Contents


```
clc
clear all
```

Setting up Initial Starting point

a0 is starting point of DESIGN VARIABLE uses sinusoidal parameterization

```
%a0 = [3 0.1 0.5 0.3 0.8 0.2 0.9 1 1 0.1 0.1 0.1]';
%a0 = [3 0 0 0 0 0 0 0 0 0 0]';
%a0 = [3 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]';
a0 = 0.12*ones(21,1); a0(1) = 3;
%a0 = ones(101,1); a0(1) = 3;a0(2:end) = rand(100,1);
%a0 = ones(101,1); a0(1) = 3;a0(2:end) = -1+2*rand(100,1);
%a0 = ones(31,1); a0(1) = 3;a0(2:end) = rand(30,1);
%a0 = 0.5*ones(21,1); a0(1) = 3;
%a0 = 0.11*ones(50,1); a0(1) = 3;
```

Setting up Constraints Matrix

Sets up the Constraints Matrix A which holds the coefficients that are linearly added up with design variables to generate the profile on the top

```
%------------------------------------------------------------------------

hmin = 1; % min height of radiator in cm
hmax = 5; % max height of radiator in cm
L = 5; % Length of Radiator in cm
n = length(a0); % length of Design var
x = (0:0.02:5)'; % descretization of x-axis

for i=1:length(x)
    for j=1:length(a0)
        if(j==1)
            A1(i,j) = 1;
        else
            A1(i,j) = sin(2*pi*(j-1)*x(i)/L);
        end
    end
end

A = [A1;-A1];
b = [ones(length(x),1)*hmax; -ones(length(x),1)*hmin];
```

optimization algorithm - fmincon used

```
options = optimoptions(@fmincon,'MaxIter',60,'Display','iter');
[X,fvalue,exitflag,output] = fmincon(@CalcFlux_obj,a0,A,b,[],[],...
    [],[],[],options);
```

Plotting

```
[flux_final,T_final,dTdX,XY] = plot_profile(X,a0);
```

Setting up Objective function

It calls the functions - Calc_h to generate the profile which is a funciton of the design variable 'a' It calls upon the function CalcFlux.m to calculate the flux CalcFlux_obj returns the -Flux calculated by CalcFlux.m Objective function only requires input - the design variable 'a'

```
%--------------------------------------------------------------------------

function Flux = CalcFlux_obj(a)
    L = 5; % cm
    Kappa = 20; % W/(m.K)
    T_top = 20; % deg cel
    T_btm = 90; % deg cel
    x = (0:0.02:5)';

    % calculate h
    h = Calc_h(x,a,L);

    % set Nx and Ny
    nx = length(h)-1;
    ny = 250;

    % Calculate Flux
    [Flux,~,~,~] = CalcFlux(L,h,nx,ny,Kappa,T_top,T_btm);

    % Negate Flux for maximization problem
    Flux = -1*Flux;
end
```

plot final profile

This function generates a plot comparing the Optimized profile with the initial profile and also another plot containing the zoomed optimal profile This function takes inputs: X- Optimized design, a0 - initial design.

```
%--------------------------------------------------------------------------

function [flux_final,T_final,dTdX,XY] =  plot_profile(X,a0)
x = (0:0.02:5)';
L = 5;
T_top = 20;
T_btm = 90;
kappa = 20;
h = Calc_h(x,X,L);
nx = length(h)-1;
ny = 150;
[flux_final,T_final,dTdX,XY] = CalcFlux(L,h,nx,ny,kappa,T_top,T_btm);
```

```
h0 = Calc_h(x,a0,L);
figure(1);
plot(x,h,'k');
hold on;
plot(x,h0);
plot([0,0],[0,h(1)],'k');
plot([5,5],[0,h(end)],'k');
plot([0,5],[0,0],'k');
legend('Optimized Profile','Starting Profile','Left Edge',...
    'Right Edge','Width');
axis([-2 7 -6 12]);
title('Optimized v Starting Profile');
xlabel('x');
ylabel('h');

figure(2)
plot(x,h,'k');
title('Optimized Profile - Zoomed');
xlabel('x');
ylabel('h');

end
```

Calculate h - Profile height

Function returns the profile height h - h(x;a) Function takes inputs - x: discretization about X-axis, a: the design variable, L: Length of the heat exchanger

```
%-------------------------------------------------------------------------

function h = Calc_h(x,a,L)
% Uses sinusoidal parameterization to create profile
% h(x) = a(1) + sigma (a(j)*sin(2*pi*(j-1)*x(i)/L)) j:1(1)N
for i=1:length(x)
    S = 0;
    for j=2:length(a)
        S = S + a(j)*sin(2*pi*(j-1)*x(i)/L);
    end
    h(i,1) = a(1) + S;
end
end
```