

2023 Math Camp

One-Dimensional Heat Transfer

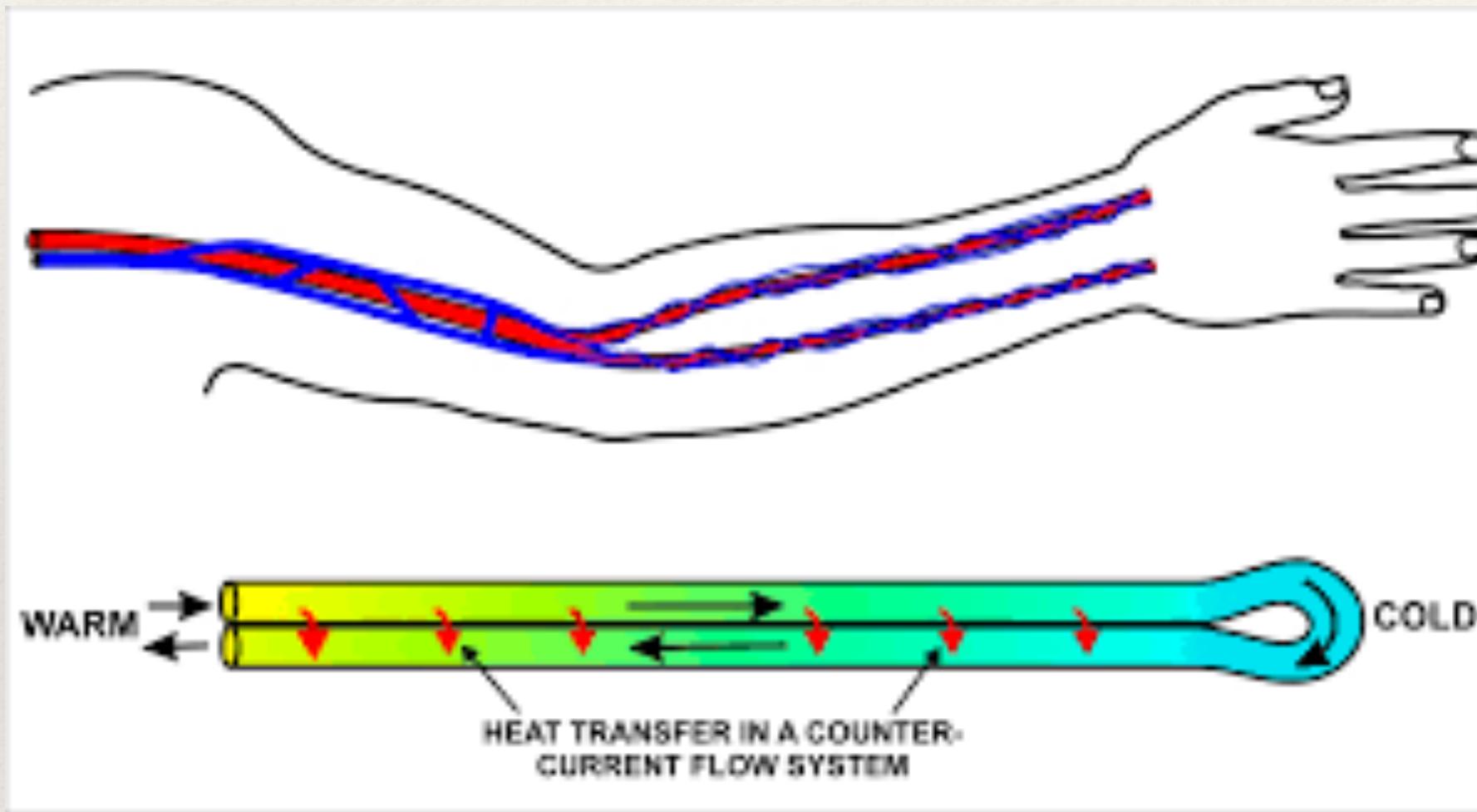
Clint Guymon, PhD PE
Brigham Young University

Who We Are

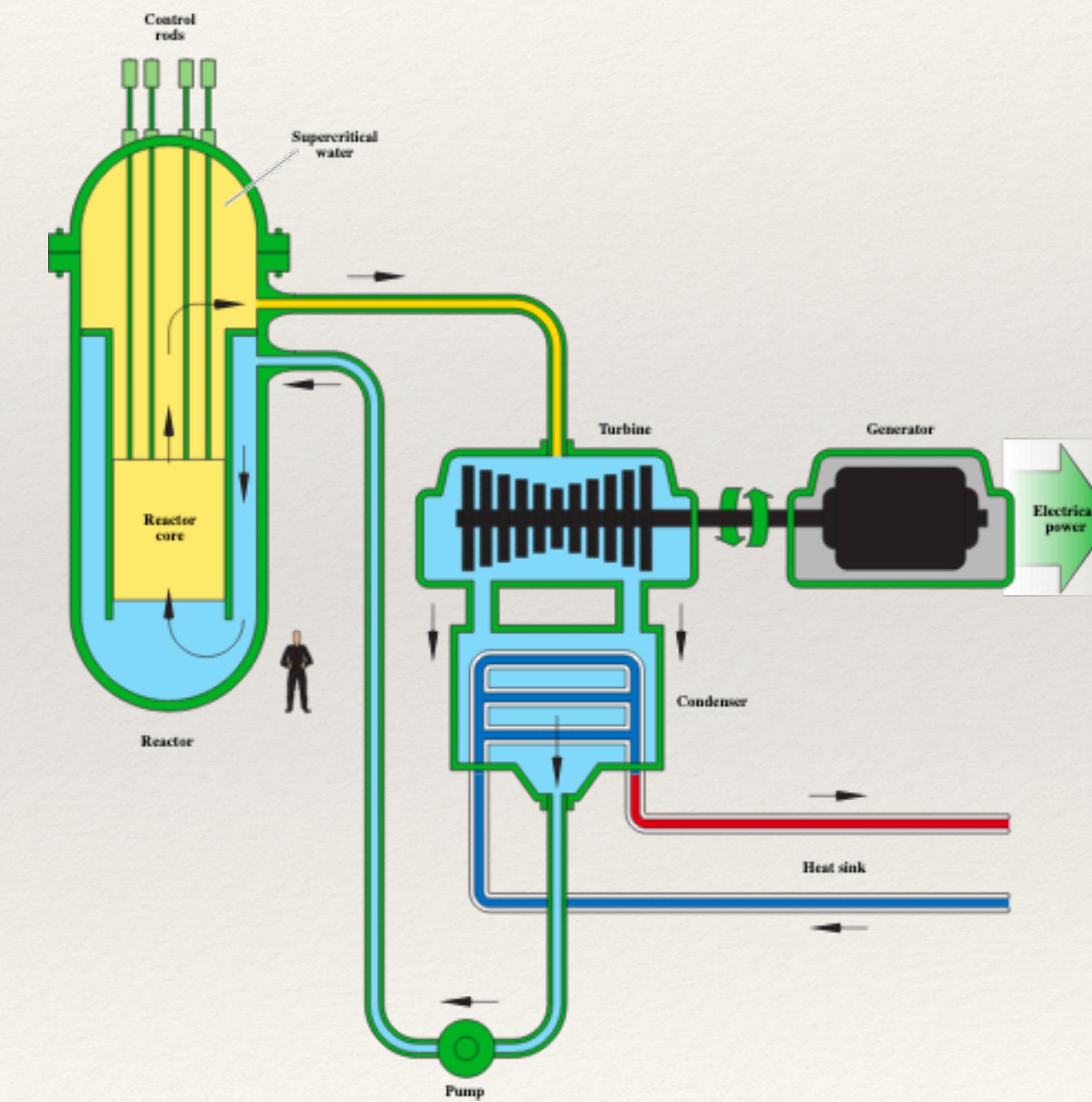


Motivation

- ❖ Understanding of Heat Transfer Details is Critical in Many Applications



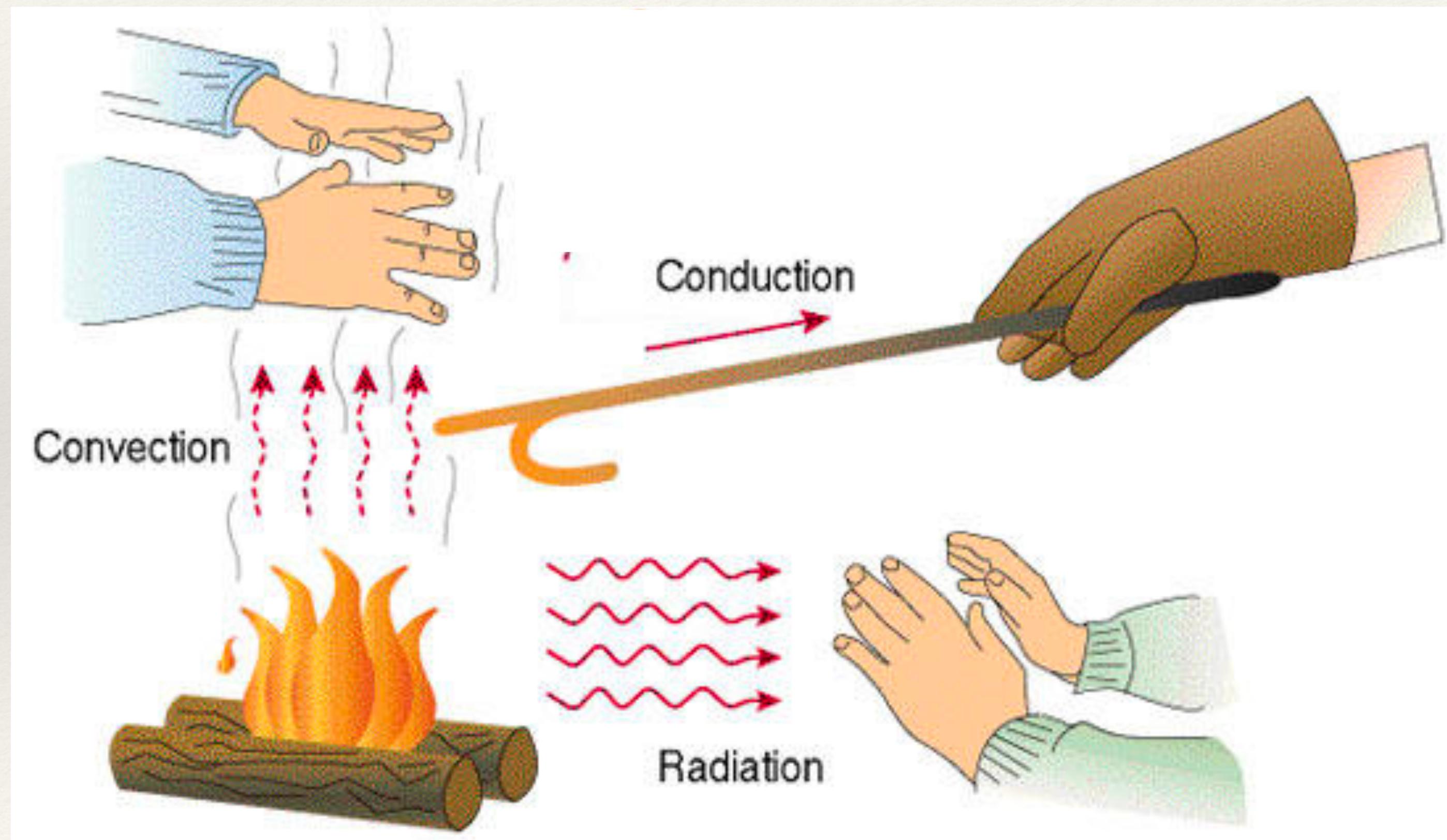
Wikimedia commons



Wikimedia commons

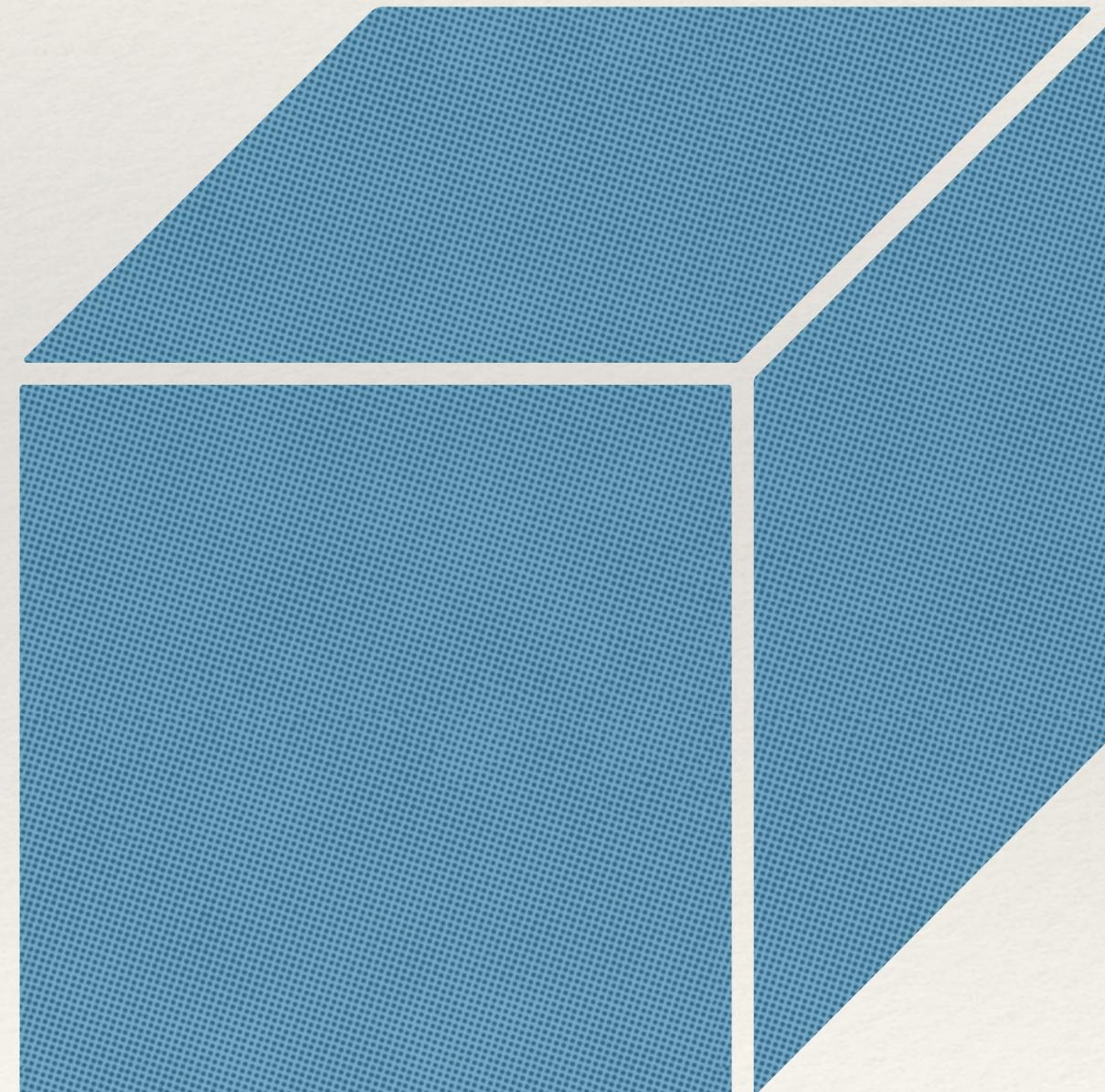
Types of Heat Transfer

- ❖ Conduction, Convection, Radiation



3-D vs 1-D Conduction

- ❖ One dimensional, 1-D, heat transfer is dominated by one direction
- ❖ 3-D it's important to include all three dimensions



—

Transient vs. Steady State

- ❖ Transient indicates conditions change with time
- ❖ Steady state indicates that the conditions are static
 - ❖ Car engine that is warmed up
 - ❖ Other scenarios after an initial startup period



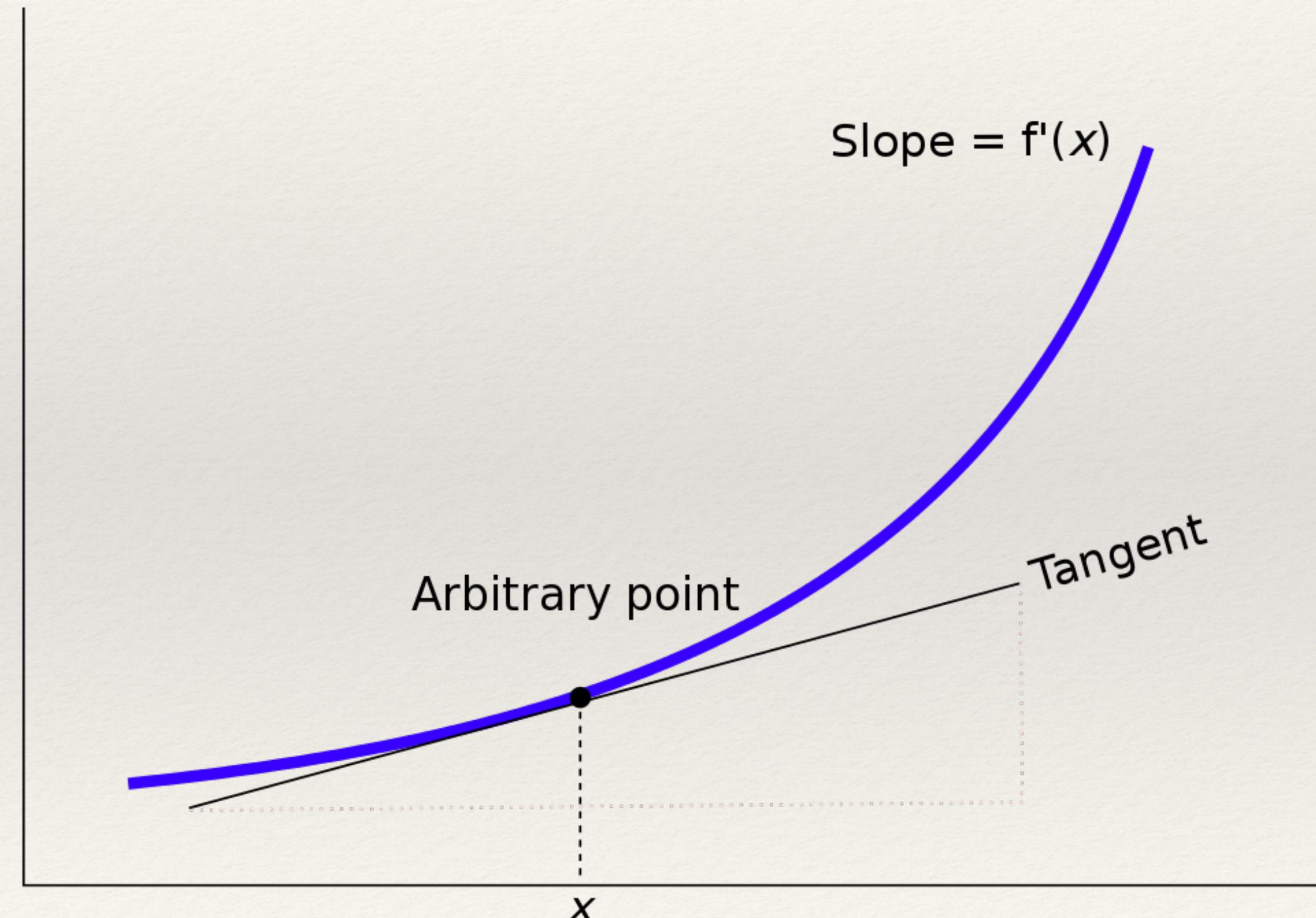
Wikimedia commons

Application Example

- ❖ 1-D Transient Heat Conduction Down a Rod
 - ❖ We'll solve this heat equation three different ways
 - ❖ Analytically
 - ❖ Numerically
 - ❖ Experimentally

Derivatives

- ❖ How things change (temperature) with respect to a variable like position or time



Analytical Solution

❖ Transient Energy Balance:

$$\frac{dE}{dt} = -\frac{\partial q}{\partial x}$$

$$q = -k \frac{\partial T}{\partial x}$$

$$\rho C_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

$$\boxed{\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}}$$

Boundary Condition:
Constant Heat Flux

$$T(x, t) = T_0 + \frac{2q_o''(at/\pi)^{0.5}}{k} \exp\left(\frac{-x^2}{4at}\right) - \frac{q_o''x}{k} \operatorname{erfc}\left(\frac{x}{2\sqrt{at}}\right)$$

Boundary Condition:
Constant End Surface Temperature

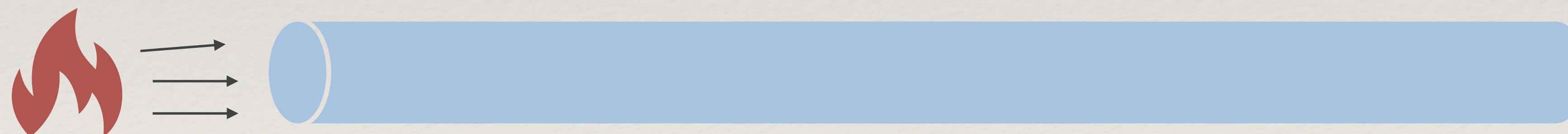
$$T(x, t) - T_i = (T_s - T_i) \cdot \left[1 - \operatorname{erf}\left(\frac{x}{2\sqrt{at}}\right) \right]$$

Boundary Conditions

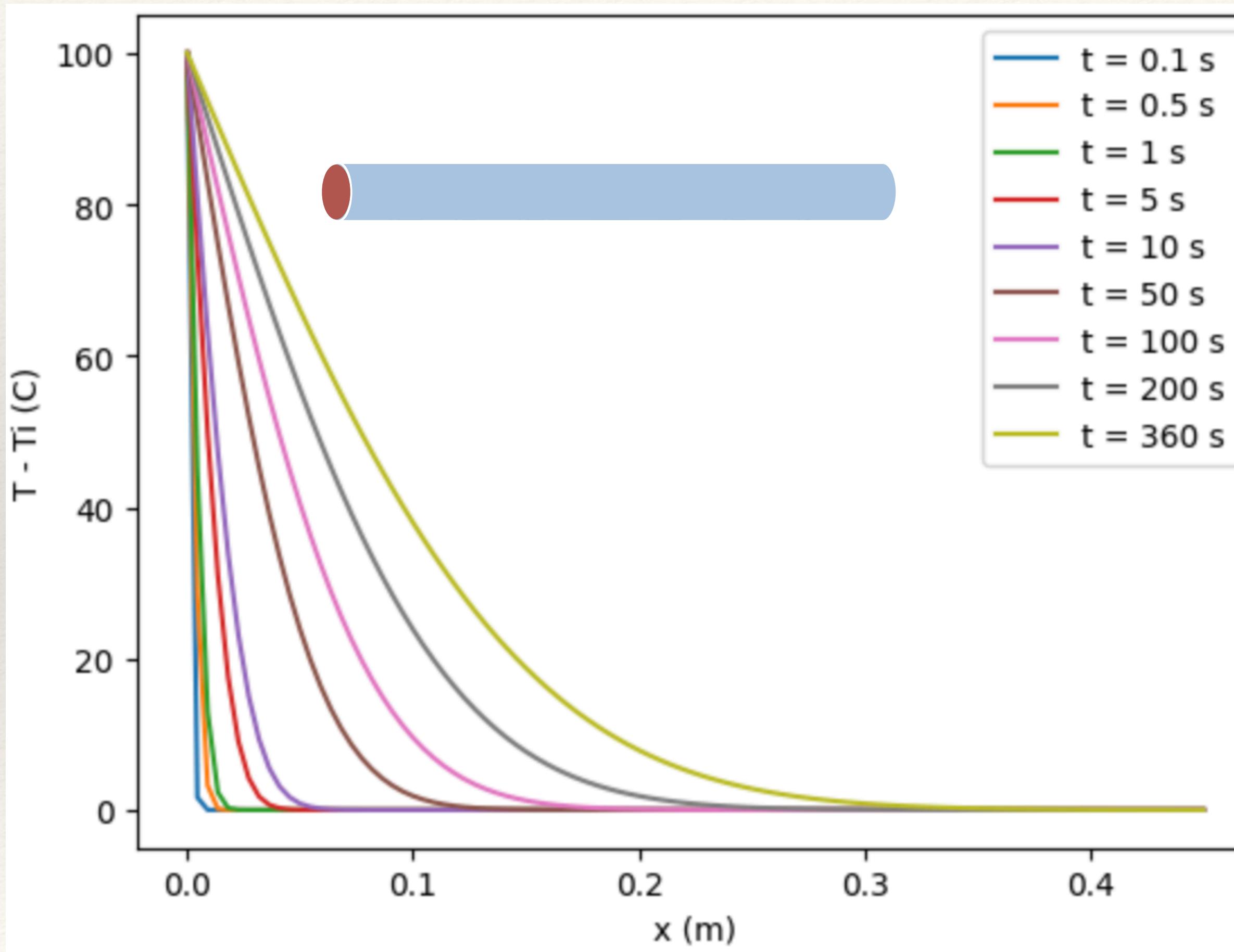
Constant End Surface Temperature



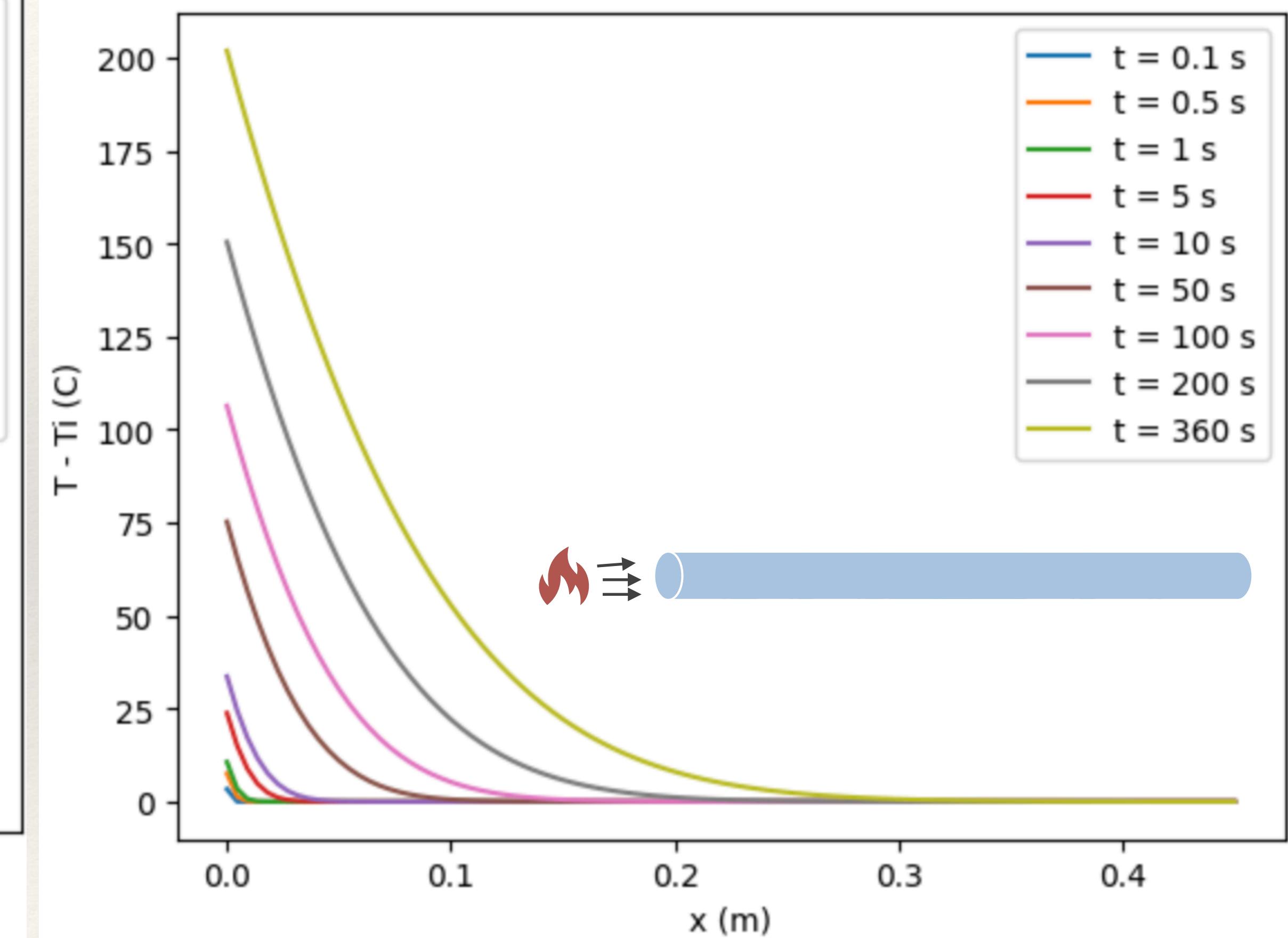
Constant Heat Flux



Analytical Solution Plots

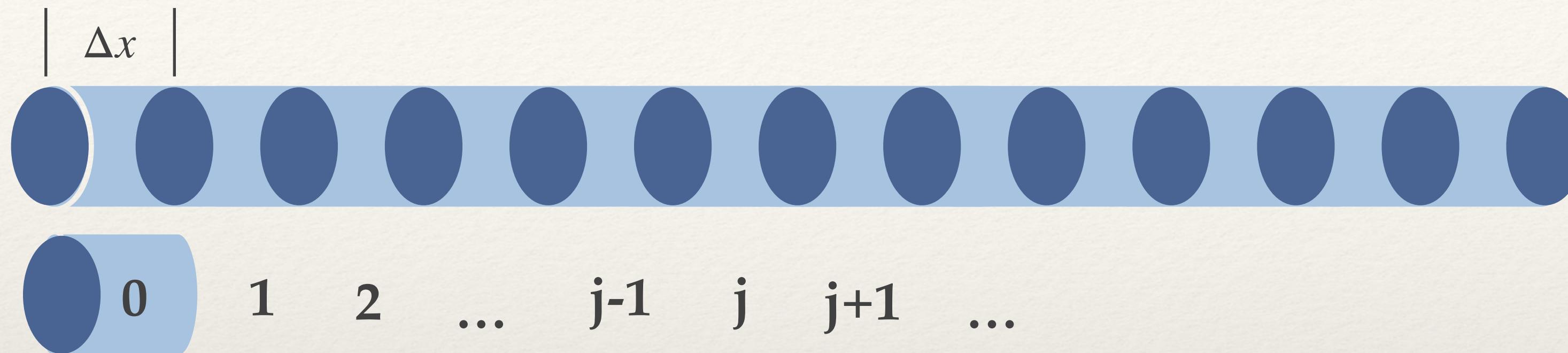


Constant Surface Temperature



Constant Heat Flux

Numerical Solution



- ❖ Process is the same where the energy balance for each segment is found and the time and x-coordinate are discretized or broken up into small pieces

$$T[i + 1, j] = T[i, j] + \alpha \cdot \Delta t / \Delta x^2 \cdot (T[i, j + 1] - 2 \cdot T[i, j] + T[i, j - 1]) \quad \text{at time } i$$

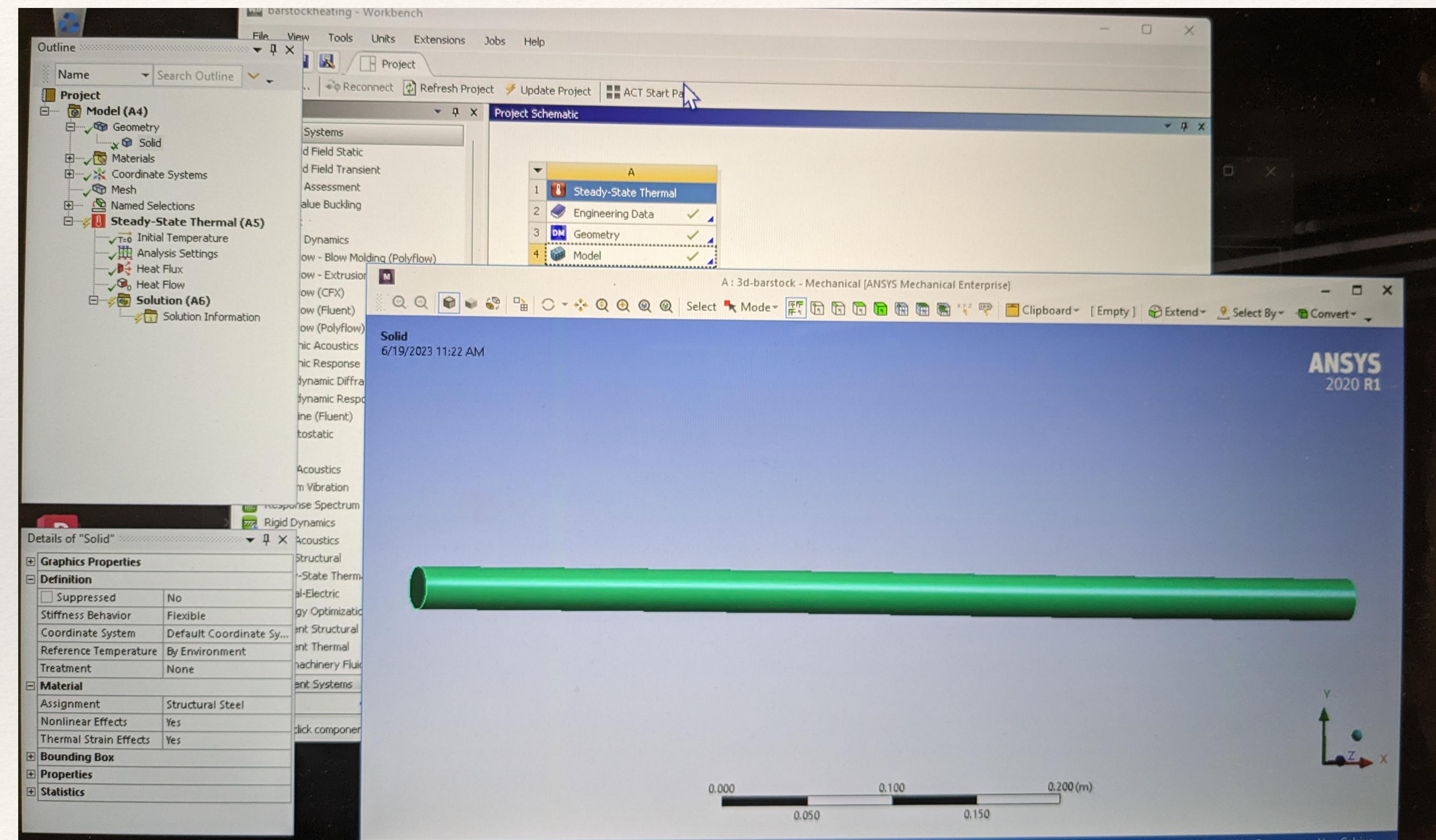
Constant surface temperature

$$T[i + 1, 0] = T[i, 0] + \alpha \cdot \Delta t / \Delta x^2 \cdot (T[i, 1] - T[i, 0]) + \alpha \cdot q_o'' \cdot \Delta t / (\Delta x \cdot k) \quad \text{For 0th node}$$

Constant heat flux

Other Numerical Options

- ❖ Ansys Workbench
- ❖ COMSOL
- ❖ Etc.



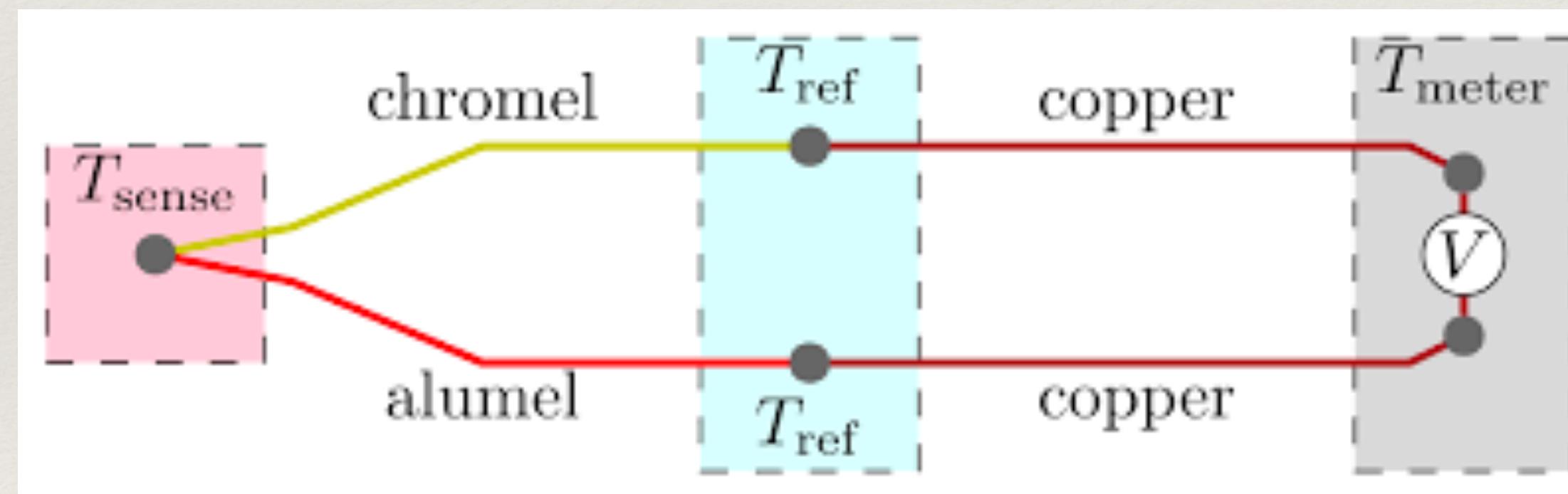
Ansys Workbench

Experimental Option

- ❖ Use a K-type thermocouple
- ❖ Use a low-cost data collection device with an amplifier: ESP32 and MAX6675
- ❖ Add code to allow the ESP32 to connect to your phone via bluetooth
 - ❖ App for Android:
- ❖ Collect the Data

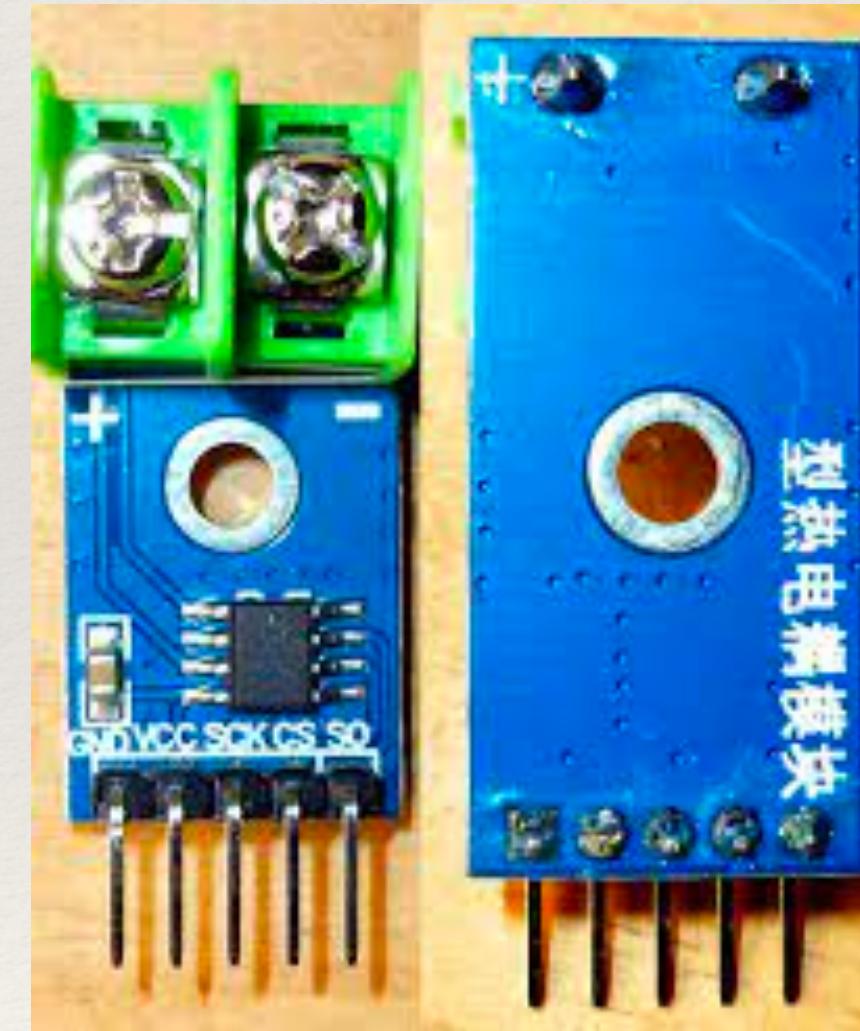
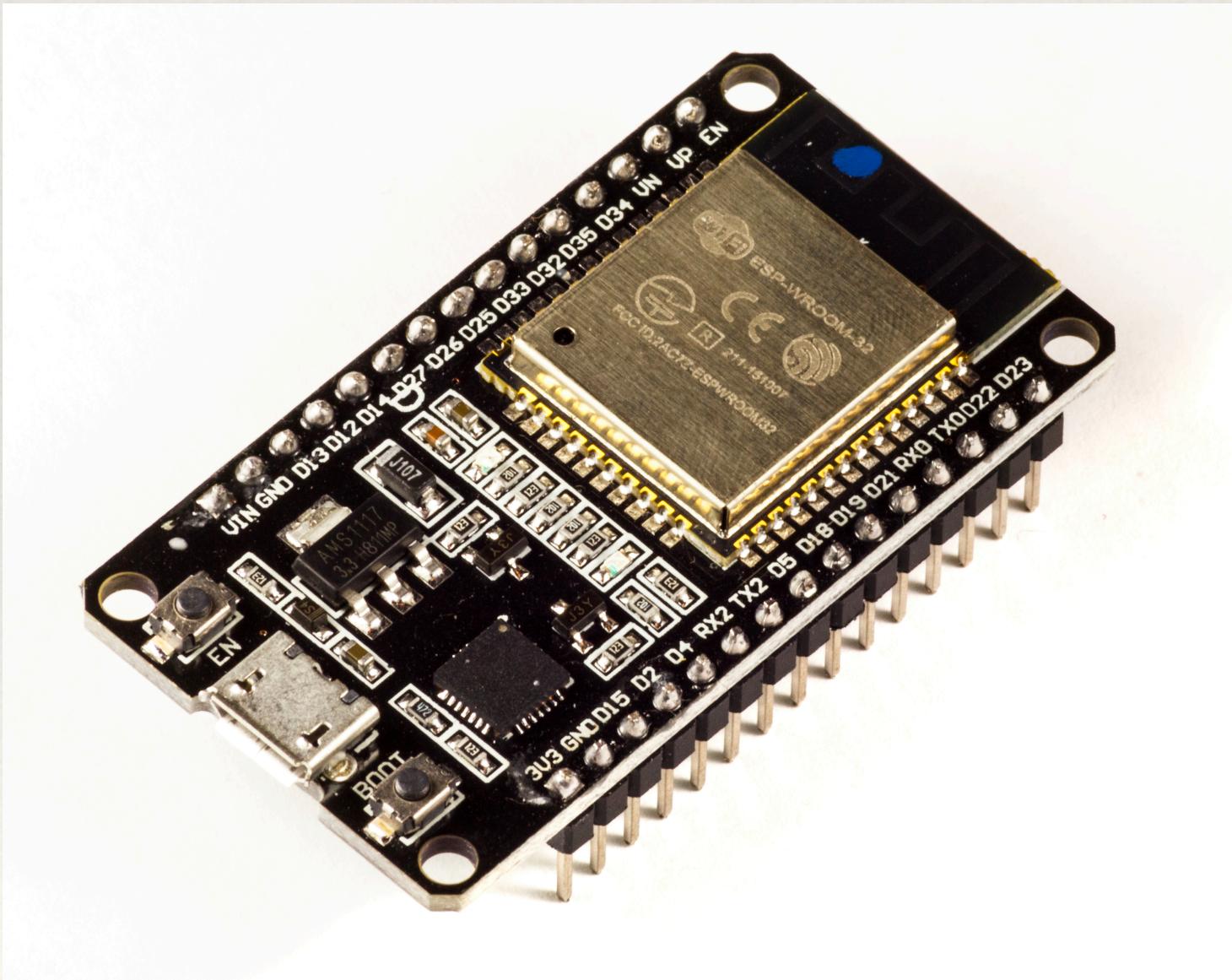
K-type Thermocouple

- ❖ Used to measure temperature
- ❖ Two dissimilar metals connect and the voltage difference indicates the temperature

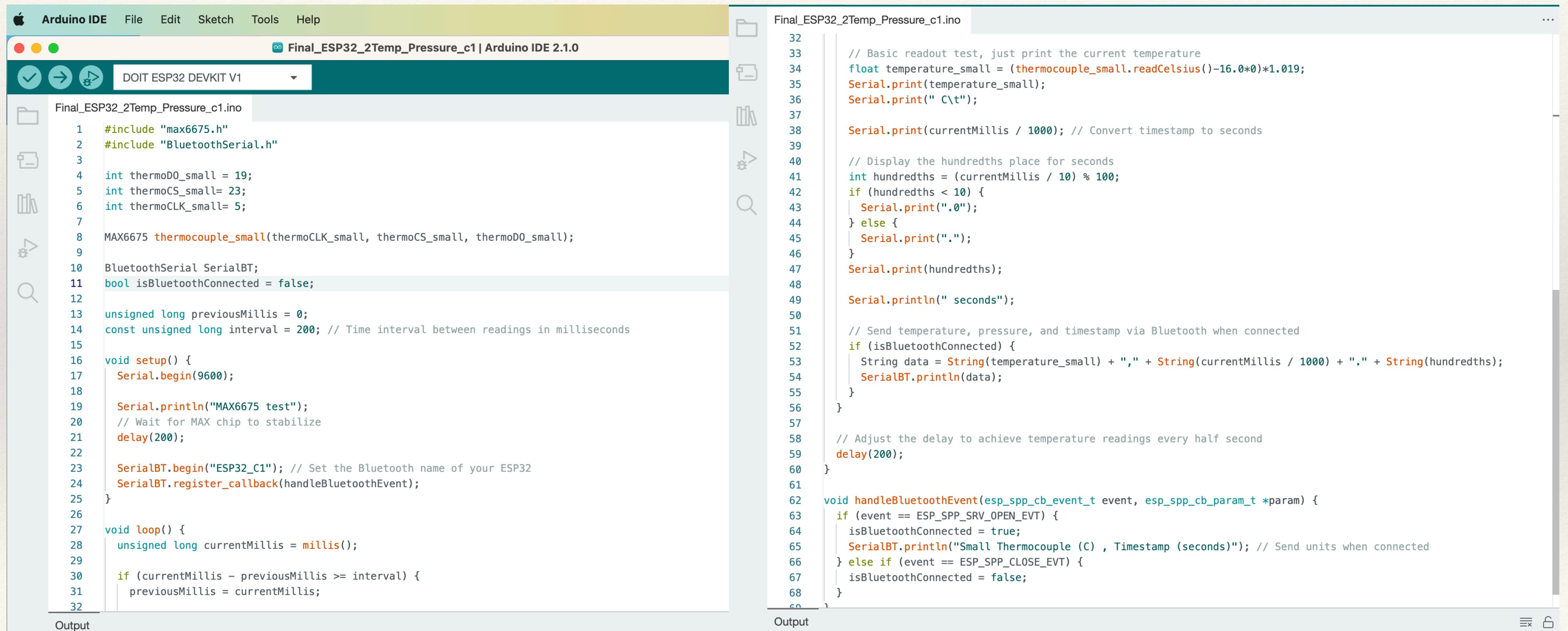


ESP32 & MAX6675

- ❖ A low cost data collection option
- ❖ Similar to an Arduino device
- ❖ A low cost thermocouple amplifier



Data Interface and Collection Code



The screenshot shows the Arduino IDE interface with two tabs open. The left tab contains the code for `Final_ESP32_2Temp_Pressure_c1.ino`, and the right tab contains the code for `Final_ESP32_2Temp_Pressure_c1.ino`. Both tabs are titled "Final_ESP32_2Temp_Pressure_c1.ino".

Left Tab (Code Preview):

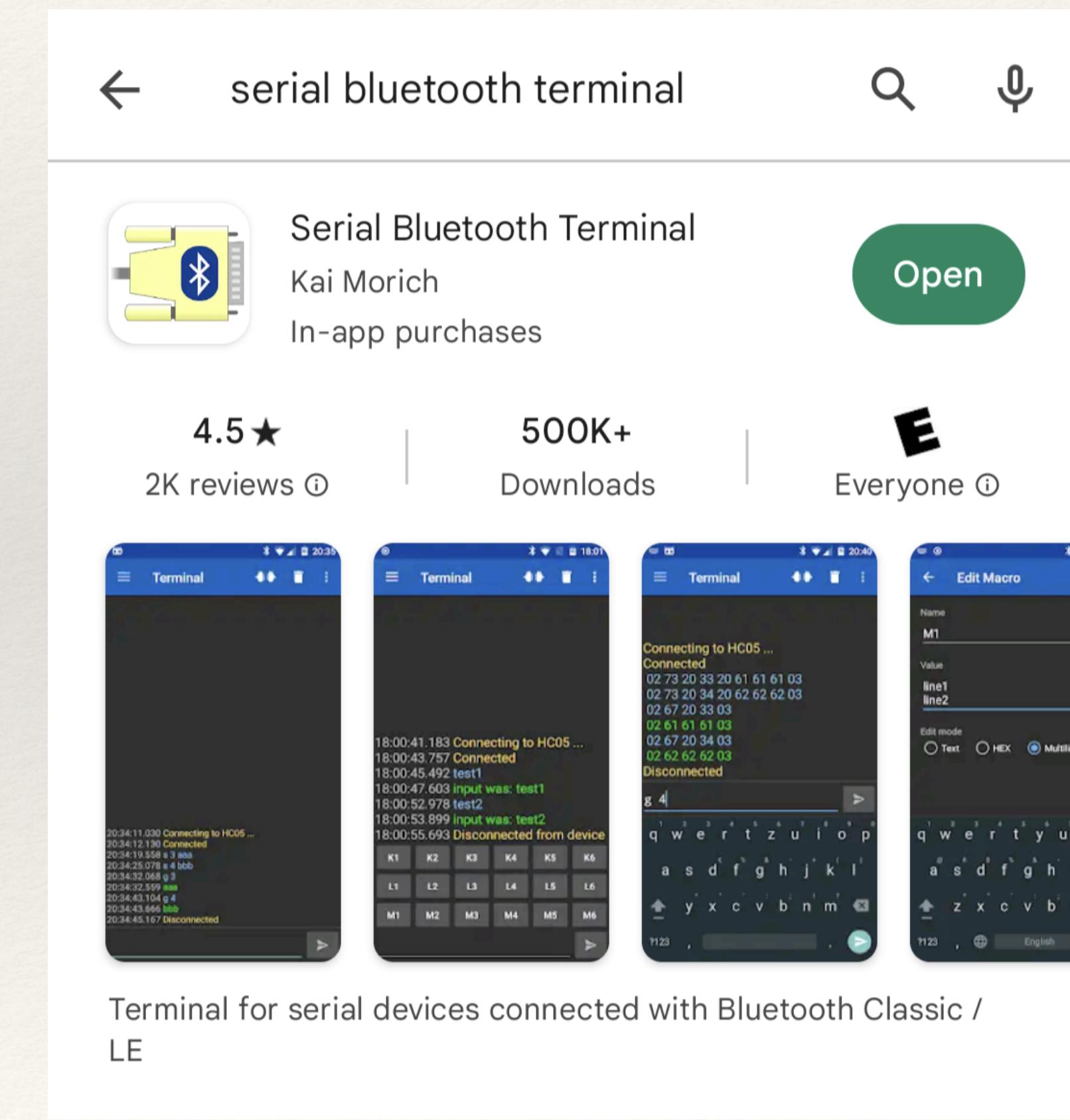
```
1 #include "max6675.h"
2 #include "BluetoothSerial.h"
3
4 int thermoDO_small = 19;
5 int thermoCS_small= 23;
6 int thermoCLK_small= 5;
7
8 MAX6675 thermocouple_small(thermoCLK_small, thermoCS_small, thermoDO_small);
9
10 BluetoothSerial SerialBT;
11 bool isBluetoothConnected = false;
12
13 unsigned long previousMillis = 0;
14 const unsigned long interval = 200; // Time interval between readings in milliseconds
15
16 void setup() {
17   Serial.begin(9600);
18
19   Serial.println("MAX6675 test");
20   // Wait for MAX chip to stabilize
21   delay(200);
22
23   SerialBT.begin("ESP32_C1"); // Set the Bluetooth name of your ESP32
24   SerialBT.register_callback(handleBluetoothEvent);
25 }
26
27 void loop() {
28   unsigned long currentMillis = millis();
29
30   if (currentMillis - previousMillis >= interval) {
31     previousMillis = currentMillis;
32 }
```

Right Tab (Full Code View):

```
32 // Basic readout test, just print the current temperature
33 float temperature_small = (thermocouple_small.readCelsius()-16.0*0)*1.019;
34 Serial.print(temperature_small);
35 Serial.print(" C\t");
36
37 Serial.print(currentMillis / 1000); // Convert timestamp to seconds
38
39 // Display the hundredths place for seconds
40 int hundredths = (currentMillis / 10) % 100;
41 if (hundredths < 10) {
42   Serial.print(".");
43 } else {
44   Serial.print(".");
45 }
46 Serial.print(hundredths);
47
48 Serial.println(" seconds");
49
50 // Send temperature, pressure, and timestamp via Bluetooth when connected
51 if (isBluetoothConnected) {
52   String data = String(temperature_small) + "," + String(currentMillis / 1000) + "." + String(hundredths);
53   SerialBT.println(data);
54 }
55
56 }
57
58 // Adjust the delay to achieve temperature readings every half second
59 delay(200);
60
61
62 void handleBluetoothEvent(esp_spp_cb_event_t event, esp_spp_cb_param_t *param) {
63   if (event == ESP_SPP_SRV_OPEN_EVT) {
64     isBluetoothConnected = true;
65     SerialBT.println("Small Thermocouple (C) , Timestamp (seconds)"); // Send units when connected
66   } else if (event == ESP_SPP_CLOSE_EVT) {
67     isBluetoothConnected = false;
68   }
69 }
```

Android App

- ❖ Data is collected with the Serial Bluetooth Terminal App
- ❖ Temperature and Time is recorded.



Visual Studio Code Example

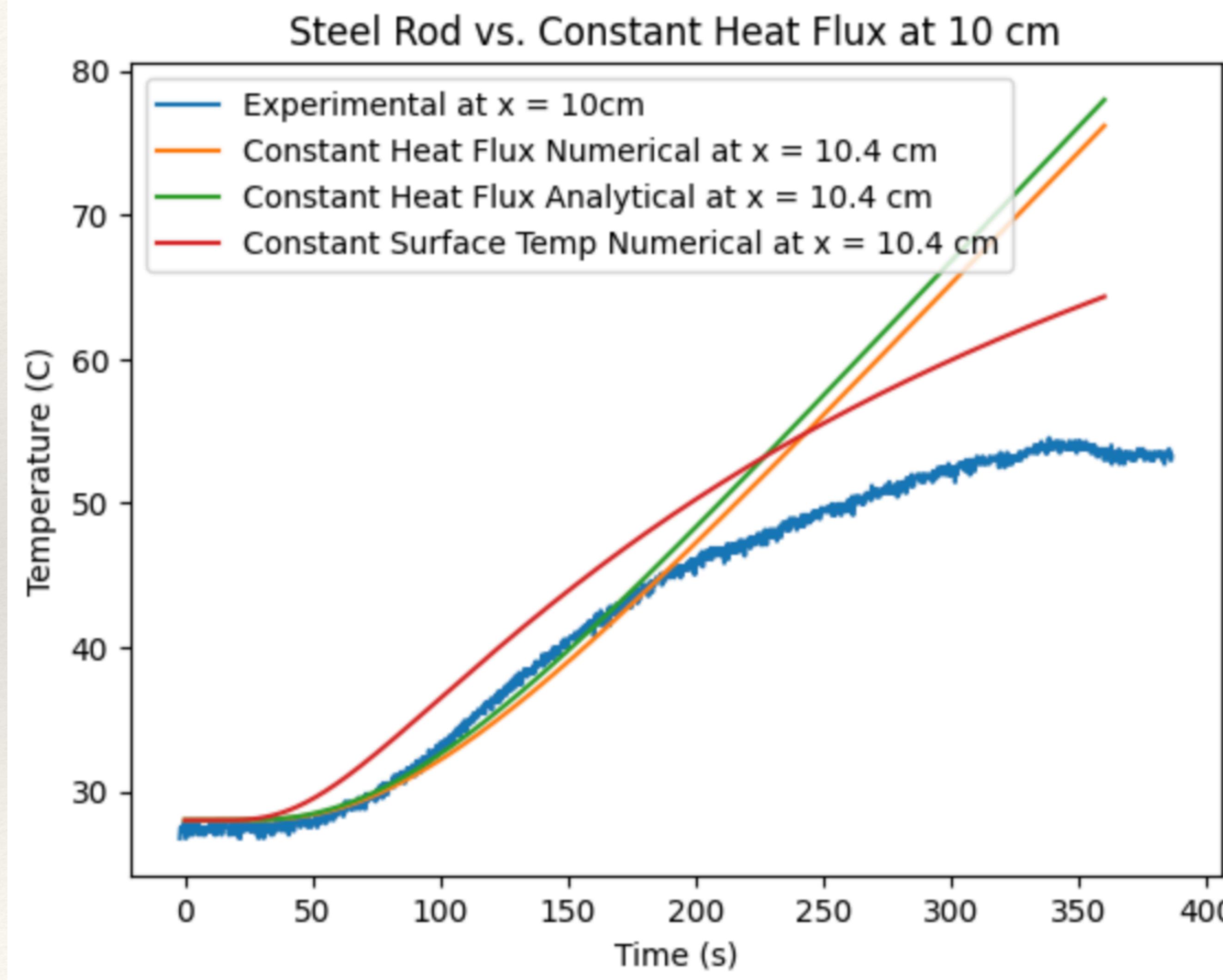
The screenshot shows a Visual Studio Code interface with a dark theme. The top bar displays file tabs: intro.md, _toc.yml, heattransfer.ipynb (marked as modified), casestudies.md, and heattransferkey.ipynb (marked as modified). The title bar also shows the search term "comptools". The left sidebar contains icons for file operations, search, and other extensions. A plot window titled "Temp" shows a graph of Temperature (s) versus Time (s), with two curves: an orange curve starting at (0,0) and a blue curve starting at approximately (200, 12). Below the plot, the text "Experimental results" is displayed. A descriptive paragraph explains the setup: "A rod is fitted with two or three thermocouples and safely heated at one end using a lighter. The rod is supported on a stand. The temperature at about 10 mm away is recorded with an ESP32 and the data is sent to a phone via bluetooth and then saved and plotted. Results are compared to the analytical and numerical solutions." At the bottom, a code editor shows a Python script:

```
#import file from folder
import pandas as pd
exdata = pd.read_csv('/Users/clintguymon/Downloads/steelheatingrod3.txt', header=3)[:-1].astype(float)
exdata.head()

[9] ✓ 0.2s
...
27.77 27.91
0 26.75 28.11
1 27.26 28.31
2 27.51 28.51
3 27.26 28.71
4 27.26 28.91
```

The status bar indicates the code is written in Python.

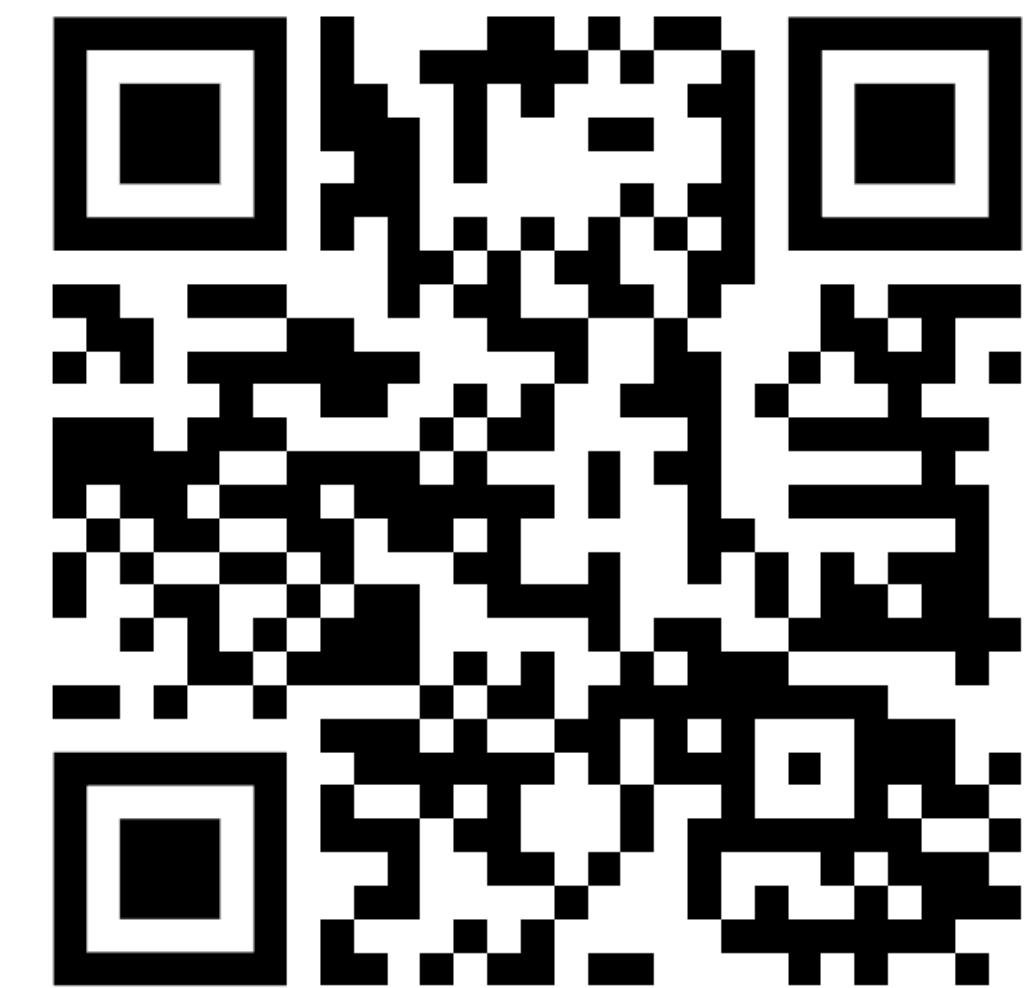
Experimental Data vs. Theory



Review: What I Hope You Got

- ❖ One application of math is to heat transfer which is used all around us
- ❖ You can describe heat transfer using an energy balance as an equation
- ❖ You can solve that equation many different ways:
 - ❖ Analytically, Numerically, Experimentally
 - ❖ There are very affordable options for you to collect data
- ❖ Your assumptions can affect your estimate (vs. real life)
- ❖ God lives and you are a child of God

This Presentation Location



<https://github.com/clint-bg/tools>