Computational Project #2: Group 6

Who wrote it and who took what role?

Vidushi has written this report, Michelle and Snow each tackled one method of solving the problem at hand, and Shangcao completed a majority of coding with help from Reese, our team leader, who oversaw our project and helped out wherever needed.

What is it about?

For this project, we went with direction 2 and decided to investigate a first-order ODE by selecting Newton's Law of Cooling. We will explain a brief background on the law of cooling and its practical uses before showing an explicit and implicit way of solving the problem with initial condition settings.

What is the main point, conclusion, or story?

Let's explore *Newton's Law of Cooling* (NLC)! This equation helps us understand and estimate the rate at which heat is transferred between the body and its environment. NLC shows us that the difference between the body temperature and environment temperature directly affects the rate of heat transfer.

The form of the equation is generally given by:

$$\frac{dT}{dt} = k(M - T),\ k > 0$$

In this equation, T is the temperature of a body, M is the temperature of its environment, and t is the time. k represents a proportionality constant that is greater than 0.

What are our initial settings?

We will be using the following *Initial Value Problem* (IVP):

$$y' = 2(25 - y),\ y(0) = 40$$

What does this mean?

In our scenario, 25 ℃ is the temperature of the surrounding environment, and we are letting y be the temperature of the object. Our initial condition, , $y(0) = 40$ tells us that the initial temperature of the object begins at 40 ℃ at time 0.

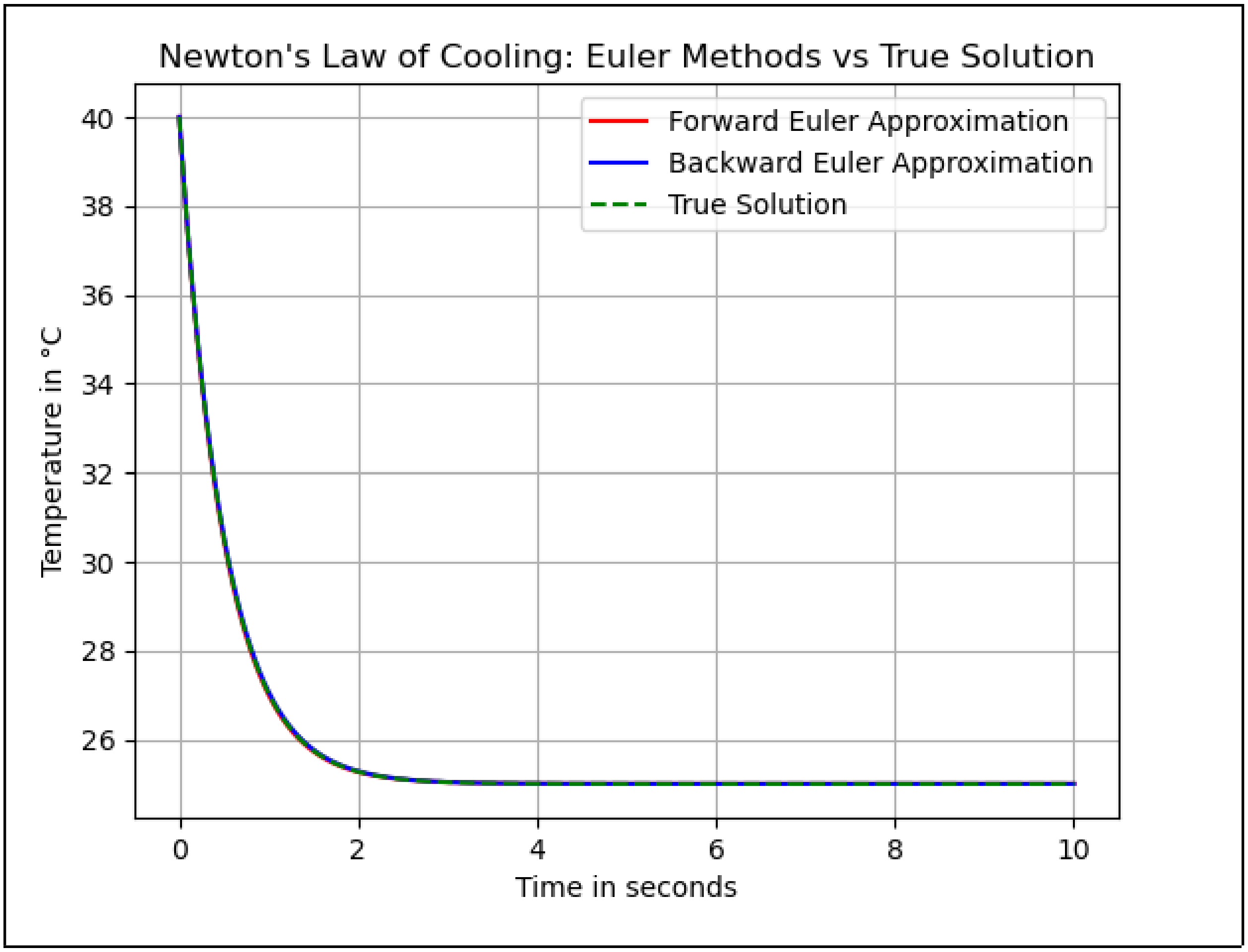
What methods were used to solve this problem?

We used one implicit and one explicit method to approach this scenario:

Implicit Euler:	Explicit Euler:
$y_1 = y_0 + hf(y_1)$ $y_1 = 40 + 10^{-3}(50 - 2y_1)$ $y_1 = \frac{6675}{167} \approx 39.97$ $y_2 = y_1 + hf(y_2)$ $y_2 = \frac{6675}{167} + 10^{-3}(50 - 2y_2)$ $y_2 = \frac{3341675}{83667} \approx 39.94$ $y_3 = y_2 + hf(y_3)$ $y_3 = \frac{3341675}{83667} + 10^{-3}(50 - 2y_3)$ $y_3 = \frac{1672929175}{41917167} \approx 39.91$ $y_4 = y_3 + hf(y_4)$ $y_4 = \frac{1672929175}{41917167} + 10^{-3}(50 - 2y_4)$ $y_4 = \frac{837512616675}{21000500667} \approx 39.88$ $\vdots$ $y_{n+1} = y_n + 10^{-3}(50 - 2y_{n+1})$	$y_1 = y_0 + hf(y_0)$ $y_1 = 40 + 10^{-3}(50 - 2y_0)$ $y_1 = 39.97$ $y_2 = y_1 + hf(y_1)$ $y_2 = 39.97 + 10^{-3}(50 - 2y_1)$ $y_2 = \frac{566799}{16700} \approx 39.94$ $y_3 = y_2 + hf(y_2)$ $y_3 = \frac{566799}{16700} + 10^{-3}(50 - 2y_2)$ $y_3 = 33.92217976 \approx 33.92$ $y_4 = y_3 + hf(y_3)$ $y_4 = 33.92217976 + 10^{-3}(50 - 2y_3)$ $y_4 = 33.9043354 \approx 33.90$ . . . $y_{n+1} = y_n + 10^{-3}(50 - 2y_n)$

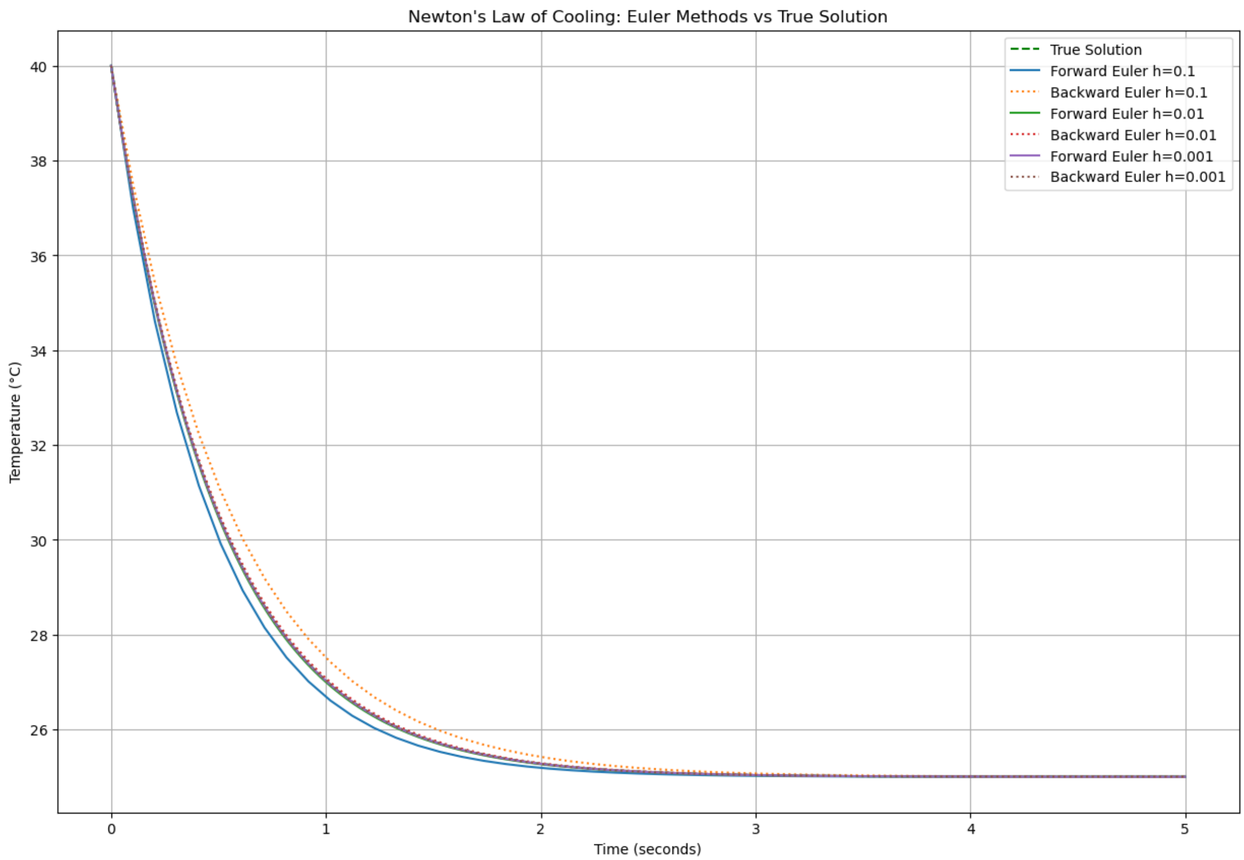
What is the evidence for the point or conclusion? Does it highlight the story?

Let's observe our calculations! We can use our graphs to highlight how our methods and calculations differed from the true result:



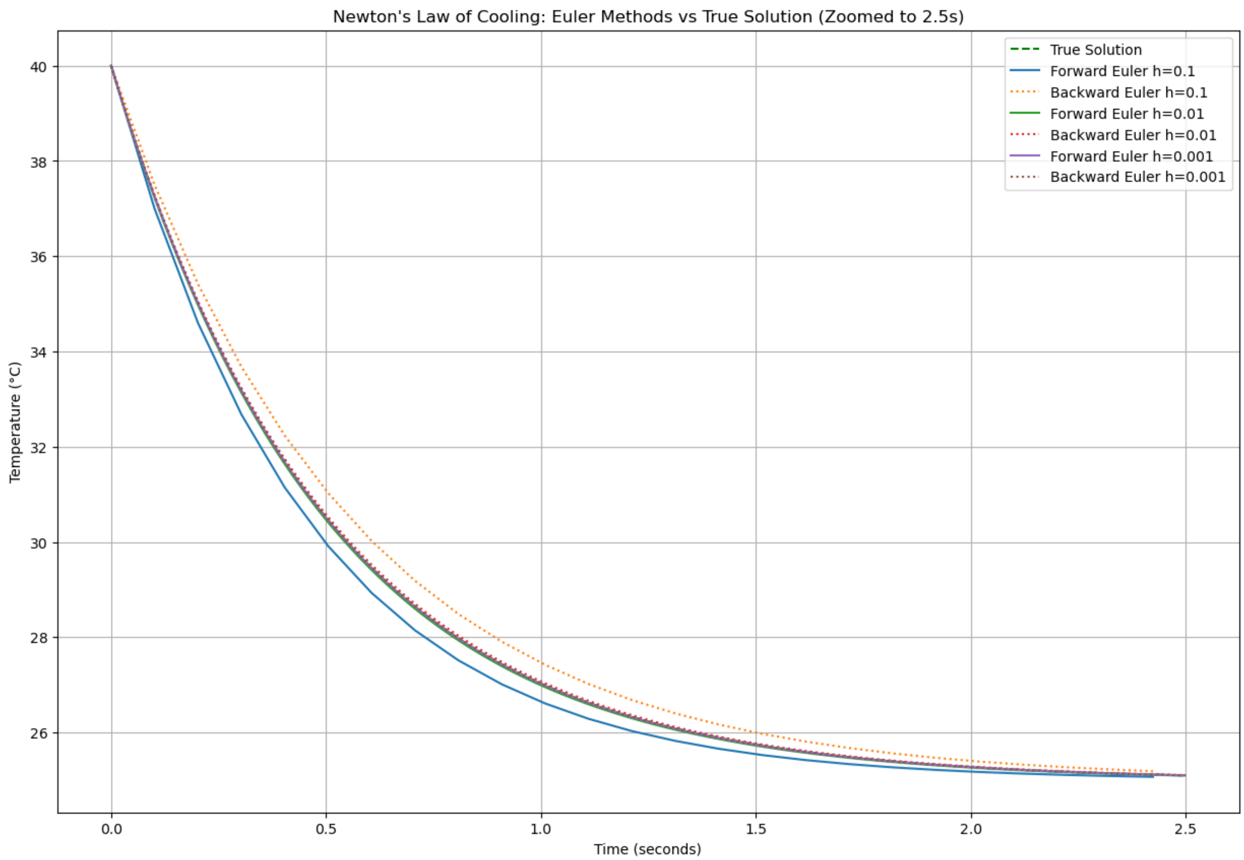
In this graph, we see how closely each method compares to the true solution and the other method. This specific simulation was taken with a step size of h = 0.1. In the scale of this graph, it is difficult to differentiate between the results from implicit euler, explicit euler, and the true solution. We can conclude that an implicit Euler method in our situation isn't necessary as long as our step size is adequate for our explicit Euler method.

We then plotted the *trajectories* for both the implicit method and explicit method using the time steps, h = 0.1, 0.01, 0.001. You can see this below:

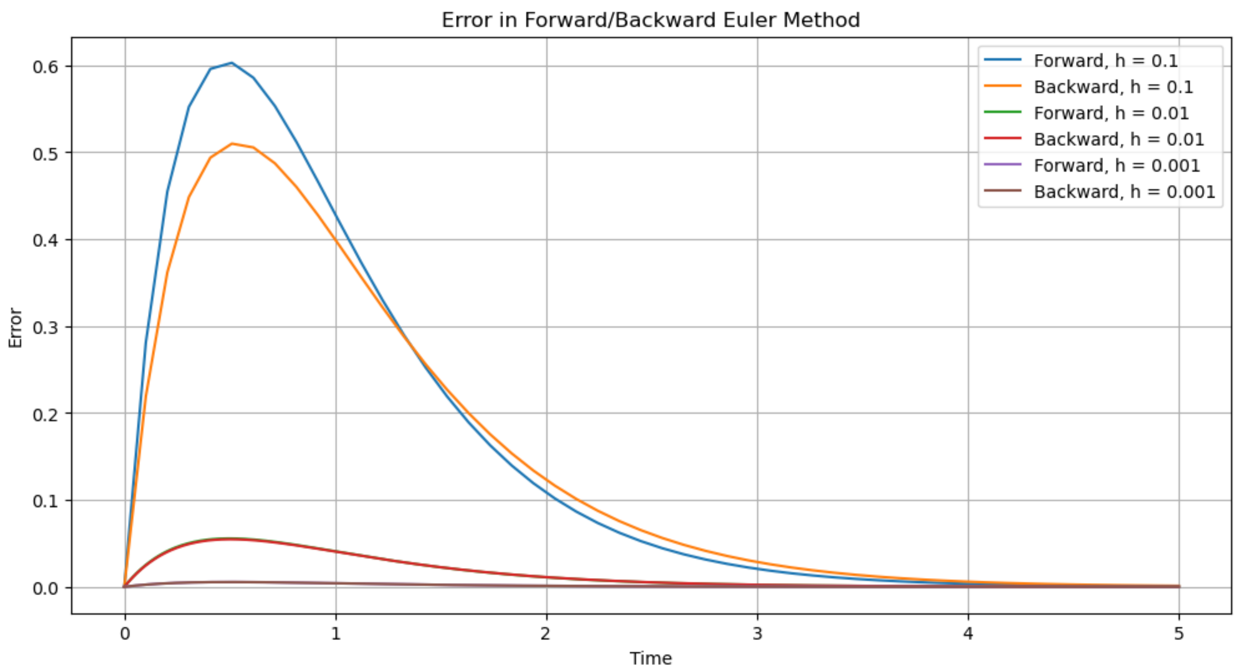


We can see from the graph above that decreasing our step size reduces the over or under fitting. Setting h = 0.1 produces an underfitting Forward Euler approximation because the approximation is under the true *trajectory*. For implicit Euler, we see the trajectory is overfitting when h = 0.1. Decreasing the step size to 0.01 or 0.001 produces an approximated *trajectory* that is nearly exact to the true solution.

Lastly, by taking a closer look at the approximation from t = 0 seconds , to t = 2.5 seconds, we can get a closer look at how each method behaves. This can be viewed in the following graph:



In the graph below, we were able to display the error between the true solution and for each time step. We note that this would be the *local truncation error* (LTE) since we are only calculating the error for each time step, rather than the accumulating error from previous steps added with the LTE, which would be considered the *global error*. We can see that from the times t = 0 seconds, to t = 2 seconds, the error from backward and forward Euler's method at the step size h = 0.1 was much larger than the other step sizes. The rest can be seen from the graph and we conclude that taking smaller step sizes does reduce error when the slope of the graph is high.



What is a limitation of what you did?

Of course, we must acknowledge that this problem and equation sets up a perfect scenario. While it can be used to estimate the true rate of heat transfer, many physical factors can affect how this scenario would play out in real life, which can be learned from the physics department.

Some key observations:

Notice that the simplicity of this problem setup allows us to see the accuracy of each method in the scenario of a 1D ODE. Our work on this equation leads to questions for future projects.... Would each method be as accurate in the context of a 2D or 3D ODE? How does the problem we chose affect the accuracy of our result?

Is it well organized?

Yes, we felt the simplicity of our chosen problem led us to be able to construct well-organized work, including our code, report, and graphs. Our methods cleanly show accuracy to the true solution and our arrangement of our work and graphs feels visually appealing.

Reflect on the whole process. What do you want to make note of to become a better programmer/presenter/documenter/leader?

I believe our group has greatly improved our efficiency since CP1. At first, we didn't know each other's strengths as well, and it took much more communication and effort to collaborate. With this project, we were able to split roles in a way that was better suited to each person's strengths and weaknesses, and complete our work in a smooth, timely manner. One important note is that starting with a simpler problem, expanding on the methods, and editing the code that implements the methods will only make it easier for us to work in higher dimensions. Thus, it was crucial for us to make sure the code was working properly for a simple 1D ODE like the NLC. For future work, we will be able to take these complex high-dimensional ODE's such as the pendulum, or maybe even in fluid dynamics. These good habits will eventually lead us to be able to work and solve problems in the world of PDE's.