19 January 2025

Project 1 Visualize ODE with SciPy

PRESTON BROWNLEE & RYAN WHITE  
COLLEGE OF ENGINEERING & TECHNOLOGY Grand Canyon University

CST-305 PRINCIPLES OF MODELLING & SIMULATION

Professor Citro

| b. Responsibilities and completed tasks by each team member | …… | *1* |
| --- | --- | --- |
| c. System performance context description | **……** | *1* |
| d. Specific problem solved | **……** | *1* |
| e. The mathematical approach for solving it | **……** | *2* |
| f. The approach for implementation in code | **……** | *4* |
| g. Screenshots depicting key phases in the program execution | **……** | *5* |
| h. References for theory and code sources | **……** | *6* |

# Responsibilities and completed tasks by each team member

## Preston Brownlee

### **Documentation:** References for Newton’s Law of Cooling and the use of ODEs to calculate CPU temperature

### **Documentation:** System performance context description

### **Documentation & Code:** The mathematical approach for solving it

## Ryan White

### **Deliverables:** Documentation Template and file communications setup through Google Drive.

### **Documentation:** Specific problem solved basic on project description and context-related details

### **Documentation & Code:** The approach for implementation in code

# System performance context description

CPUs (Central Processing Units) often generate high temperatures, so cooling systems are implemented into computers to keep the temperature low and keep the hardware functional (Khan Academy).

Newton’s Law of Cooling provides a framework for understanding the way objects, in this case CPUs, heat up and cool down while taking their environment into account. This law shows us a differential equation that models the rate of temperature change as being proportional to the difference between the object's temperature and its surroundings (Aland, Mithapelli, Raheja, with JETIR). If we apply this principle to CPU cooling, the Law describes theoretical predictions of temperature changes over time under various conditions.

# Specific problem solved

The program develops the mathematical and computer programming solution to perform and visualize Newton’s Law of Cooling in theoretical temperatures. The program processes the ODE (Ordinary Differential Equation) of Newton’s Law of Cooling with regard to the initial conditions of CPU temperature and surrounding environmental temperature taken as input from the user (in Celsius).

The program creates a chart visualizing the way the temperature (in Celsius) of the CPU changes over time (in seconds) for the use of performance estimates and efficiency evaluation.

# The mathematical approach for solving it

**Newton’s Law of Cooling (Differential Form)**

: The rate of change of temperature over time (in °C/s).

: The cooling constant.

*T* : The temperature of the object at time *t* (in °C).

: The ambient temperature (in °C).

**Newton’s Law of Cooling (Analytical Solution)**

: The temperature of the CPU at time *t* (in °C).

: The ambient temperature (in °C).

: The initial temperature of the CPU at = 0(in °C).

: The cooling constant.

: Time.

: The exponential decay factor.

**Steps to Solve/Visualize**

1. Define Input Parameters:
   * Initial Temperature .
   * Ambient temperature .
   * Cooling constant .
     1. is dependent on the circumstances of the environment and therefore must be solved using a known output of the formula .
   * Time range over which to calculate.
2. Calculate the Temperature:
   * Use the formula to compute the temperature at time intervals.
3. Visualization:
   * Plot the temperature against time to visualize how the CPU heats up or cools down over time.

# The approach for implementation in code

The program implements individual functions for each step described in solving and visualizing the mathematics of the problem.

**Defining Input Parameters**

A new function requests the input for the initial conditions of the simulation, temperatureInitial, temperatureSurrounding,and timeRange.  
See “Solving for the Cooling Constant k” below for the value of k.

All input parameters are defined globally and can be accessed by functions of the whole program.

**Calculating the Temperature**

The SciPy function odeint takes input the differential equation , temperatureInitial, timeRange, and arguments to pass for temperatureSurrounding and k.

odeint returns the solution to the differential equation

**Visualization**

The final function uses matplotlib to plot and display the array temperatureReading over 0 to timeRange as a readable model of the simulated change in temperature.

**Solving for the Cooling Constant k.**

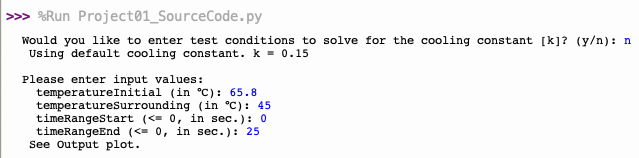
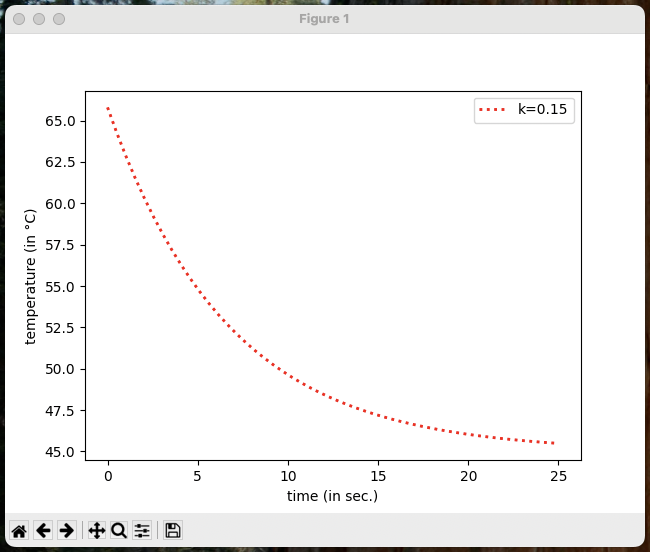
The user can optionally provide a set of input parameters to solve for the cooling constant.

If this is chosen, a function will take input solvingInitial, solvingTime, solvingReading, and temperatureSurrounding, and use the formula to calculate k.

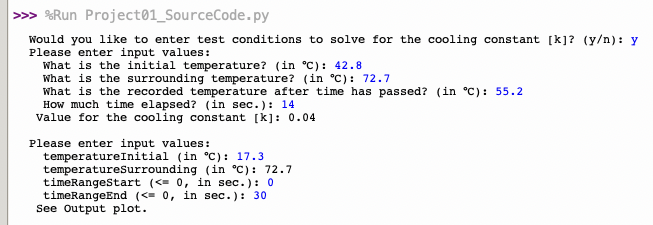
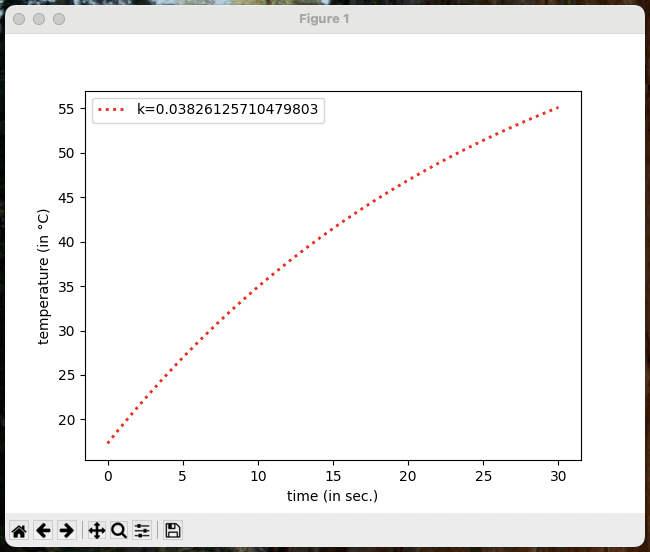
temperatureSurrounding will not be taken as input a second time when Defining Input Parameters and instead stored as a global variable with k. This is because the simulation should consider the simulation to be the same environment that the cooling constant was calculated in.

This function is to be ignored if the user declines solving for k, in which k will be based off of the study provided by JETIR and assigned the value 0.15.

# Screenshots depicting key phases in the program execution

*  
*

*The program uses the default cooling constant and creates a plot with data resembling JETIR’s study.*

*  
*

*The program solves for the cooling constant and models simulated conditions.*

# References for theory and code sources

[Application Of Newton's Law Of Cooling for a System](https://www.jetir.org/papers/JETIRBU06044.pdf):  
<https://www.jetir.org/papers/JETIRBU06044.pdf>

[Newton's Law of Cooling: Definition, Proof, Formulas, & Examples - GeeksforGeeks](https://www.geeksforgeeks.org/newtons-law-of-cooling):  
<https://www.geeksforgeeks.org/newtons-law-of-cooling>

[Newton's Law of Cooling | Differential equations (video) | Khan Academy: https://www.khanacademy.org/math/differential-equations/first-order-differential-equations/exponential-models-diff-eq/v/newtons-law-of-cooling#:~:text=Newton's%20law%20of%20cooling%20can,%E2%81%BB%E1](https://www.khanacademy.org/math/differential-equations/first-order-differential-equations/exponential-models-diff-eq/v/newtons-law-of-cooling#:~:text=Newton's%20law%20of%20cooling%20can,%E2%81%BB%E1)