NATIONAL INSTITUE OF TECHNOLOGY KARANATAKA, SURATHKAL

ASSIGNMENT 12

APPLIED COMPUTATIONAL METHODS IN MECHANICAL SCIENCES

RAJAT A CHANDAVAR – 16ME156

04-Nov-19

ASSIGNMENT ON STEADY CONVECTION DIFFUSION

Answer

- The problem was solved for 2 values of diffusion coefficient T = 0.001 and T = 0.01
- 3 Grid sizes were analysed – 41X41, 61X61 and 101X101
- Convergence criteria → Approximate error < 1e-6

Code(C++)

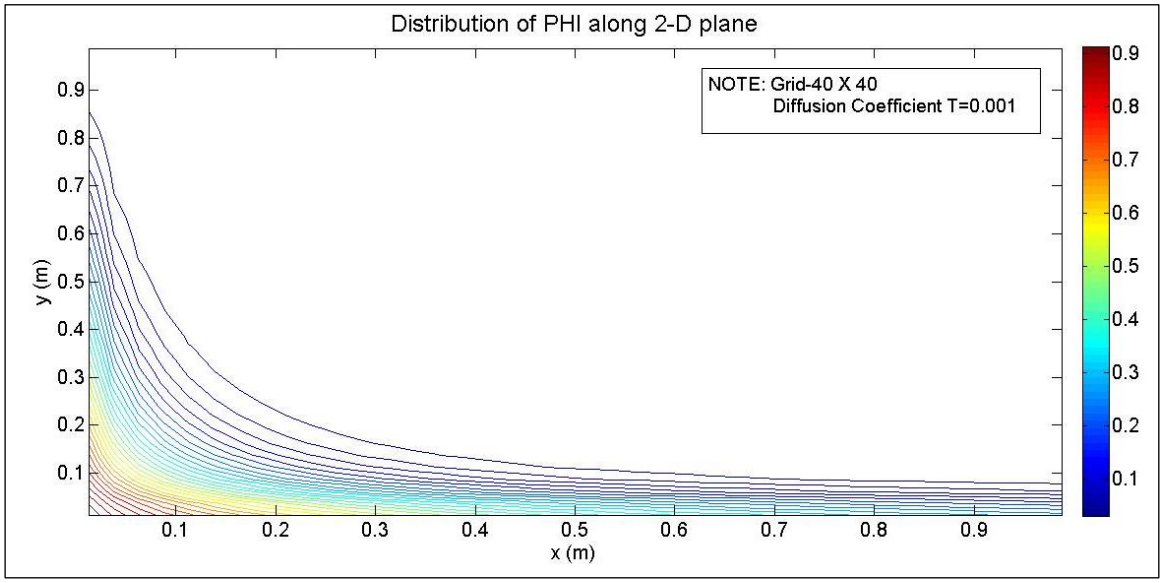
```
#include<iostream>
#include<cmath>
#include<fstream>
using namespace std;
main()
{
    fstream f,g,h;
    f.open("Phi.txt",ios::out);
    g.open("x_grid.txt",ios::out);
    h.open("y_grid.txt",ios::out);
    int i,j,nx=40,ny=40,iter=0;
    float lx=1.0,ly=1.0,T=0.001,dx=lx/nx,dy=ly/ny,r=dx/dy;
    float x[nx+1],y[ny+1],u[ny][nx+1],v[ny+1][nx],phi[ny+2][nx+2]={ };//Phi is initialized to zero
    float aw,ae,ap,an,as,q,omega=0.15,error,max_error,temp;
    //Calculation of x and y
    for(j=0;j<nx+1;++j)
        x[j]=j*dx;
    for(i=0;i<ny+1;++i)
        y[i]=i*dy;
    //Calculation of x and y velocity
    for(i=0;i<ny+1;++i)
    {
        for(j=0;j<nx+1;++j)
        {
            if(i!=ny)
                u[i][j]=x[j];
            if(j!=nx)
                v[i][j]=-y[i];
        }
    }
    //Input boundary conditions
        //Top and bottom Boundary
    for(j=1;j<nx+1;++j)
    {
        phi[ny+1][j]=-phi[ny][j];
        phi[0][j]=phi[1][j];
    }
        //Right and Left Boundary
    for(i=1;i<ny+1;++i)
    {
        phi[i][nx+1]=phi[i][nx];
        phi[i][0]=2*(1-(y[i-1]+dy/2))-phi[i][1];
    }
    //SOR
    do
    {
        ++iter;
        for(i=1;i<ny+1;++i)
        {
            for(j=1;j<nx+1;++j)
            {
                aw=-dy/2*u[i-1][j-1]-T/r;
                ae=dy/2*u[i-1][j]-T/r;
                as=-dx/2*v[i-1][j-1]-r*T;
                an=dx/2*v[i][j-1]-r*T;
                ap=(aw+ae+as+an)+4*T/r+4*r*T;
```

```
                    q=0;
                    if(i==1)//South
                    {
                        ap=ap+as;
                        as=0;
                    }
                    else if(i==ny)//North
                    {
                        ap=ap-an;
                        an=0;
                    }
                    if(j==1)//Left
                    {
                        q=-2*aw*(1-(y[i-1]+dy/2));
                        ap=ap-aw;
                        aw=0;
                    }
                    else if(j==nx)//Right
                    {
                        ap=ap+ae;
                        ae=0;
                    }
                    temp=phi[i][j];
                    phi[i][j]=(1-omega)*temp+omega*(q-aw*phi[i][j-1]-
ae*phi[i][j+1]-an*phi[i+1][j]-as*phi[i-1][j])/ap;
                    error=abs(phi[i][j]-temp);
                    if(i==1 && j==1)
                        max_error=error;
                    else if(error>max_error)
                        max_error=error;
                }
            }
            //Boundary conditions
            for(j=1;j<nx+1;++j)
            {
                phi[ny+1][j]=-phi[ny][j];
                phi[0][j]=phi[1][j];
            }
            for(i=1;i<ny+1;++i)
            {
                phi[i][nx+1]=phi[i][nx];
                phi[i][0]=2*(1-(y[i-1]+dy))-phi[i][1];
            }
            cout<<"\nIteration: "<<iter<<"    Error: "<<max_error<<"\n";
        }while(max_error>1e-6);
        //Grid written to DATA files
        for(j=0;j<nx;++j)
            g<<x[j]+dx/2<<"\n";
        for(i=ny-1;i>=0;--i)
            h<<y[i]+dy/2<<"\n";
        //PHI written to data file
        for(i=ny;i>0;--i)
        {
            for(j=1;j<nx+1;++j)
                f<<phi[i][j]<<" ";
            f<<"\n";
        }
        f.close();
        g.close();
        h.close();
    }
```

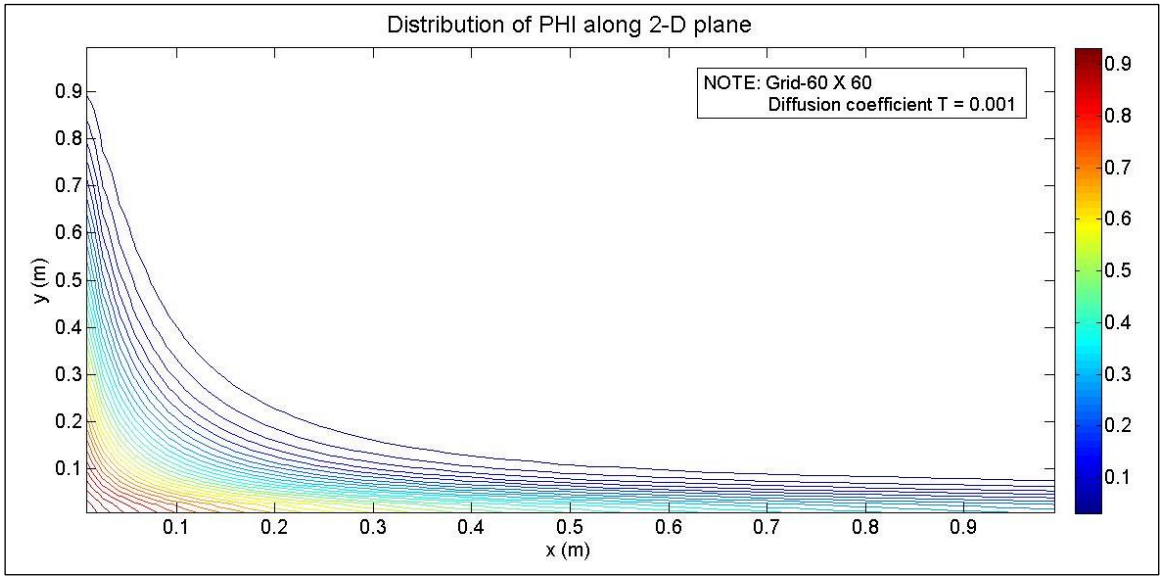
Output

For Diffusion coefficient T = 0.001

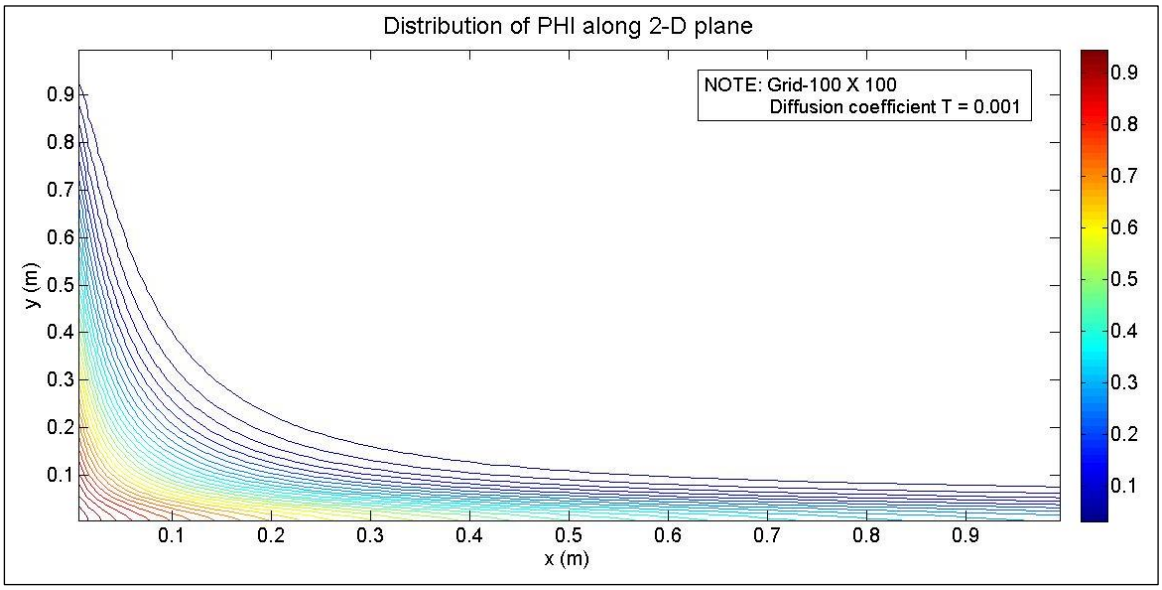
- *Grid 40X40*



- *Grid 60X60*



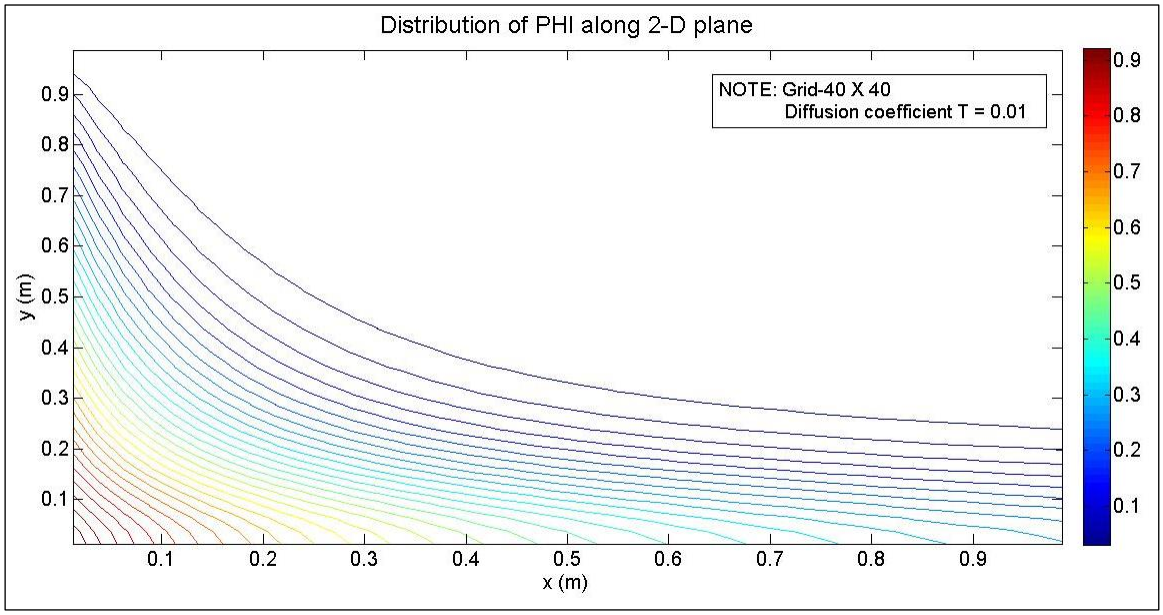
- *Grid 100X100*



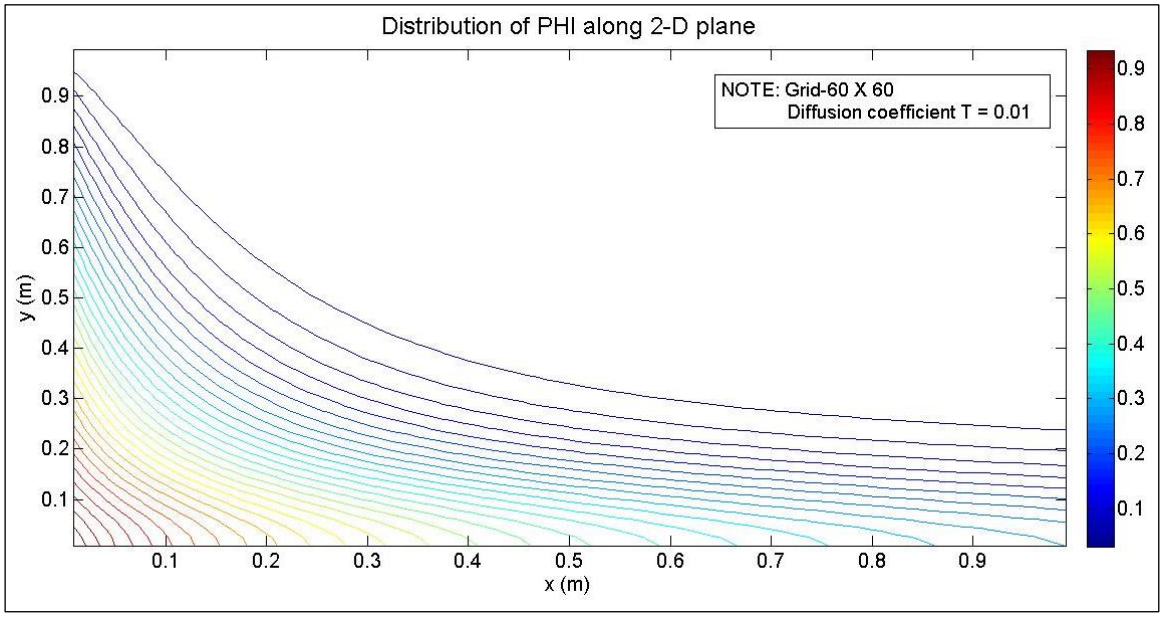
Observation: The function PHI seemed to have high values only for smaller x and y values. All grids gave approximately the same result. With increase in grid size, contour lines seemed to smoothen out for smaller x values and higher y values.

For Diffusion coefficient T = 0.01

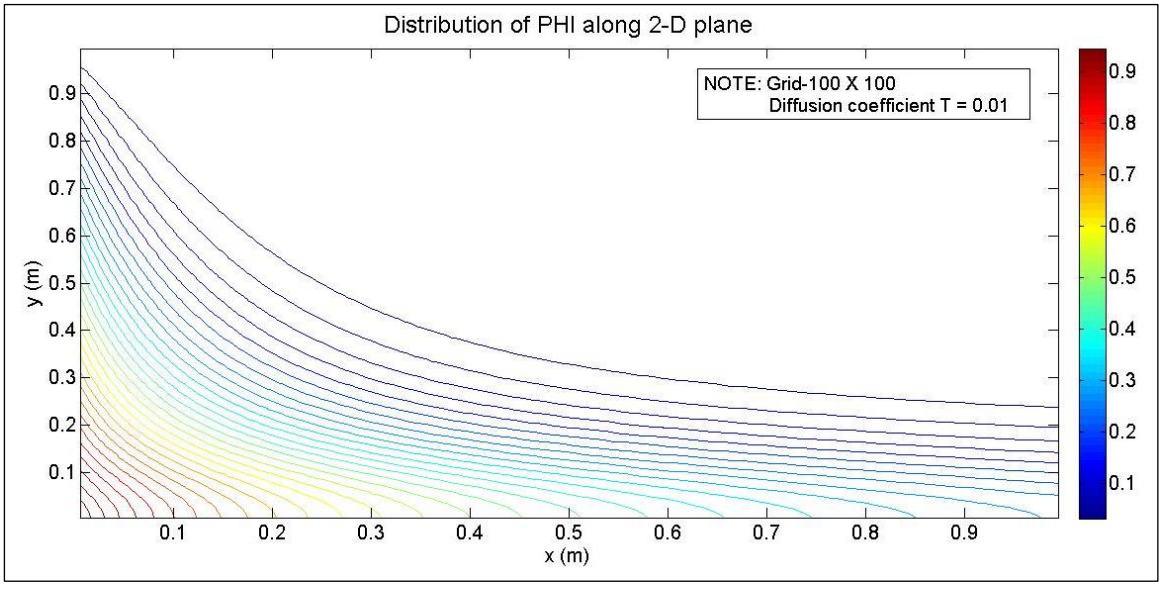
- *Grid 40X40*



- *Grid 60X60*



- *Grid 100X100*



Observation: The increased value of diffusion coefficient T led to larger distribution of PHI across 2-D surface when compared to previous T value. For a larger value of T, PHI is not attenuated easily and diffuses to a larger region in comparision to smaller value of T.