توضیحات معادله:

یک فین ربع نامتناهی داریم که معادله آن به شکل زیر است.

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=\frac{1}{\alpha}\frac{\partial T}{\partial t} \quad (x>0, y>0, t>0) \tag{3a}$$

و شرایط اولیه و مرزی آن به شکل زیر است.

$$T(x=0, y, t)=0 \quad (y>0, t>0) \tag{3b}$$

$$-k\left(\frac{\partial T}{\partial y}\right)_{y=0}+hT(x, y=0, t)=f_0 \quad (x>0, t>0) \tag{3c}$$

$$T(x, y, t=0)=0 \quad (x>0, y>0) \tag{3d}$$

از آنجا که فین ربع نامتناهی است میدانیم از دو سمت دیگر (در xوy بینهایت) دمای آن ها به دمای محیط یا Tinf میرسد، لذا برای مابقی شرایط مرزی که به صورت دقیق ذکر نشده از دمای 298 استفاده میکنیم.

برای حل صریح معادله را به صورت forward باز میکنیم و حل میکنیم.

```
clear
close all
clc

xl=1; %xlast in meter
x0=0; %x0 in meter
yl=1; %ylast in meter
y0=0; %y0 in meter
Tl=1000; %Tlast in secs
dt=1; %time step
dx=0.1; %x step
dy=0.1; %y step
a=2.3e-5; %alpha in m^2/s
k=200; %(W/m k)
h=100; %W/(m2 K)
tinf=400; %T(inf)
Ax=a*dt/dx^2; %lambda for x
Ay=a*dt/dy^2; %lambda for y
lx=(xl-x0)/dx+1; %length(x) with discretization
ly=(yl-y0)/dy+1; %length(y) with discretization
lt=Tl/dt+1; %length(t) with discretization
t=ones(lx,ly,lt)*273; %prealocating
f0=h*tinf; %f0 is a factor that can be considered as h*tinf
for n=1:lt-1
    for j=2:ly-1
        for i=2:lx-1
            %Boundry and initial conditions
```

```
            t(i,2,n)=(f0*dy/k)-(h*dy/k*t(i,1,n))+(t(i,1,n));
            t(:,:,1)=273;
            t(:,end,:)=298;
            t(1,:,:)=273;
            t(end,:,:)=298;
            % calculation
            t(i,j,n+1)=Ax*(t(i+1,j,n)-2*t(i,j,n)+t(i-1,j,n))+Ay*(t(i,j+1,n)-
2*t(i,j,n)+t(i,j-1,n))+t(i,j,n);
        end
    end
end
%extending x,y and time for surfing
x=x0:dx:xl;
y=y0:dy:yl;
T=0:dt:Tl;
[x,y]=meshgrid(x,y);
%surfing for the last time node(t=Tl)
surf(x,y,t(:,:,1001))
```

برای حل ضمنی و کرنک نیکلسون باید معادلات ضمنی و مقادیر مرزی را به صورت همزمان حل کنیم. با انجام دادن آن از طریق برنامه ی زیر:

```
clear
close all
clc

xl=1; %xlast in meter
x0=0; %x0 in meter
yl=1; %ylast in meter
y0=0; %y0 in meter
Tl=1000; %Tlast in secs
dt=1; %time step
dx=0.1; %x step
dy=0.1; %y step
a=2.3e-5; %alpha in m^2/s
k=200; %(W/m k)
h=100; %W/(m2 K)
tinf=400; %T(inf)
lambda=a*dt/dx^2; %lambda
lx=(xl-x0)/dx+1; %length(x) with discretization
ly=(yl-y0)/dy+1; %length(y) with discretization
lt=Tl/dt+1; %length(t) with discretization
t=ones(lx,ly,lt)*273; %prealocating
f0=h*tinf; %f0 is a factor that can be considered as h*tinf

for n=1:lt-1
    A=zeros(lx^2); %prealocating A
    id=reshape(1:(lx^2),[lx ly]); %creating an id matrix for determing
constant positions in matrix
    [s1,s2]=size(id);
    ii=0; %equation counter
    B=zeros(1,lx^2); %prealcating B
    for j=2:ly-1
        for i=2:lx-1
            ii=ii+1;
            A(ii,id(i+1,j))=-lambda*0.5;
```

```
            A(ii,id(i,j+1))=-lambda*0.5;
            A(ii,id(i-1,j))=-lambda*0.5;
            A(ii,id(i,j-1))=-lambda*0.5;
            A(ii,id(i,j))=2*lambda+1;
            t(i,2,n)=(f0*dy/k)-(h*dy/k*t(i,1,n))+(t(i,1,n));
            t(:,:,1)=273;
            t(:,end,:)=298;
            t(1,:,:)=273;
            t(end,:,:)=298;
            B(ii)=t(i,j,n)+0.5*lambda*(t(i+1,j,n)+t(i,j+1,n)+t(i-
1,j,n)+t(i,j-1,n)-4*t(i,j,n));
        end
    end
    %% Alocating initial and boundry conditions to A and B
    T=t(:,:,n);
    idd=id([1,end],:);idd=idd';
    id([1,end],:)=[];
    for jj=1:numel(idd)
        ii=ii+1;
        A(ii,idd(jj))=1;
        B(ii)=T(idd(jj));
    end
    idd=id(:,[1,end]);id([1,end],:)=[];
    for jj=1:numel(idd)
        ii=ii+1;
        A(ii,idd(jj))=1;
        B(ii)=T(idd(jj));
    end
    %% calculating Temprature for time(n+1)
    T=A\B';
    t(:,:,n+1)=reshape(T,[lx ly]);
end
% extending x,y and time for surfing
x=x0:dx:xl;
y=y0:dy:yl;
T=0:dt:Tl;
[x,y]=meshgrid(x,y);
%surfing T for last time node
surf(x,y,t(:,:,1001))
```

میتوان اختلاف روش ها را با کم کردن جواب نود آخر معادله ی صریح از معادله ی کرنک نیکلسون و معادله‌ی ضمنی از کرنک نیکلسون را با در نظر گرفتن خطا از معادله ی کرنک نیکلسون مقایسه کرد.

برای اینکار ابتدا از خطا قدر مطلق گرفته و با هم جمع میکنیم تا تنها یک عدد برای مقایسه داشته باشیم برای این کار از کد زیر استفاده میکنیم.

```
clear
close all
clc

load cranck.dat
load implicit.dat
load explicit.dat
```

```
Error_explicit=abs(cranck-explicit);
sError_explicit=sum(sum(Error_explicit));
Error_implicit=abs(cranck-implicit);
sError_implicit=sum(sum(Error_implicit));
Err=sError_explicit-sError_implicit;
fprintf('explicit error - implicit error= %d \n',Err)
```

حال جواب ها به دست آمده را نگاه میکنیم:

explicit error - implicit error= 5.900000e-04

از آنجا که کرنک نیکلسون جواب دقیق تری میدهد (به علت مشمول بودن هر دو معادله ی صریح و ضمنی) آن را به عنوان جواب دقیق در نظر میگیریم. حال با توجه به تفاوت خطا ها از مقدار دقیق آن ها را با هم مقایسه میکنیم. با توجه به اینکه تفریق با جواب مثبت دارد میفهمیم که خطای explicit بیشتر از implicit بوده ولی با توجه به استپ های کوچک و order خطا میدانیم که هرچند خطا کم تر میشود اما تعداد معادلات بیشتری باید حل شود و با توجه به ساده بودن حل صریح و کم خطا بودن آن میتوان از آن در این مسأله استفاده نمود.