac{\Delta x\Delta y}{\alpha\Delta t} = \left(\left[\frac{(T_E^n - T_P^n)}{\delta xe} + \frac{(T_P^n - T_W^n)}{\delta xw}\right]\Delta y + \left[\frac{(T_N^n - T_P^n)}{\delta yn} + \frac{(T_P^n - T_S^n)}{\delta ys}\right]\Delta x + q_g^{'''}\Delta x\Delta y\right)$$

Final Discretized form

$$a_P^n T_P^n = a_E T_E^n + a_W T_W^n + a_N T_N^n + a_S T_S^n + a_p^o T_P^o + b$$

$$a_E = \frac{\Delta y}{\delta xe}; \quad a_W = \frac{\Delta y}{\delta xw}; \quad a_N = \frac{\Delta x}{\delta yn}; \quad a_S = \frac{\Delta x}{\delta ys};$$

$$a_P^o = \frac{\Delta x\Delta y}{\alpha\Delta t}; \quad b = \frac{q_g^{'''}}{k}\Delta x\Delta y$$

$$a_P^n = \frac{\Delta x\Delta y}{\alpha\Delta t} + \frac{\Delta y}{\delta xe} + \frac{\Delta y}{\delta xw} + \frac{\Delta x}{\delta yn} + \frac{\Delta x}{\delta ys}$$

Code of the problem

```
%==============================================================%
% Assignment 2 CHDHT ME 415
% Problem Number 2. 2D Transient Problem
% Progarm uses FVM to solve the problem of 2D conduction
% Heat generation is present
% Designed only for rectangular co-ordinate system
%
% AUTHOR:
% Sanit P. Bhatkar (173109003@iitb.ac.in)
% Roll No: 173109003
% Place: IIT BOMBAY.
%==============================================================%

%% Input of variables %%

clc
clearvars
fprintf('\n============ Unsteady Heat conduction in solid ============');
fprintf('\n\n----------------------- INPUT ------------------------\n');



% l=1;
% h=1;
% k=16.2;
% rho=7750;
% Cp=500;
% dt=10;
% sx=12;
% sy=12;
% eps=10^-4;
% qg=5e4;

%------------Input from user----------------

fprintf('\nMaterial and domain properties -->\n');
l=input('\nLength of domain (m): ');
h=input('Height of domain (m): ');
k=input('Conductivity of Material W/m-k: ');
rho=input('Density of material kg/m3: ');
Cp=input('Specific heat of material J/kg-k: ');
dt=input('\nTime step: ');
sx=input('Number of points along length: ');
sy=input('Number of points along height: ');
eps=input('\nInput Convergence error: ');
qg=input('\nHeat generation rate W/m3: ');
alp=k/(rho*Cp);

%% Boundary Conditions %%

%tbn = temp on north boundary
%tbs = temp on south boundary
%tbe = temp on east boundary
%tbw = temp on west boundary

% tbn=400+273;
% tbs=200+273;
% tbe=300+273;
% tbw=100+273;

fprintf('\nBoundary Conditions -->\n');
tbn=input('\nBoundary condition at north boundary in degree C:  ');
```

```
tbs=input('Boundary condition at south boundary in degree C:  ');
tbe=input('Boundary condition at east boundary in degree C:  ');
tbw=input('Boundary condition at west boundary in degree C:  ');

tbn=tbn+273;
tbs=tbs+273;
tbe=tbe+273;
tbw=tbw+273;

%% Grid Formation %%

%============================FINITE VOLUME METHOD==========================

%STEP 1: Divide domain into finite sized subdomain called control volumes

%Our domain is uniform
%But boundary and the first point grid space is different
%That is why separate calculation is done
%sx represents the number of points in a grid in x direction
%NOTE: dl has to be some multiple of length

dl=1/(sx-2);
x(1,1)=0;
x(sx,1)=l;
x(2,1)=x(1,1)+(dl/2);
for m=3:sx-1
    x(m,1)=x(m-1,1)+dl;
end

%sy represents the number of points in a grid in y direction

dh=h/(sy-2);
y(1,1)=0;
y(sy,1)=h;
y(2,1)=y(1,1)+(dh/2);

for m=3:sy-1
    y(m,1)=y(m-1,1)+dh;
end


%STEP 2: Integrate governing equation over boundaries of control volume

%STEP 3: Profile assumption

%Piecewise linear profile is assumed for integral
%We will get equation of type
%ApTp=AeTe+AwTw+AnTn+AsTs+B
%Assuming isotropic material and uniform control volume with no heat generation
%We get 4*Tp=Te+Tw+Tn+Ts

%% Method of ADI%%


%Formulating solution of type [A][Tp]=[B]

%Defingin matrix for temperature
%T represents matrix for visualization of grids.


T=zeros(sy,sx);
T(1:sy,1:sx)=30+273;
```

```
% T(1:sy,1:sx)= input('\nInitial guess of temperature for the domain: ');
T_ini=T(1,1);

%Application of boundary conditions
T(1,1:sx)=tbn;
T(sy,1:sx)=tbs;
T(1:sy,sx)=tbe;
T(1:sy,1)=tbw;

%At corners discotinuity is present
%To take care of that, average has been taken at the corner
T(1,1)=(tbn+tbw)/2;
T(sy,1)=(tbs+tbw)/2;
T(1,sx)=(tbn+tbe)/2;
T(sy,sx)=(tbe+tbs)/2;

sw_error=1;



xsweep=T;


while sw_error>eps



%% Sweeping along x-direction  %%

u(1,1)=tbn;
u(sy,1)=tbs;


for j=2:sx-1

%Defining problem for TDMA
%Forward Elimination

for i=2:sy-1

%dxe is distance from point P to east point
%dxw is distance from point P to west point
%dyn is distance from point P to north point
%dys is distance from point P to south point

dxe=x(j+1,1)-x(j,1);
dxw=x(j,1)-x(j-1,1);
dyn=y(i,1)-y(i-1,1);
dys=y(i+1,1)-y(i,1);

%apTp=aeTe+awTw+anTn+asTs

ae=k*dh/dxe;
aw=k*dh/dxw;
an=k*dl/dyn;
as=k*dl/dys;
apo=rho*Cp*dl*dh/(dt/2);
ap=ae+aw+an+as+apo;
S=(qg)*(dl*dh);

%a represents digonal element
%b and c are upper and lower digonals respectively
```

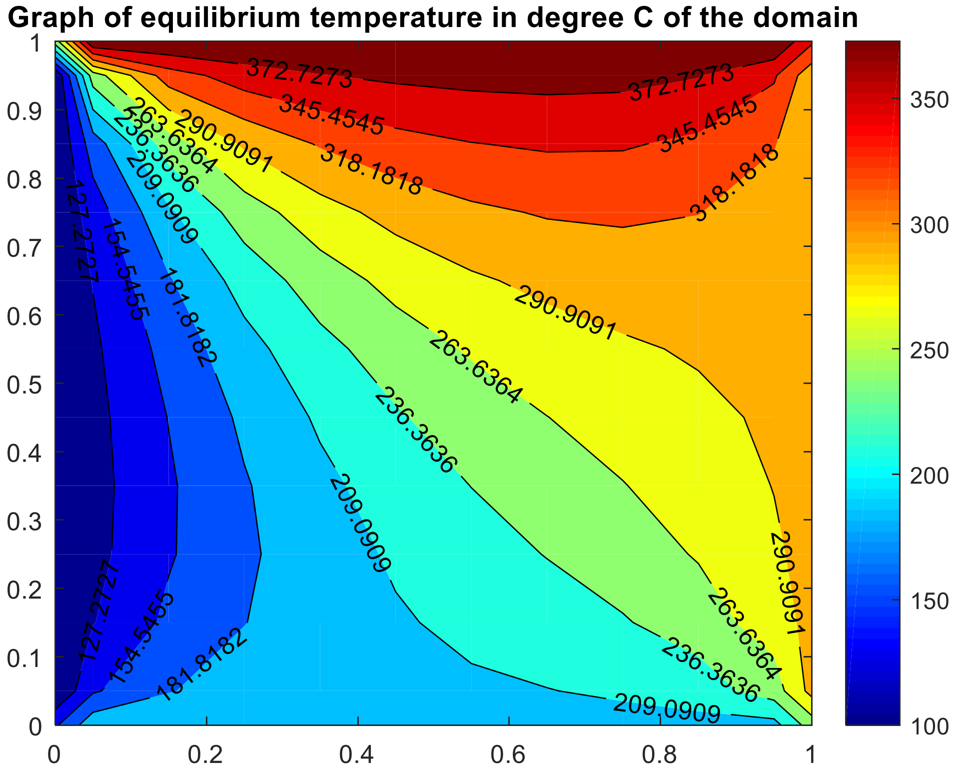
```
a=ap;

if i==2

    b=as;
    c=0;
    d=ae*T(i,j+1)+aw*T(i,j-1)+S+an*tbn+apo*T(i,j);
    %Forward Elimination
    P(1,1)=b/a;
    Q(1,1)=d/a;



elseif i==sy-1

    b=0;
    c=an;
    d=ae*T(i,j+1)+aw*T(i,j-1)+S+as*tbs+apo*T(i,j);
    %Forward Elimination
    P(i-1,1)=b/(a-c*P(i-2,1));
    Q(i-1,1)=(d+c*Q(i-2,1))/(a-c*P(i-2,1));



else

    b=as;
    c=an;
    d=ae*T(i,j+1)+aw*T(i,j-1)+S+apo*T(i,j);
    %Forward Elimination
    P(i-1,1)=b/(a-c*P(i-2,1));
    Q(i-1,1)=(d+c*Q(i-2,1))/(a-c*P(i-2,1));



end

end

%Back Substitution
u(sy-1,1)=Q(sy-2,1);
for v=sy-2:-1:2
    u(v,1)=P(v-1,1)*u(v+1,1)+Q(v-1,1);
end

xsweep(1:sy,j)=u';


end


ysweep=xsweep;

%% Sweeping along y-direction
u(1,1)=tbw;
u(sx,1)=tbe;


for i=2:sy-1
```

```
%Defining problem for TDMA
%Forward Elimination

for j=2:sx-1

%dxe is distance from point P to east point
%dxw is distance from point P to west point
%dyn is distance from point P to north point
%dys is distance from point P to south point

dxe=x(j+1,1)-x(j,1);
dxw=x(j,1)-x(j-1,1);
dyn=y(i,1)-y(i-1,1);
dys=y(i+1,1)-y(i,1);


%apTp=aeTe+awTw+anTn+asTs

ae=k*dh/dxe;
aw=k*dh/dxw;
an=k*dl/dyn;
as=k*dl/dys;
apo=rho*Cp*dl*dh/(dt/2);
ap=ae+aw+an+as+apo;
S=(qg)*(dl*dh);

%a represents digonal element
%b and c are upper and lower digonals respectively

a=ap;

if j==2

    b=ae;
    c=0;
    d=an*T(i-1,j)+as*T(i+1,j)+S+aw*tbw+apo*T(i,j);
    %Forward Elimination
    P(1,1)=b/a;
    Q(1,1)=d/a;




elseif j==sy-1

    b=0;
    c=aw;
    d=an*T(i-1,j)+as*T(i+1,j)+S+ae*tbe+apo*T(i,j);
    %Forward Elimination
    P(j-1,1)=b/(a-c*P(j-2,1));
    Q(j-1,1)=(d+c*Q(j-2,1))/(a-c*P(j-2,1));




else

    b=ae;
    c=aw;
    d=an*T(i-1,j)+as*T(i+1,j)+S+apo*T(i,j);
    %Forward Elimination
    P(j-1,1)=b/(a-c*P(j-2,1));
    Q(j-1,1)=(d+c*Q(j-2,1))/(a-c*P(j-2,1));
```

```
end

end

%Back Substitution
u(sx-1,1)=Q(sx-2,1);
for v=sy-2:-1:2
    u(v,1)=P(v-1,1)*u(v+1,1)+Q(v-1,1);
end

ysweep(i,1:sx)=u';


end

sw_error=abs(ysweep-T);
sw_error=max(max(sw_error));
T=ysweep;

end


%% Output %%

%For conversion from kelvin to degree C
T=T-273;
figure (1)
r=ones(sy,sx);
map = [0,1,1];
colormap(map)
pcolor(x,y,r)
title('Grid Layout');
xlabel('Length (m)');
ylabel('Height (m)');
%Plot of the temperature contours as the output.

figure(2)
colormap jet
contourf(x,y,flipud(T),10,'showtext','on')
colorbar
title('Graph of equilibrium temperature in degree C of the domain')


%_______________________________________________________________________________
```

Results and Discussion

A. Steady state temperature profile without heat generation



============ Unsteady Heat conduction in solid ============

--------------------- INPUT ----------------------

Material and domain properties -->

Length of domain (m): 1
Height of domain (m): 1
Conductivity of Material W/m-k: 16.2
Density of material kg/m3: 7750
Specific heat of material J/kg-k: 500

Time step: 1000
Number of points along length: 12
Number of points along height: 12

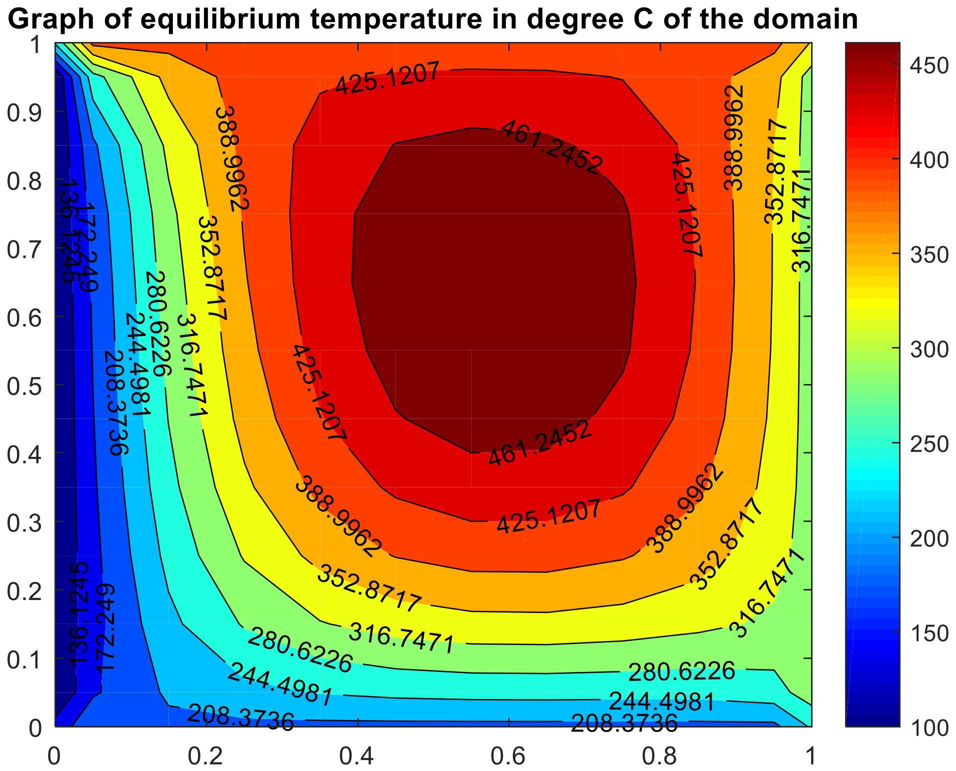
Input Convergence error: 1e-4

Heat generation rate W/m3: 0

Boundary Conditions -->

Boundary condition at north boundary in degree C: 400
Boundary condition at south boundary in degree C: 200
Boundary condition at east boundary in degree C: 300
Boundary condition at west boundary in degree C: 100

B. Steady state temperature profile without heat generation



============ Unsteady Heat conduction in solid ============

---------------------- INPUT -----------------------

Material and domain properties -->

Length of domain (m): 1
Height of domain (m): 1
Conductivity of Material W/m-k: 16.2
Density of material kg/m3: 7750
Specific heat of material J/kg-k: 500

Time step: 1000
Number of points along length: 12
Number of points along height: 12

Input Convergence error: 1e-4

Heat generation rate W/m3: 5e4

Boundary Conditions -->

Boundary condition at north boundary in degree C: 400
Boundary condition at south boundary in degree C: 200
Boundary condition at east boundary in degree C: 300
Boundary condition at west boundary in degree C: 100