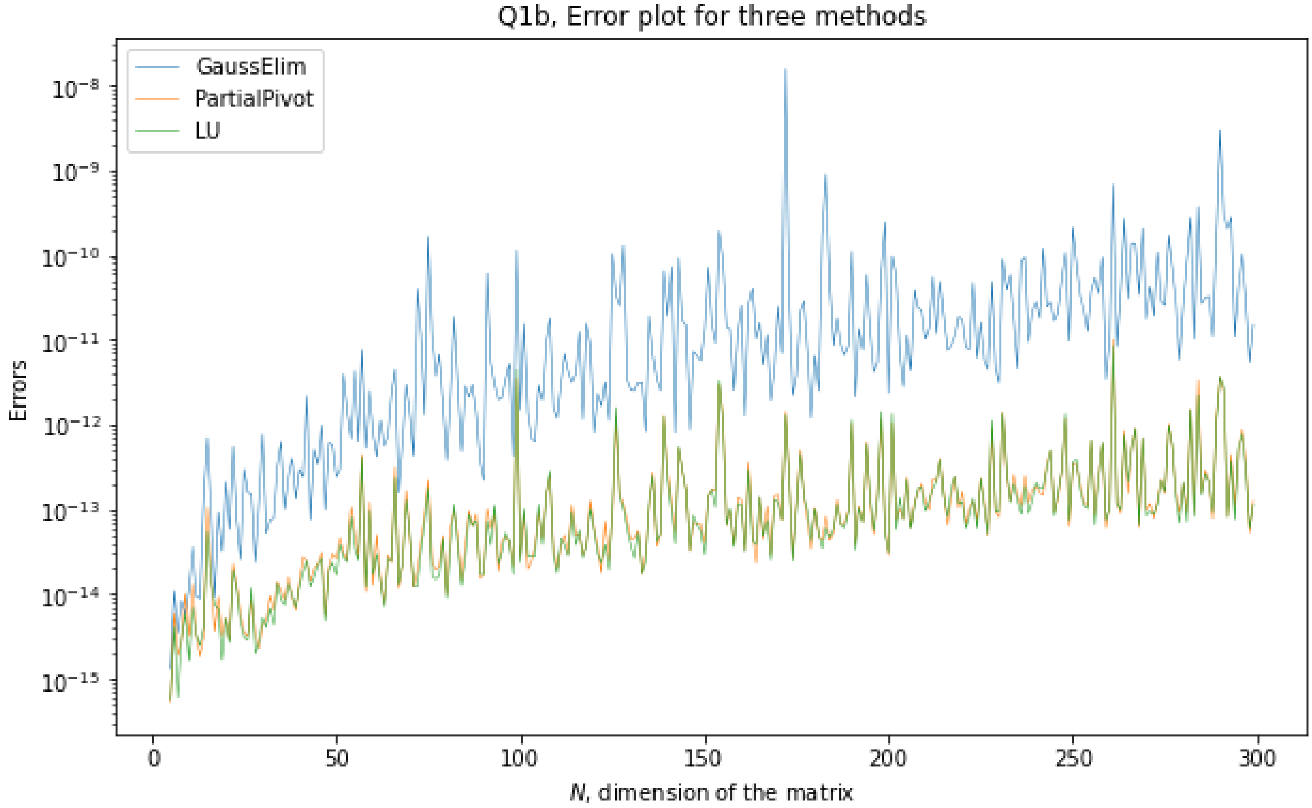
Lab4 Report

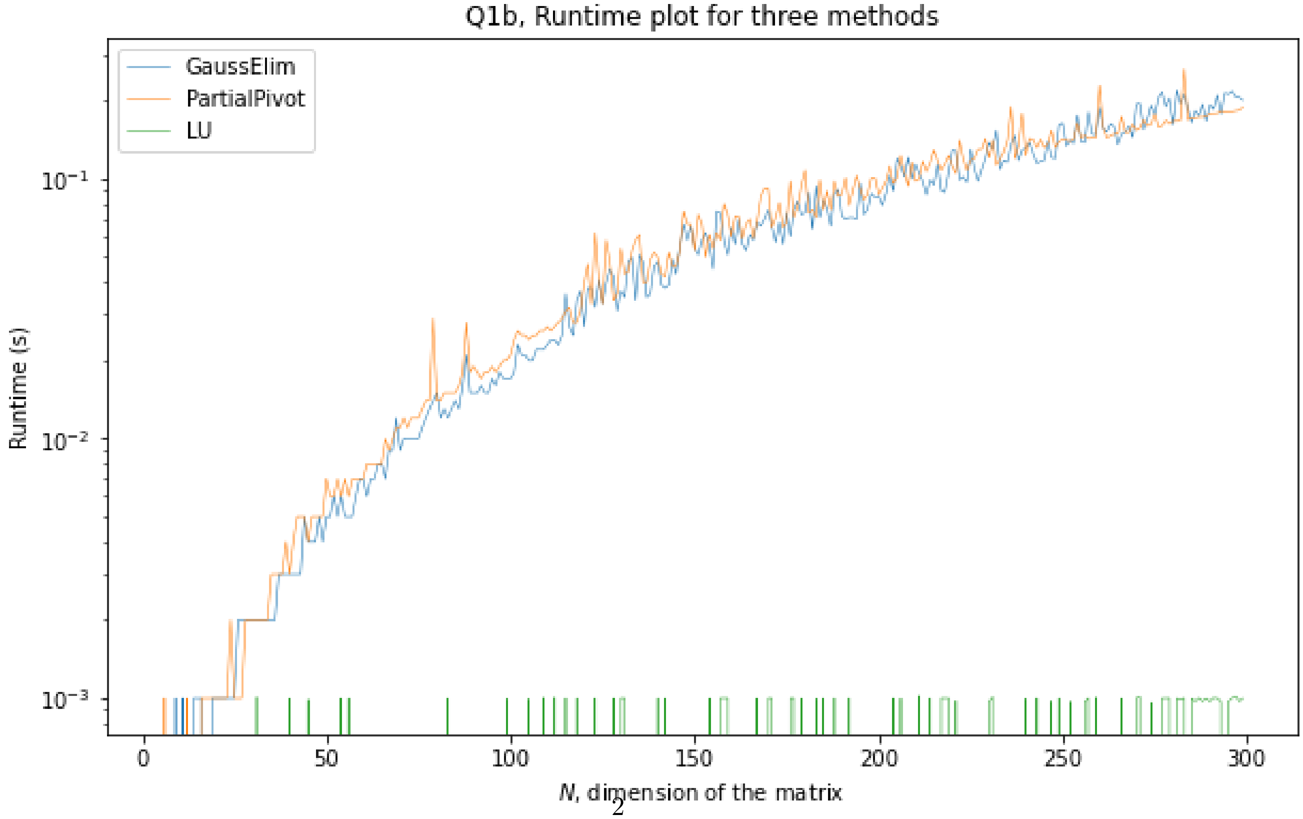
Zirui Wan (Question 2, 3), Rundong Zhou (Question 1)

October 9, 2020

1 Question 1

1.1 Q1b





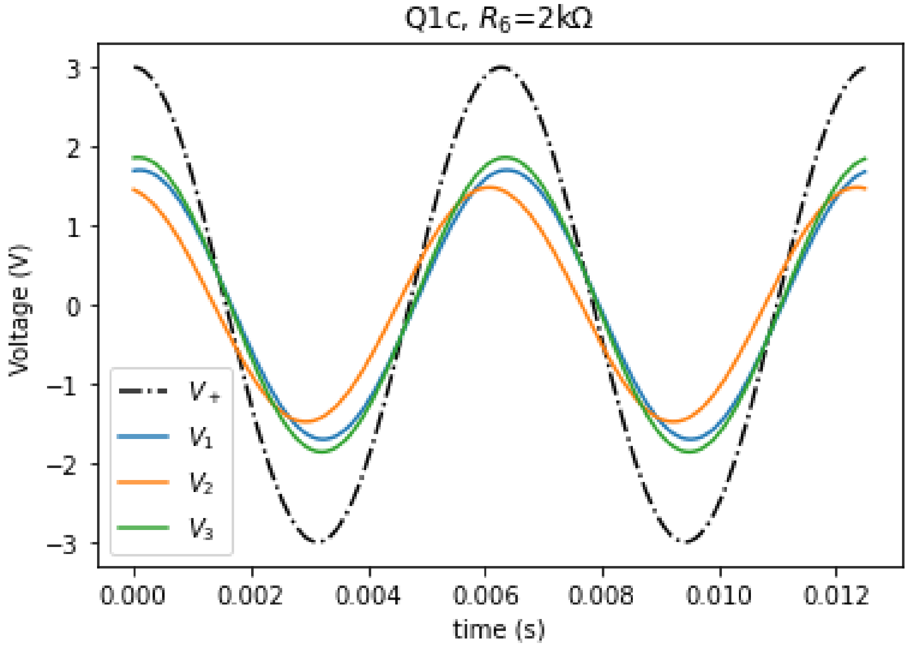
So I plot the the errors and timings for three methods on a log scale against the dimension N of the linear system up to $N = 300$.
Error wise, the partial pivot method provides a very close result to the LU method. The Gauss elimination method has greater errors, around 2 digits larger than LU and partial pivot. This is wired, since partial pivot uses the same algorithm as Gauss elimination, but it has the same magnitude of errors with LU method. So I googled a little bit, some articles say that partial pivot reduces rounding errors since it avoids dividing by small values.
Time wise, Gauss elimination and partial pivot have very close results. They both grow linearly with N. The LU method however, has a very low run time and does not show a trend with growing N up to $N = 300$. Since it is a built in function of the `NumPy` library, it must have some advanced algorithm to reduce the run time.

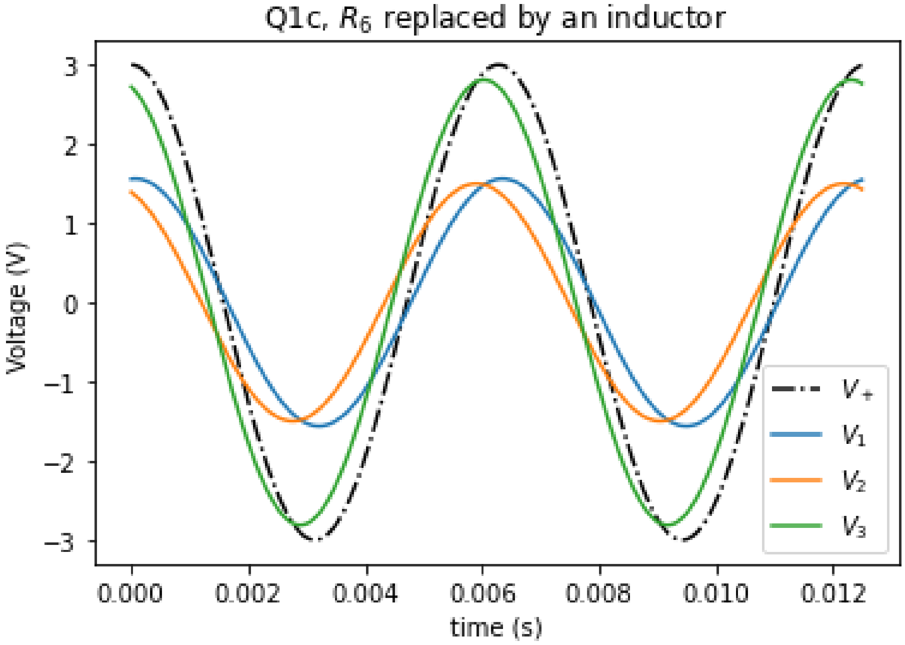
1.2 Q1c

The printed output is the following:

```
----------- R6 = 2k Om -------------------
The amplitude for V1 is 1.7014390658777336 V. The phase is -5.469094970111944 degree
The amplitude for V2 is 1.4806053465364062 V. The phase is 11.583418604687065 degree
The amplitude for V3 is 1.8607693200562132 V. The phase is -4.164672651865924 degree
----------- R6 replaced by an inductor -----------------
The amplitude for V1 is 1.5621181940219633 V. The phase is -4.025908819603362 degree
The amplitude for V2 is 1.4994286802306562 V. The phase is 21.63928264257023 degree
The amplitude for V3 is 2.8112763903392537 V. The phase is 14.352479528588603 degree
```

The following are the plots of three voltages as well as the reference voltage for two cases:





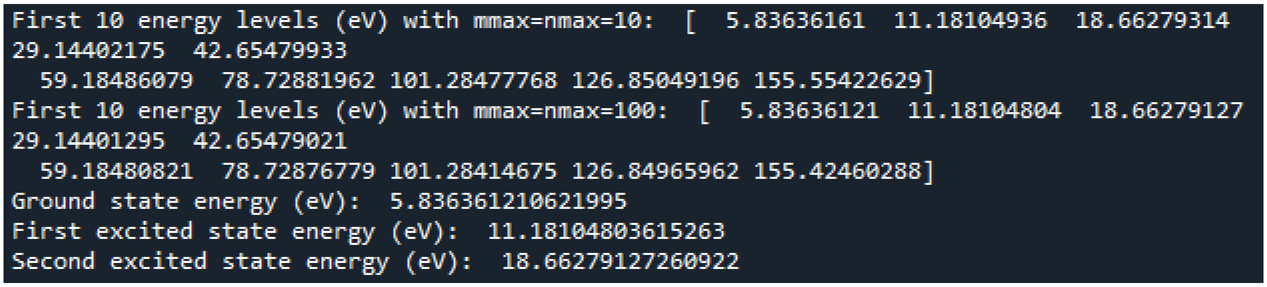
- The most significant difference between the two plots is, the amplitude for V_3 is raised from $1.86V$ to $2.81V$. This makes quite sense, since an inductor acts like a short circuit for a constant voltage ($V = V_0$), and an open circuit for a very high frequency ($V = V_0e^{i\omega t}, \omega \to \infty$) voltage. In our case, $\omega = 1000$, which is considerably high, thus, V_3 gains in amplitude due to the inductor.

- The phase of V_3 also changes a bit. It shifts about 28 degree to the left. This is due to the change in imaginary impedance.

- V_1 and V_2 mostly remain unchanged. V_2 shifts about 10 degree to the left.

2 Question 2

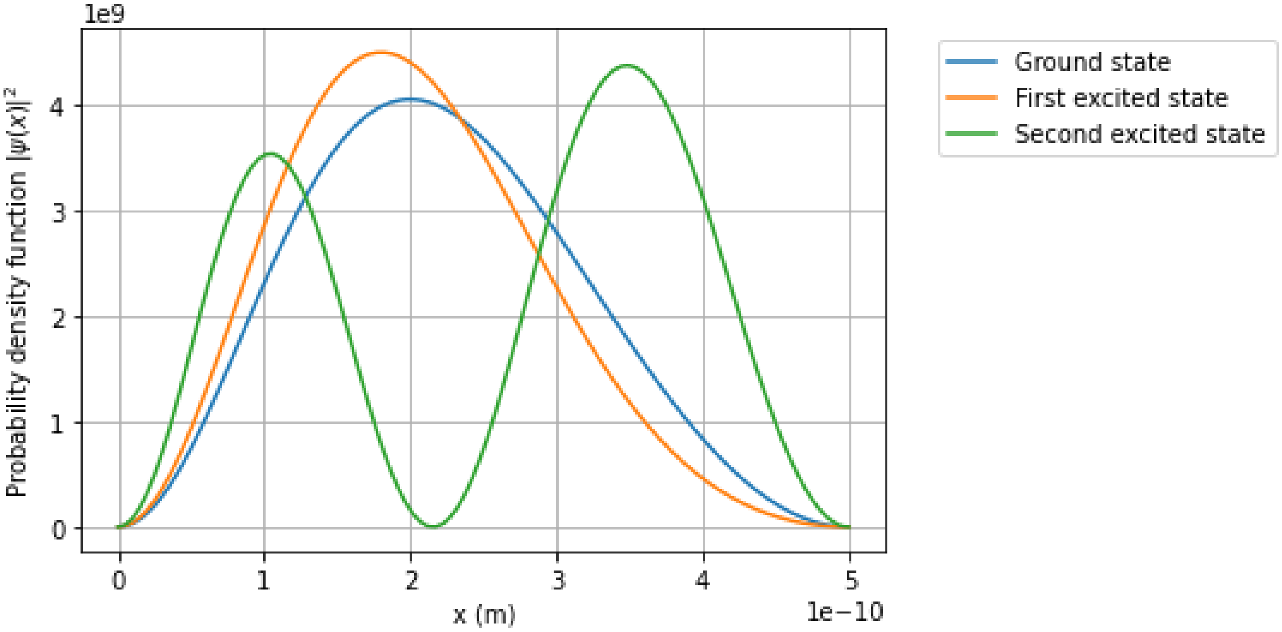
2.1 Part (c)(d) Output



```
First 10 energy levels (eV) with mmax=nmax=10:  [  5.83636161  11.18104936  18.66279314
29.14402175  42.65479933
  59.18486079  78.72881962 101.28477768 126.85049196 155.55422629]
First 10 energy levels (eV) with mmax=nmax=100:  [  5.83636121  11.18104804  18.66279127
29.14401295  42.65479021
  59.18480821  78.72876779 101.28414675 126.84965962 155.42460288]
Ground state energy (eV):  5.836361210621995
First excited state energy (eV):  11.18104803615263
Second excited state energy (eV):  18.66279127260922
```

The calculated first 10 eigenvalues of the Hamiltonian matrix are shown above, for matrices with size of 10 elements and 100 elements respectively. You can see that the eigenvalues for two matrices are slightly different, although they correspond to the same problem. The differences are very small though. For example, the difference between the two eigenvalues corresponding to the ground state is $4 \times 10^{-7} eV$, which is just about a relative error of 1×10^{-7}. Thus, both sets of eigenvalues are accurate from each other.

2.2 Part (e) plot



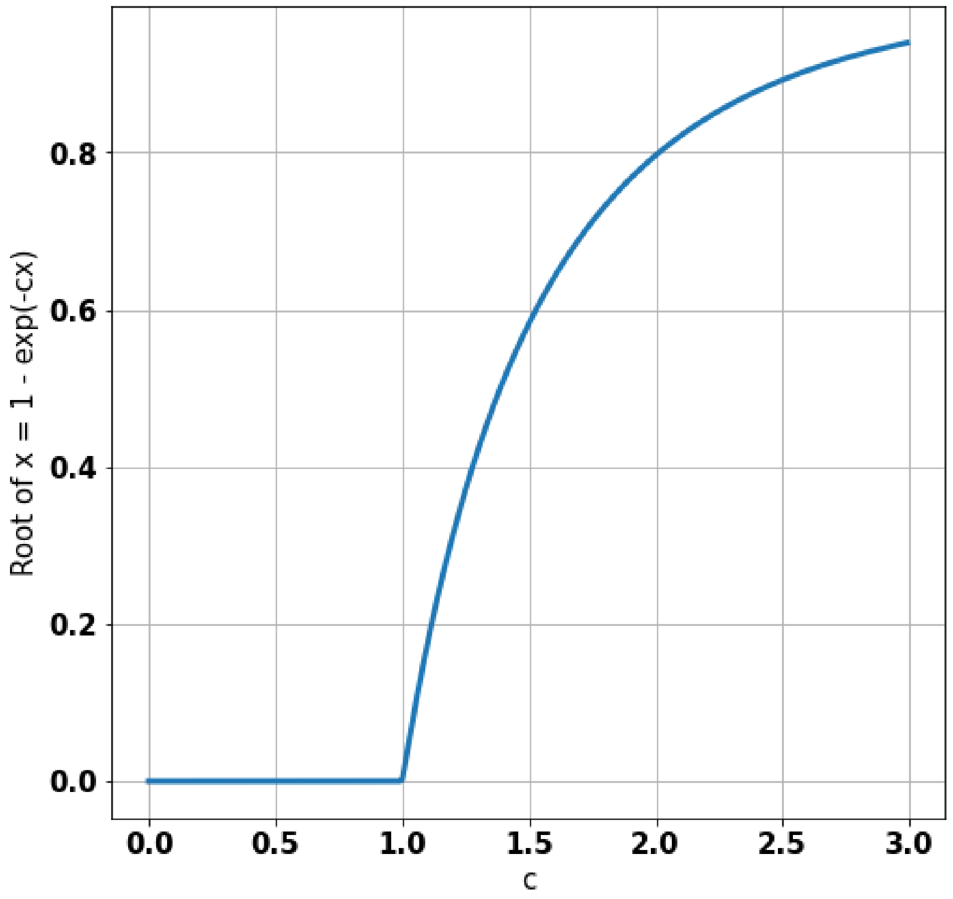
2.3 Explanatory Notes

The probability density functions for ground state and the first two excited states are plotted above. You can see that the pdf's are not symmetric about the middle point of the potential well, as expected. The reason for that is because the potential inside this well is a linear line, so that the probability of the particle to be appear will be different from location to location. For example, a particle with energy level at the second excited state will be most likely to appear at two locations, $x_1 = 1$Å and $x_2 = 3.5$Å. The probability for the particle to appear at x_2 is slightly higher than x_1.

3 Question 3

3.1 Part (a)

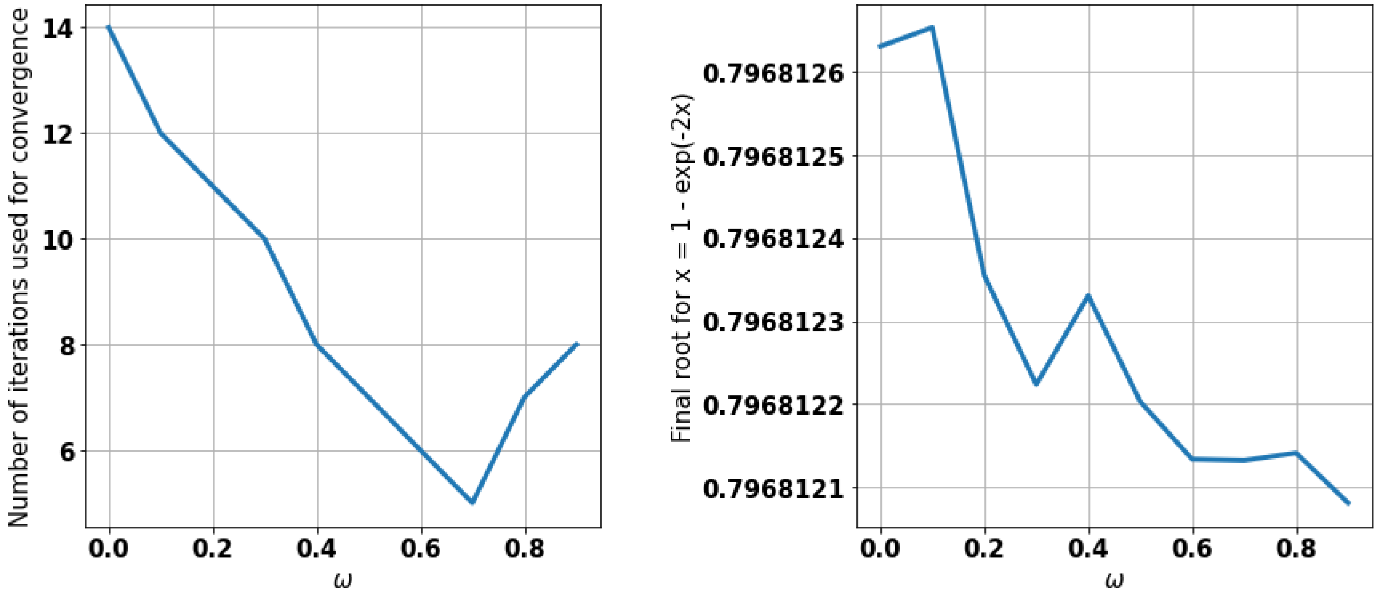
3.1.1 Plot



The plot for roots of the equation $x = 1 - e^{-cx}$ is shown above. The transition from zero roots to non-zero roots takes place at $c = 1.0$.

3.2 part (b)(c)

3.2.1 Plot



3.2.2 Discussion

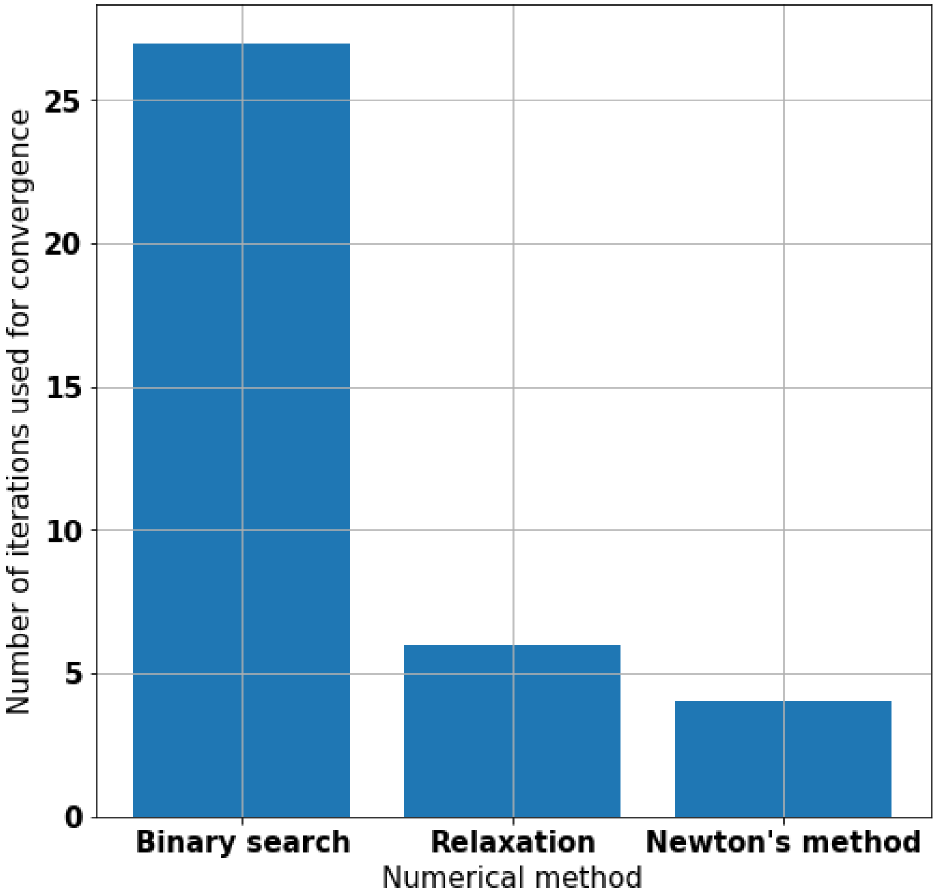
In order the investigate the impact of the overrelaxation rate ω on the speed of convergence and the accuracy of the final solution. I plotted number of iterations used for convergence versus overrelaxation rate ω on left above. On right shows the final solution of the root versus overrelaxation rate ω. You can see that as the overrelaxation rate ω increases from 0 (corresponding to original relaxation method without overrelaxation) to around 0.7, the number of iterations before full convergence decreases from 14 to 4. Namely, the speed of convergence increases by about 70%. On the other hand, the solution vs. ω shows that, although the speed increases significantly, the accuracy of the final solution is well preserved. The relative differences among the final solutions are just on the order of 1×10^{-6}.

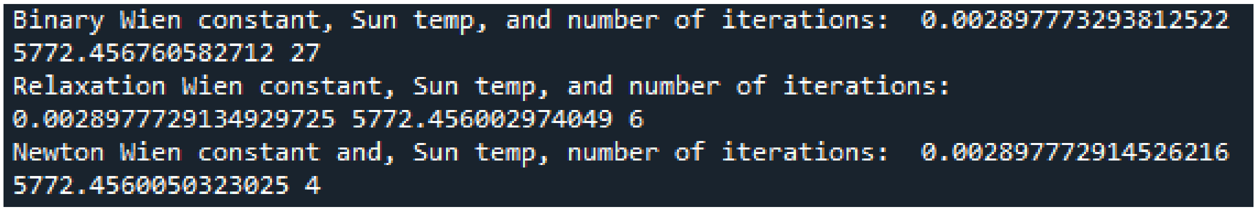
3.2.3 part (d) short answer

I think there is. Negative ω essentially means that, instead of taking larger steps, the algorithm now takes finer steps in searching for the final solution. This could be very helpful in speeding up the convergence, when the function has very high non-linearity. For such problems, large steps could lead to "overshoot" of the true solution, which will take very long time to finally converge as the numerical solution will be jumping around the true solution.

3.3 Part (c)

3.3.1 Plot





```
Binary Wien constant, Sun temp, and number of iterations:  0.002897773293812522
5772.456760582712 27
Relaxation Wien constant, Sun temp, and number of iterations:
0.0028977729134929725 5772.456002974049 6
Newton Wien constant and, Sun temp, number of iterations:  0.002897772914526216
5772.4560050323025 4
```

3.3.2 Explanatory note

Above I plot the comparison of number of iterations used before convergence for the 3 different methods. The binary search method takes more than 25 iterations to converge, which is the slowest among the 3 methods. Relaxation method and Newton's method have similar performance, taking 6 and 4 iterations respectively. The calculated Wien displacement constants and estimated surface temperatures of the Sun outputted from my code are also shown above. You can see the relative differences between the surface temperatures are just on the order of 1×10^{-7}. Thus, all 3 methods have almost the same performance in terms of accuracy, but the binary search method is the slowest one. Relaxation method and Newton's method have similar speed. However, the relaxation method might be the more practical choice. Because the Newton's method requires the derivation of analytical form of the derivative of the function. But sometimes the derivative of the function does not have an analytical form.