International Journal of Innovation in
Mechanical Engineering and Advanced Materials
Vol. 6 (No. 2). 2024. pp. 78-84
Journal homepage: publikasi.mercubuana.ac.id/index.php/ijimeam
DOI: 10.22441/ijimeam.v6i2.25356

Supplementary Document

Heat Distribution Simulation in a Square Aluminum 7075 Plate Using Laplace Equation and MATLAB

Sri Pudjiwati [1,*], Andi F. Sudarma [1], Kontan Tarigan [1], and Deni S. Khaerudini [1,2], and Eldwin Djajadiwinata [3,4]

[1]Department of Mechanical Engineering, Universitas Mercu Buana, Meruya Selatan, Jakarta 11680, Indonesia
[2]Research Center for Advanced Materials, National Research and Innovation Agency (BRIN), Tangerang Selatan 15314, Indonesia
[3]King Abdullah City for Atomic and Renewable Energy (K.A.CARE), Energy Research and Innovation Center at Riyadh, Riyadh 12244, Saudi Arabia
[4]Department of Mechanical Engineering, King Saud University, Riyadh 11451, Saudi Arabia
*Corresponding Authors: pudjiwati@gmail.com (SP)

Appendix

A.1. MATLAB code for error limitation

```
for i=1:9;
        for j=1:9;
                T(i,j)=0
        end
end

%BOUNDARY CONDITIONS
T(:,1)=75 ; %leftboundary
T(:,j)=50 ; %rightboundary
T(1,:)=100 ; %upperboundary
T(j,:)=0 ; %lowerboundary

lamda = 1.5

while(1)
        for j=2:6
                for i=2:6
                        T_old(i,j)=T(i,j);
                        T(i,j)=(T(i+1,j)+T(i-1,j)+T(i,j+1)+T(i,j-1))/4;
                        T(i,j)=lamda*T(i,j)+(1-lamda)*T_old(i,j)
                end
        end
        error=dot((((T(i,j)-T_old(i,j))/T(i,j))*100),...
                (((T(i,j)-T_old(i,j))/T(i,j))*100))
        if error<=1
                break;
        end
end
```

A.2. MATLAB code for the heat distribution on 5 grids model

```
clc;clear;close all;
x0=0; xf=1;
y0=0; yf=1;

nx = 5; ny = 5; %checking for ref book

x=linspace(x0,xf,nx);
```

```
y=linspace(y0,yf,ny);
dx=x(2)-x(1);
dy=y(2)-y(1);

T=zeros(nx,ny); %preallocation matrix

kmax = 1e5; %number of iterations,
% Try until the result gives a constant value or graphic
% A higher number of iterations make better result
% Suggestion for using k=10000, since it may be taking more CPU time but give a good result

for k = 1:kmax % perform iterative calculation
    for i = 2:nx-1
        for j = 2:ny-1
            T(i,j)=((dy^2*(T(i+1,j)+T(i-1,j)))+(dx^2*(T(i,j+1)+T(i,j-1))))/(2*(dx^2+dy^2));
        end
    end
    T(:,1)=75; %Left BC
    T(:,ny)=50; %Right BC
    T(1,:)=100; %Top BC
    T(nx,:)=(4.*T(nx-1,:)-T(nx-2,:))/3; %Neumann BC
end

figure (1)
imagesc(x,y,T)
title('Profile Temperature in square Plate')
xlabel('x-axis')
ylabel('y-axis')
colormap jet
colorbar
```

A.2. MATLAB code for heat distribution on 81 grids model

```
clc;clear;close all;
x0=0; xf=1;
y0=0; yf=1;

nx = 81; ny = 81; %checking for ref book

x=linspace(x0,xf,nx);
y=linspace(y0,yf,ny);
dx=x(2)-x(1);
dy=y(2)-y(1);

T=zeros(nx,ny); %preallocation matrix

kmax = 1e5;%number of iteration,
%try until the result gives a constan value or graphic
%A higher number of iterations make better result
% I suggest using k=10000, (it may be taking more CPU time but give a good result

for k = 1:kmax % perform iterative calculation
    for i = 2:nx-1
        for j = 2:ny-1
            T(i,j)=((dy^2*(T(i+1,j)+T(i-1,j)))+(dx^2*(T(i,j+1)+T(i,j-1))))/(2*(dx^2+dy^2));
        end
    end
    T(:,1)=75; %Left BC
    T(:,ny)=50; %Right BC
    T(1,:)=100; %Top BC
```

```
    T(nx,:)=(4.*T(nx-1,:)-T(nx-2,:))/3; %Neumann BC
end

figure (1)
imagesc(x,y,T)
title('Profile Temperature in square Plate')
xlabel('x-axis')
ylabel('y-axis')
colormap jet
colorbar
```

A.3. MATLAB code for the heat distribution on 101 grids model

```
clc;clear;close all;
x0=0; xf=1;
y0=0; yf=1;

nx = 101; ny = 101; %checking for ref book

x=linspace(x0,xf,nx);
y=linspace(y0,yf,ny);
dx=x(2)-x(1);
dy=y(2)-y(1);

T=zeros(nx,ny); %preallocation matrix

kmax = 1e5;%number of iteration,
%try until the result gives a constan value or graphic
%A higher number of iterations make better result
% I suggest using k=10000, (it may be taking more CPU time but give a good result

for k = 1:kmax % perform iterative calculation
    for i = 2:nx-1
        for j = 2:ny-1
            T(i,j)=((dy^2*(T(i+1,j)+T(i-1,j)))+(dx^2*(T(i,j+1)+T(i,j-1))))/(2*(dx^2+dy^2));
        end
    end
    T(:,1)=75; %Left BC
    T(:,ny)=50; %Right BC
    T(1,:)=100; %Top BC
    T(nx,:)=(4.*T(nx-1,:)-T(nx-2,:))/3; %Neumann BC
end

figure (1)
imagesc(x,y,T)
title('Profile Temperature in square Plate')
xlabel('x-axis')
ylabel('y-axis')
colormap jet
colorbar
```