Project 1: Heat Exchanger Design

Unique ID: 5062

September 24, 2021

Abstract

Constrained optimization using a gradient based method of a heat exchanger is performed to maximize the heat flux per unit length. A finite volume implementation of the 2-D heat equation is used to determine heat flux. The exchanger profile is represented by a sum of sine waves of varying frequency. Variation of the number of design variables used to define the exchanger profile indicates higher frequency sine waves for the profile lead to greater heat flux likely due to an increased surface area. Convergence of the optimal geometry for 9 design variables indicates an asymptote for maximum heat flux per unit length of roughly 25900 [W/m]. Simulation with 250k mesh volumes (500 × 500) provides a value within 1% of this asymptote estimate. A convergence history for determining the optimal geometry with 9 design variables and 40k mesh volumes (200 × 200) is also performed indicating low first order optimality and no constraint violations.

1 Analysis Method

The analysis for this problem is performed using a finite volume discretization to approximate the heat equation. For this case, the 2-dimensional steady heat equation is used and can be seen in Equation 1 where T is the temperature [K], k is the coefficient of thermal conductivity [W/m·K], and Ω is the domain over which we are modeling. Also, periodicity of the exchanger section is required.

$$\nabla \cdot (k \nabla T) = 0, \quad \forall x \in \Omega \tag{1}$$

The 2-D finite volume discretization used breaks the simulation domain into a number of quadrilaterals over which the heat equation is applied. The heat equation is integrated over the surfaces of these quadrilaterals and the boundary conditions are applied on the sides of the quadrilaterals in contact with the air or water. Integration of the domain leads to a linear system of equations which can be solved to determine the temperature in each location. Integrating over the width of the exchanger is then used to determine the heat flux per unit length.

This model assumes that the temperatures of each side of the exchanger in contact with the fluids is the temperature of the respective fluid which is a simplification of the convection process occurring at the boundaries. Simplifying like this allows implementation of Dirchlet conditions at the top and bottom of the exchanger for the boundary conditions with T_{air} at the top and T_{water} at the bottom. This method does require that the top profile of the exchanger be expressed as an explicit function of the horizontal dimension. Also, extreme differences in the heights between points can present issues for the solver due the compression/expansion of the mesh vertically at low/high points in the profile.

2 Geometry Parameterization

The geometry of the heat exchanger is discretized into quadrilateral sections which are compatible with the finite volume method described above. The upper profile of the exchanger is modeled via Equation 2 as a function of the design variables $a_1, ..., a_N$.

$$h(x) = a_1 + \sum_{k=2}^{N} a_k \sin\left(\frac{2\pi(k-1)}{L}x\right) \tag{2}$$

In the above equation, L is the width of the exchanger section (which can be repeated) and x is the horizontal position in the section. Selection of the sine function provides several advantages for solving this problem. It is periodically repeating which ensures that when repeating this section the full exchanger will have a continuous surface. The smooth curve also ensures that the discretization will be accurate when simulating the physics as sharp points or near vertical slopes can provide issues for this solver. The discretization of this geometry for a 10×10 grid can be seen below in Figure 1 for an example with a single period sine wave.

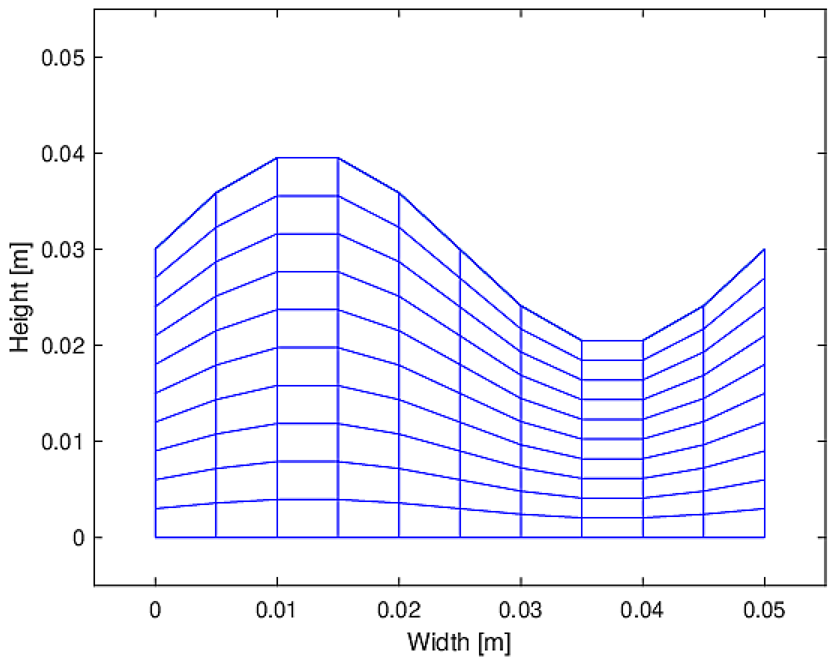

Figure 1: 10×10 2-D discretization of sinusoidal exchanger geometry

The geometry of the exchanger is constrained vertically with a minimum height $h_{\text{min}} = 0.01$ [m] and a maximum $h_{\text{max}} = 0.05$ [m]. These constraints are implemented using a system of linear inequalities which is passed into the solver as stated in the next section.

3 Problem Statement and Optimzation Method

The goal of this optimization is to find the design variables which maximize heat flux per unit length through the heat exchanger. As stated above the design variables determine the upper profile of the exchanger which in turn determines the heat flux of the geometry. The top of the exchanger is in contact with $20°C$ air, the bottom with $90°C$ water, and the exchanger has a thermal conductivity of $20W/(m \cdot K)$.

Optimization for this problem was performed used MATLAB's `fmincon` – a gradient based constrained optimization solver which minimizes the input objective function. The `sqp` algorithm available in this solver was selected to perform the optimization.

The code included in the Appendix was used to run the optimization routine. Listing 1 (height calculation) takes the design variables and determines the top profile of the exchanger according to Equation 2. Listing 2 (the objective function) uses this height information to determine the heat flux per unit length of geometry in question and returns the reciprocal of this value as the objective function value. Listing 3 (constraint calculation) creates the necessary matrices used in the solver to enforce the linear height constraints. Listing 4 is used to execute a single run of the optimization routine based on a random input vector for the initial guess. Note that provided code was used to discretize the geometry as well as perform the heat flux calculation.

Listing 5 performs a sweep over several different number of design variables to investigate the effect changing the number has on the optimized geometry and consequently maximum heat flux per unit length. For each number of design variables, a sweep over the number of quadrilaterals in the discretization of the exchanger shape (`Nx` $\times$ `Nx` as we set `Nx` $=$ `Ny`) is performed. Five runs of the optimization routine at each combination of number of design variables and `Nx` are completed in order to ensure finding the best minimum within the constraints and not a false local minimum. `Nx` is increased until there is less than 1% change between consecutive values of $q_{\max}$ or the max `Nx` value is reached.

The code used for the mesh granularity convergence study can be found in Listing 6. This code takes the optimal exchanger profile determined for 9 design variables and sweeps over values of `Nx` (selection of the specific exchanger profile is elaborated upon in later sections). Listing 7 contains the code used for an example optimization convergence history run.

4 Results

Testing began by running the sweep over number of design variables and values of `Nx` using the code in Listing 5. The results of this run are presented below in Table 1.

Table 1: Sweep over number of design variables and Nx

`N`	$q_{\max}[W/m]$	`Nx`	`x`
3	7535.2	200	[0.03, 0, 0.02]
5	13544	500	[0.03, 0, 0, 0, 0.02]
7	19675	1000	[0.03, 0, ..., 0, 0.02]
9	25665	500	[0.03, 0, ..., 0, 0.02]

The ellipsis in the table indicates all '0' entries for the middle of the array. Note the test for `N` $= 9$ was interrupted early as it became too computationally intensive. The table indicates that the geometry created with 9 design variables led to the highest heat flux per unit length with a value of 25665 W/m. The shape of the exchanger for this case can be seen in Figure 2. The equation representing the profile of the top edge is seen below in Equation 3 (found by plugging the optimal values for the design variables into Equation 2).

$$h(x) = 0.03 + 0.02 \cdot \sin(0.8\pi \cdot x) \text{ [m]} \tag{3}$$

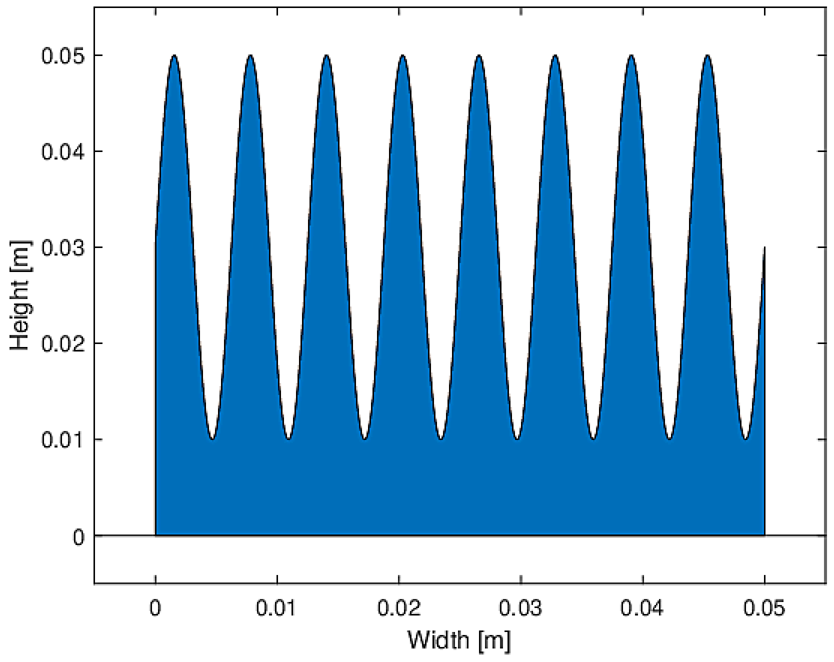

Figure 2: Shape of optimal exchanger for 9 design variables

Taking the optimal geometry for 9 design variables, a convergence study over the number of mesh volumes in each direction was performed with Listing 6. The results of the study can be seen in Figure 3.

A convergence history for the case of 9 design variables and 40k mesh volumes is presented in Figure 4 (created using the code in Listing 7). This routine had to be run several times until the optimal solution was found as a result of the likely presence of other local minima. First order optimality here is a measure of how close a point is to optimal (mathematically the norm of the gradient). Feasibility is a measurement of constraint violations. In this context, low feasibility indicates few to no constraint violations.

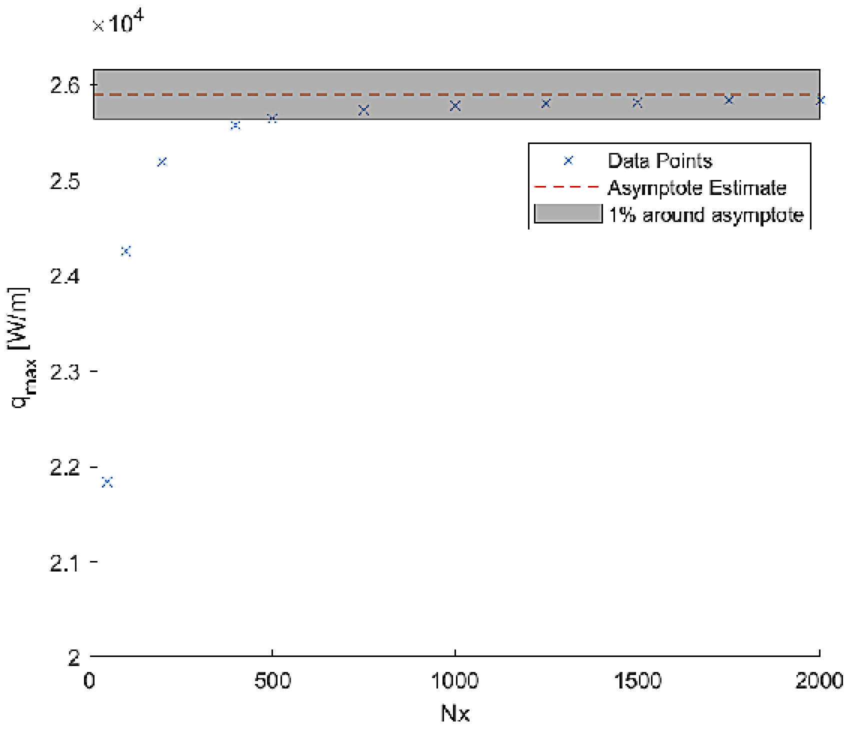

Figure 3: Convergence study for optimal geometry with 9 design variables

5 Discussion

The results of the initial test run over the number of design variables show that the maximum heat flux per unit length of the exchanger in general increases as the number of variables increases. This relationship seems logical as more design variables allow for more complexity which leads to more designs to choose from. A constant trend can also be seen in the results of Table 1 as the optimal design variable array always has 0.03 as the first entry, 0.02 as the last entry, and the rest filled with zeros. One can conclude from this that the optimal design is most likely the one that fits the most periods of the sine wave into the exchanger width with the maximum amplitude allowed by the constraints. This result makes sense as this creates the largest surface area at the top of the exchanger which reduces the convective resistance of the surface (allowing more heat flow). Based on this result, it is likely that more design variables will lead to even higher heat flux per unit length as higher frequency sine waves are possible with more design variables.

The convergence study in Figure 3 indicates an asymptote of approximately 25900 W/m for the max heat flux per unit length in the case of 9 design variables (when increasing `Nx`). This asymptotic behavior is of interest as it represents what happens as `Nx` $\to \infty$ which is the best indication of reality (minimization of error due to discretization). To be within 1% of this bound (reasonably close to the asymptote), 500 mesh volumes in each direction (250k total) are required. Being able to compute a

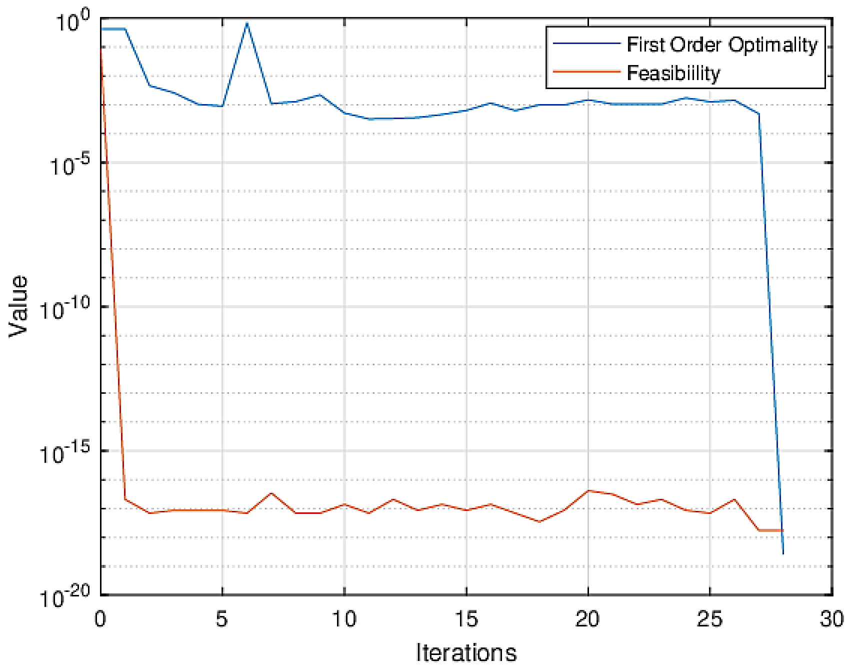

Figure 4: Convergence history for 9 design variables and 200 × 200 mesh

good approximation with less mesh volumes is desireable as computation time for greater numbers of mesh volumes grows considerably (faster than linear) causing simulations with `Nx` approaching infinity to be not feasible. Similar studies can be performed for other numbers of design variables to determine the asymptotic behavior for these geometries.

The convergence history presented in Figure 4 demonstrates the progress of the optimization routine towards the local minimum. First order optimality remains roughly constant over the course of the routine and then drops to 2.57×10^{-19} at the last iteration indicating the arrival at what is likely a local minimum. The feasibility of the initial guess (iteration 0) marks the violation of some of the constraints (which is understandable as this was from a random guess). However, the first step of the optimization routine brings this value to roughly 10^{-17} and remains here for the rest of the routine, indicating that constraints are being obeyed.

A Source Code

Listing 1: Height calculation function

```
function [h] = calcHeight(a,L,Nx)
%calcHeight - determine exchanger profile based on design variables
%INPUTS:
%   a  - array of design variables
%   L  - length of exchanger section [m]
%   Nx - number of section
%OUTPUT:
%   h  - height array of exchangers

h = a(1)*ones(Nx+1,1);
x = linspace(0,L,Nx+1);

for i = 1:Nx+1
    for k = 2:length(a)
       h(i) = h(i) + a(k)*sin((2*pi*(k-1)*x(i))/L);
    end
end

```

```
end
```

Listing 2: Objective function calculation

```
function [e] = objective(a,L, Nx, Ny, kappa, Ttop, Tbot)
%objective - objective function for optimization in terms of design
%variables
%INPUTS:
%   a - array of design variables
%   L - width of exchanger section
%   Nx/Ny - number of quadrilateral discretizations in x/y direction
%   kappa - coefficient of thermal conductivity
%   Ttop - temperature at air surface of exchanger
%   Tbot - temperature at water surface of exchanger
%OUTPUT:
%   e - objective function (reciprocal of flux)

    h = calcHeight(a,L,Nx);
    e = 1/CalcFlux(L,h,Nx,Ny,kappa,Ttop,Tbot);
end
```

Listing 3: Linear constraints calculation

```
function [Aineq,bineq] = calcConstraints(n,Nx,x,L,hmin,hmax)
%calcConstraints - determines linear constraints for the optimization
%INPUTS:
%    n - number of design variables
%    Nx - number of quadrilateral approximations in x direction
%    x - x direction values
%    L - width of exchanger
%OUTPUTS:
%    Aineq,bineq - linear constraints in the form Aineq x <= bineq

nvar = n; % number of design variables
Aineq = zeros(2*(Nx+1),nvar);
bineq = zeros(2*(Nx+1),1);
for i = 1:(Nx+1)
  % first, the upper bound
  Aineq(i,1) = 1.0; % this coefficient corresponds to variable a_1
  for k = 2:nvar
    Aineq(i,k) = sin(2*pi*(k-1)*x(i)/L); % this coefficient corresponds to variable
    a_k
  end
  bineq(i) = hmax; % the upper bound value
  % next, the lower bound; we use ptr to shift the index in Aineq and bineq
  ptr = Nx+1;
  Aineq(ptr+i,1) = -1.0; % note the negative here!!! fmincon expects inequality in a
    form A*x < b
  for k = 2:nvar
    Aineq(ptr+i,k) = -sin(2*pi*(k-1)*x(i)/L); % again, a negative
  end
  bineq(ptr+i) = -hmin; % negative lower bound
end
end
```

Listing 4: Optimization run function

```
function [q_max,x] = runOpt(n,Nx_in,Ny_in)
%runOpt - Runs the optimization routine for a number of design variables and Nx/Ny
%INPUTS:
%    N - number of design variables
%    Nx_in/Ny_in - number of mesh volumes in x/y direction
%OUTPUTS:
%    q_max - maximum heat flux per unit length
%    x - design variable values for maximum heat flux

    %% Problem Setup
    Ttop  = 20; %[*C]
    Tbot  = 90; %[*C]
    L     = 0.05; %[m]
    hmin  = 0.01; %[m]
    hmax  = 0.05; %[m]
    kappa = 20; %[W/mK]
    Nx = Nx_in;
    Ny = Ny_in;
    x = linspace(0,L,(Nx+1)); %x values

    obj = @(x) objective(x,L,Nx,Ny,kappa,Ttop,Tbot);

    [Aineq, bineq] = calcConstraints(n,Nx,x,L,hmin,hmax);

    x0 = rand(n,1)*0.05; %random initial condition

    %set output and algorithm
    options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
    [x_min,feval_min] = fmincon(obj,x0,Aineq,bineq,[],[],[],[],[],options);

    %take reciprocal to get heat flux per unit length (fmincon return recip)
    q_max = 1/feval_min;
    x = x_min;
end
```

Listing 5: Driver for sweep over different number of design variables

```
close all; clear; clc;
%% Design Variable Sweep
% Run sweeps over Nx (=Ny) to determine optimal geometry for each number
% of design variables
N = [3,5,7,9];
Nx = [10;100;200;500;1000];
[~,n] = size(N);
qs = zeros(n,1);
xs = -1*ones(n,max(N));
Ns = zeros(n,1);

for i = 1:n %number of design variables
    q_max = 1e-16;
    for j = 1:size(Nx)
        q_last = q_max;
        r = 5; %number of times to repeat optimization
        q_m = zeros(r,1);
        x_m = zeros(r,N(i));
        for k = 1:r %runs over same setup to ensure optimal minimum found
            [q_m(k),x_m(k,:)] = runOpt(N(i),Nx(j),Nx(j));
        end

        [q_max, arg_max] = max(q_m);
        x_max = x_m(arg_max,:);

        if abs(q_last-q_max)/q_last < 0.01 %less than 1% change in value
            break
        end
    end
    qs(i) = q_max;
    xs(i,1:N(i)) = x_max;
    Ns(i) = Nx(j);
end
```

Listing 6: Driver for convergence study

```
%% Convergence Study
%Performs a spacial convergence study for the 9 design variable geometry

a = [0.03;0;0;0;0;0;0;0;0.02];

Ttop  = 20; %[*C]
Tbot  = 90; %[*C]
L     = 0.05; %[m]
hmin  = 0.01; %[m]
hmax  = 0.05; %[m]
```

```
kappa = 20; %[W/mK]

obj = @(x,Nx) objective(x,L,Nx,Nx,kappa,Ttop,Tbot);

Nxs = [10; 50; 100; 200; 400; 500; 750; 1000; 1250; 1500; 1750; 2000];
[n,~] = size(Nxs);
qs = zeros(n,1);

for i = 1:n
   qs(i) = 1/obj(a,Nxs(i))
end

save convergence_study.mat
```

Listing 7: Driver for convergence study

```
%% Convergence History Script
%Optimization run with output to track convergence to minimum

%Setup
n = 9; %number of design variable
Ttop  = 20; %[*C]
Tbot  = 90; %[*C]
L     = 0.05; %[m]
hmin  = 0.01; %[m]
hmax  = 0.05; %[m]
kappa = 20; %[W/mK]
Nx = 200;
Ny = Nx;
x = linspace(0,L,(Nx+1)); %x values

obj = @(x) objective(x,L,Nx,Ny,kappa,Ttop,Tbot);

[Aineq, bineq] = calcConstraints(n,Nx,x,L,hmin,hmax);

x0 = rand(n,1)*0.05;

options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
[x_min,feval_min,~,output] = fmincon(obj,x0,Aineq,bineq,[],[],[],[],[],options);

q_max = 1/feval_min;
x = x_min;

save convergence_history.mat
```