
```
%Shivam_Swarnakar_184106011_ME415
%MATLAB code for Implicit method-based solver for Multi-Solid
Conduction in a 2D plate
clc;
clear;

L =1; imax = 22;
H =1; jmax = 22;

Qgen = 0;

rho = [8000, 2700, 8900, 7900];
Cp = [ 502, 896, 385, 452];
k = [16.2, 220, 385, 72];

T0 = 30;
Tleft = 100; Tright = 300;
Tbottom = 200; Ttop = 400;

T(1:imax,1:jmax) = T0;
T(1,:) = Tleft; T(imax,:) = Tright;
T(:,1) = Tbottom; T(:,jmax) = Ttop;

Tag(1:imax,1:jmax) = zeros;
Tag(1:imax/2,1:jmax/2) = 1;
Tag(1:imax/2,jmax/2+1:jmax) = 3;
Tag(imax/2+1:imax,1:jmax/2) = 2;
Tag(imax/2+1:imax,jmax/2+1:jmax)=4;

Dx = L/(imax-2);
Dy = H/(jmax-2);

% Calculating aE and aN
for j = 2:jmax-1
    for i = 1:imax-1

        if i==2
            dx=Dx/2;
        else
            dx=Dx;
        end

        if i==imax-1
            dx=Dx/2;
        else
            dx=Dx;
        end

        if (i <imax/2 && j <jmax/2) % steel
            aE(i,j) = k(1)*Dy/dx;
        end
        if (i == imax/2 && j <jmax/2) % interface of Steel and Al
```

```

        ke = (2*k(1)*k(2))/(k(1)+k(2));
        aE(i,j) = ke*Dy/dx;
    end
    if (i == imax/2 && j == jmax/2)
        ke = (2*k(1)*k(2))/(k(1)+k(2));
        aE(i,j) = ke*Dy/dx;
    end
    if (i > imax/2 && j < jmax/2) % Al
        aE(i,j) = k(2)*Dy/dx;
    end
    if (i < imax/2 && j > jmax/2) %Cu
        aE(i,j) = k(3)*Dy/dx;
    end
    if (i == imax/2 && j > jmax/2) % interface of Steel and Al
        ke = (2*k(3)*k(4))/(k(3)+k(4));
        aE(i,j) = ke*Dy/dx;
    end
    if (i > imax/2 && j > jmax/2) %Cu
        aE(i,j) = k(4)*Dy/dx;
    end
end
end

for i = 2:imax-1
    for j = 1:jmax-1

        if j==2
            dy=Dy/2;
        else
            dy=Dy;
        end

        if j==jmax-1
            dy=Dy/2;
        else
            dy=Dy;
        end

        if (j < jmax/2 && i < imax/2) % steel
            aN(i,j) = k(1)*Dx/dy;
        end
        if (j == jmax/2 && i < imax/2) % interface of Steel and Cu
            ke = (2*k(1)*k(3))/(k(1)+k(3));
            aN(i,j) = ke*Dx/dy;
        end
        if (j == jmax/2 && i == imax/2)
            ke = (2*k(1)*k(3))/(k(1)+k(3));
            aN(i,j) = ke*Dx/dy;
        end
        if (j > jmax/2 && i < imax/2 ) % Cu
            aN(i,j) = k(3)*Dx/dy;
        end
        if (j < jmax/2 && i > imax/2) %Al
            aN(i,j) = k(2)*Dx/dy;

```

```

        end
        if (j == jmax/2 && i > imax/2) % interface of Al and Fe
            ke = (2*k(2)*k(4))/(k(2)+k(4));
            aN(i,j) = ke*Dx/dy;
        end
        if (j > jmax/2 && i > imax/2) %Fe
            aN(i,j) = k(4)*Dx/dy;
        end
    end
end

Dt = 1;

for j=2:jmax-1
    for i=2:imax-1
        if Tag(i,j) == 1
            ap0(i,j) = (rho(1)*Cp(1)*Dx*Dy)/Dt;
        elseif Tag(i,j) == 2
            ap0(i,j) = (rho(2)*Cp(2)*Dx*Dy)/Dt;
        elseif Tag(i,j) == 3
            ap0(i,j) = (rho(3)*Cp(3)*Dx*Dy)/Dt;
        else
            ap0(i,j) = (rho(4)*Cp(4)*Dx*Dy)/Dt;
        end
    end
end

for j=2:jmax-1
    for i=2:imax-1
        aP(i,j) = ap0(i,j) + aE(i,j) + aE(i-1,j) + aN(i,j) +
        aN(i,j-1);
    end
end

for j=1:jmax
    for i=1:imax
        if Tag(i,j) == 1
            alpha(1) = k(1)/(rho(1)*Cp(1));
        elseif Tag(i,j) == 2
            alpha(2) = k(2)/(rho(2)*Cp(2));
        elseif Tag(i,j) == 3
            alpha(3) = k(3)/(rho(3)*Cp(3));
        else
            alpha(4) = k(4)/(rho(4)*Cp(4));
        end
    end
end

unsteadiness_nd = 1;
n = 0;
e_st = 0.0001;
DTc = Ttop - Tleft;

while unsteadiness_nd >= e_st

```

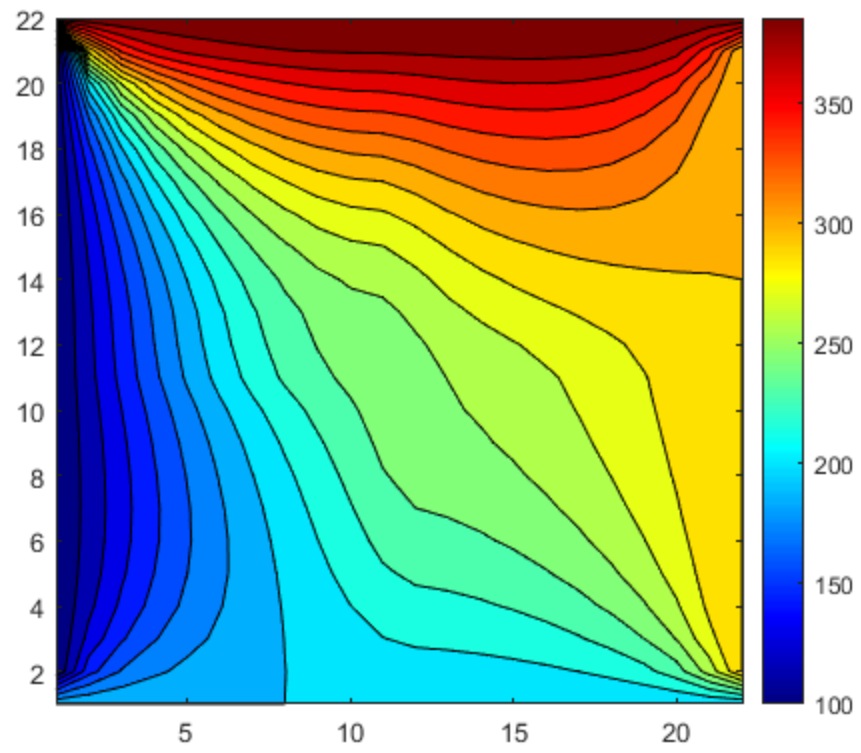
```

n = n+1;
Told = T;
for j = 2:jmax-1
    for i = 2:imax-1
        b(i,j) = (ap0(i,j)*Told(i,j)) + Qgen*Dx*Dy;
    end
end
error =1; N =0; e = 0.0001;

while error >= e
    T_old_iter = T;
    for j = 2:jmax-1
        for i = 2:imax-1
            T(i,j)=aE(i,j)*T(i+1,j)+aE(i-1,j)*T(i-1,j)+aN(i,j)
            *T(i,j+1) +aN(i,j-1)*T(i,j-1)+b(i,j);
            T(i,j)=T(i,j)/aP(i,j);
        end
    end
    error = max(abs(T-T_old_iter));
end
unsteadiness = max(max(abs(T- Told)))/Dt;
unsteadiness_nd = unsteadiness*L*H/((min(alpha))*DTc);
end

contourf(T',20);
colormap jet; axis equal; colorbar;

```



Published with MATLAB® R2020b