

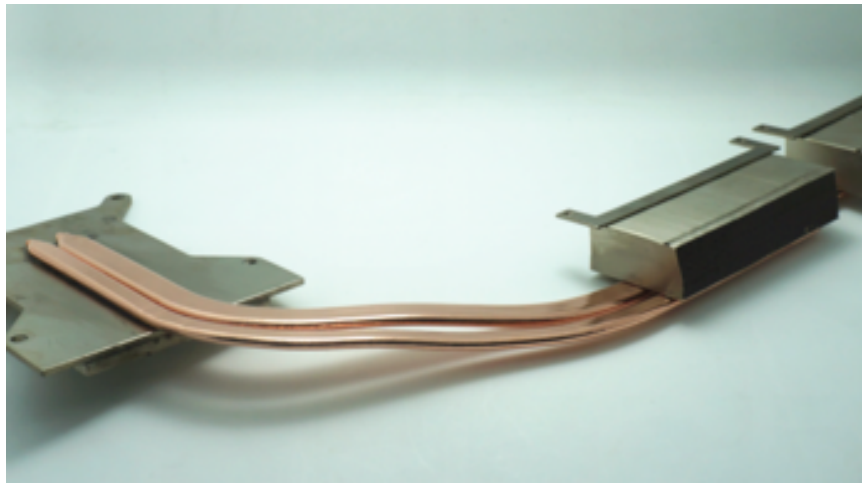
# Project Proposal Methods Fall 2017

Michael Laufer

January 14, 2018

## **Steady State Thermal Simulation of a 2D Heatpipe-based Cooler for Mobile Processors**

This project will attempt to predict the temperature distribution of a characteristic heatpipe based cooler for mobile processors. A cooler of this type is generally comprised of a copper contactor plate and a fin array connected with a super-high effective thermal conductivity element known as a heatpipe. The main challenge in this project will be to modify the Unspy code and implement the new boundary conditions characteristic of the problem. An accurate simulation of this type of cooler has many potential uses in industry that I will elaborate more on in my final project. A basic heatpipe-based cooler can be seen in the following image:



In this work I plan to modify and fine-tune the generic Finite Volume Method solver code that was provided to us in the class. After analyzing the code given along with the physics of the problem at hand, a number of modifications and new elements must be added to the solver. Some of these additions include:

- Explicitly setting the diffusion coefficient (the code implies a value of 1)
- Implementing non-uniform diffusion coefficient
- Implementing the inverse distance-weighted harmonic mean formula for calculating the diffusion coefficient at the faces (needed for dealing with the discontinuous thermal conductivity in the problem).
- Adding convective heat transfer BC
- Adding adiabatic BC
- Verifying Neuman BC for use as a specified heat flux.
- Accelerate the code with a python JIT compiler such as Numba, to allow for much speedier execution. This is a needed step for future use as a base for an unsteady solver.

An additional challenge will be to build the complex geometry and mesh within gmsh. To validate and check the work done in this project, an identical heat transfer simulation will be setup using an industry standard Finite Element solver with the same BCs. The two solutions will be compared as well as the solving time needed for convergence.