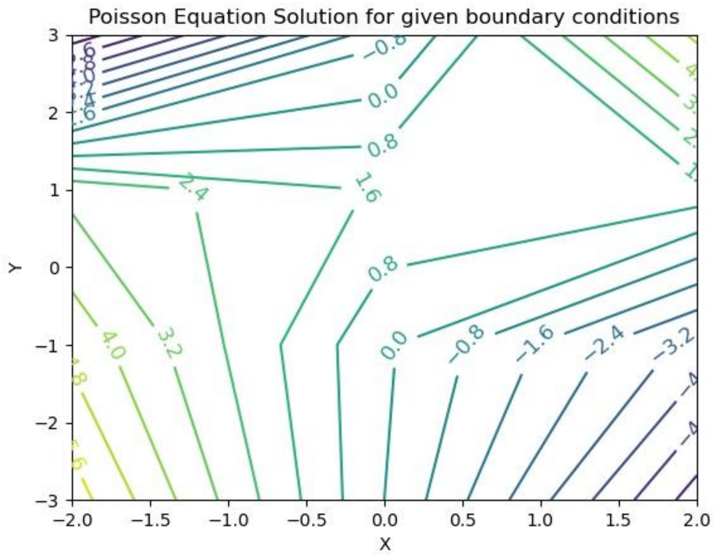
Lab 5 - Finite Differences - EngSci 331

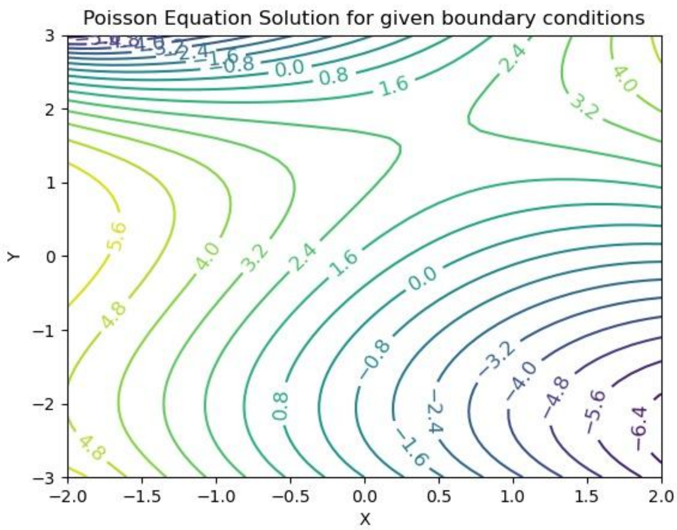
Daniel Clark - 343733502

1.1



Here is the result of the poisson finite difference solver when given a desired mesh spacing (delta) of 2. As we can see, it does not appear to be very convincing/accurate because there a lot of sharp, jagged edges and corners to our contours that we would not expect in a real solution

1.2



This result is a lot closer to what we would expect. This used a spacing (delta) of 0.1, which is much smaller and therefore provides much more accurate results. The results also appear ‘smoother’ which is an indication that they are displaying higher accuracy.

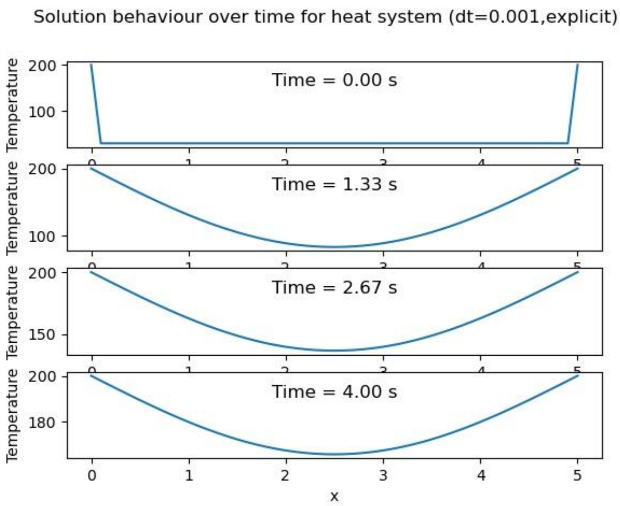
While the second option (delta=0.1) is computationally more expensive, it still runs fairly quickly (even on my old laptop), so is definitely worth the trade off because of the very large increase in accuracy as opposed to using the first option (delta=2).

2.1

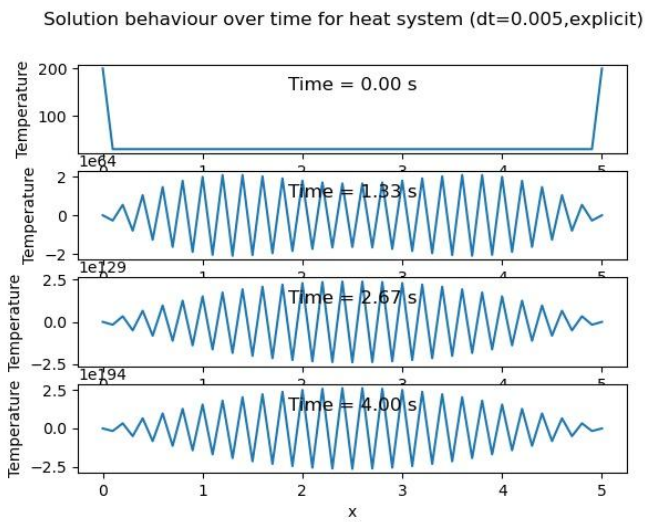
For this, I have used the equation $r = alpha*dt/dx^2$ to calculate the largest possible dt (timestep) that I can use which guarantees stability of our solutions. I have then set r less than or equal to 0.5, as this ensures stability. I have used dx=0.1 as stipulated in the project brief, and have used the alpha values related to each of the metals used in each of the three bars.

- For silver, alpha=1.5. $0 < r <= 0.5$ where $r = alpha*dt/dx^2$ gives dt<=0.003
- For copper, alpha=1.25. $0 < r <= 0.5$ where $r = alpha*dt/dx^2$ gives dt<=0.004
- For aluminium, alpha=1. $0 < r <= 0.5$ where $r = alpha*dt/dx^2$ gives dt<=0.005

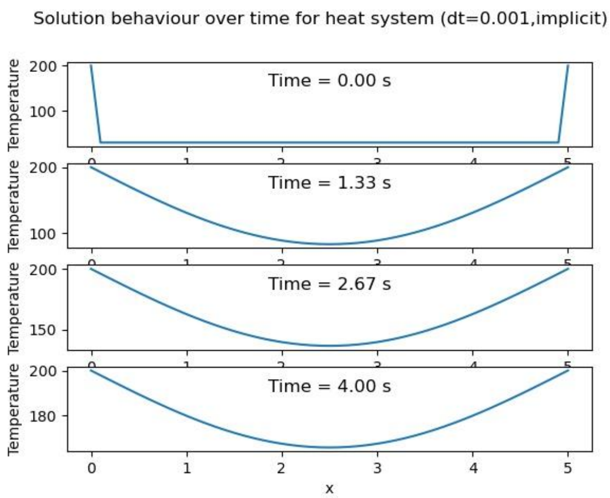
2.2

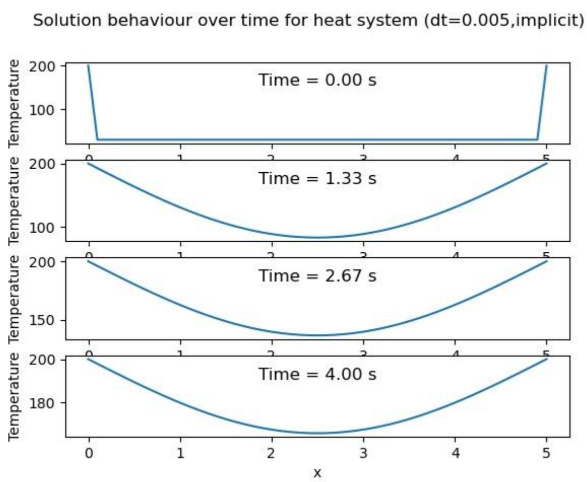


As expected, this result appears to be ‘stable’, because a timestep of 0.001 which is being used is below the 0.003 instability limit that was found in 2.1



Similarly, this result is UNSTABLE as expected, because the time step of 0.005 is > the instability limit of 0.003 that I found in 2.1. Because of this we can see the errors grow and compound over time.





This implicit solver is able to handle both the small timestep of 0.001 and the larger one of 0.005, as implicit solvers tend to be more stable than explicit solvers. This results in a larger possible maximum timestep before we start seeing unstable behaviour, and thus both of these outputs are stable.

2.3

For this task I decided to use the implicit solver and a timestep of 0.005 to ensure numerical stability but keep the computational cost fairly low. Here is the output from the code in my task 2:

```
The silver all reaches a temperature of at least 170 degrees in 4 seconds
The copper all reaches a temperature of at least 170 degrees in 4 seconds
the aluminium does not

Process finished with exit code 0
```

This shows us that both the silver and copper rods will be heated up to above 170 degrees across their whole width, while the aluminium will not transfer the heat as fast, and therefore will not reach 170 degrees in parts of the middle of the rod.