PHY407 Lab 01

Sang Bum Yi, 1004597714

Jianbang Lin, 1004970720

The workload was distributed as followings:

- Sang Bum Yi did question 3
- Jianbang Lin did questions 1 and 2.

Q1

Three methods are commonly used for solving a matrix; Gaussian elimination, particle pivoting, and LU decomposition. These three methods are mathematically identical, but they have different run-time and error when implemented in python. This question explores run-time and error with these three different methods.

Part b

The following graph shows the change in run time with respect to change in matrix size:

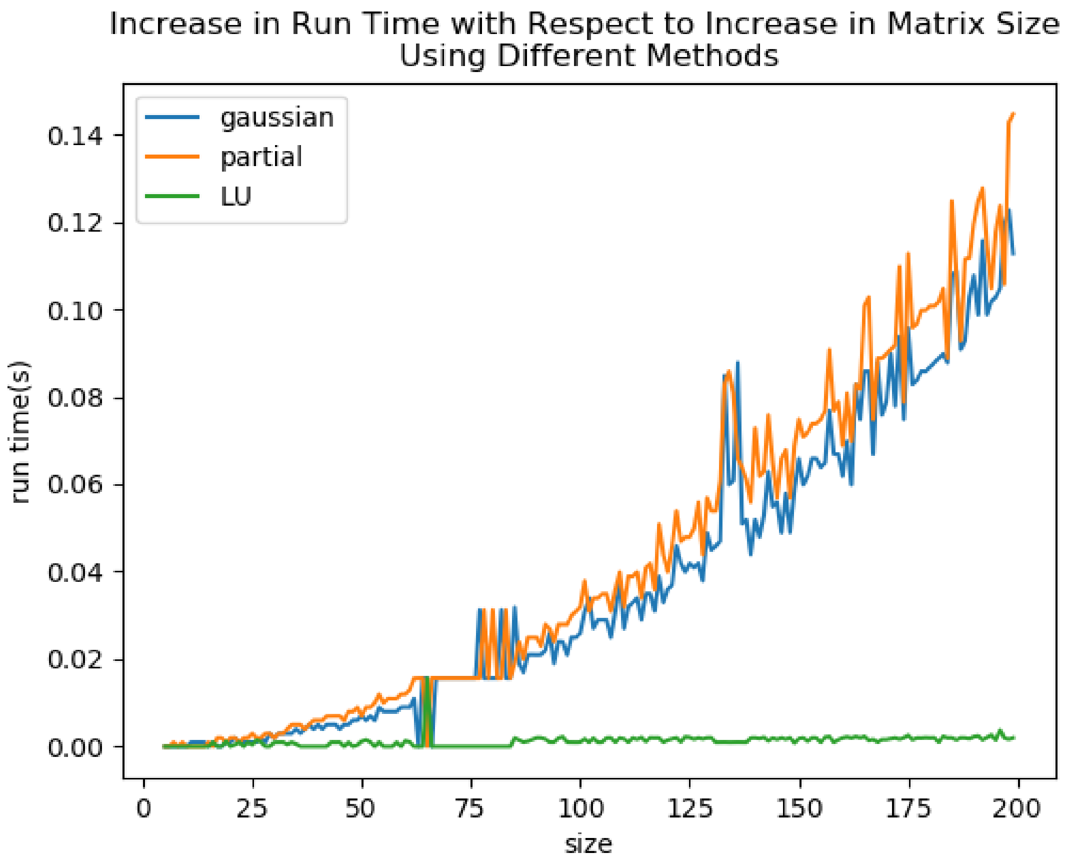


Figure 1: Increase in run-time with respect to increase in matrix size for three different methods; gaussian elimination, partial pivoting, and LU decomposition.

This graph shows that run time for LU composition remains close to constant with increase in matrix size. On the other hand, Gaussian elimination and partial pivoting methods exhibit similar run time, their run time increases linearly with time. Noticeably, the run time fluctuation for LU decomposition is small, but fluctuation for Gaussian elimination and partial pivoting methods are relatively large.

The error made by each methods is shown as following:

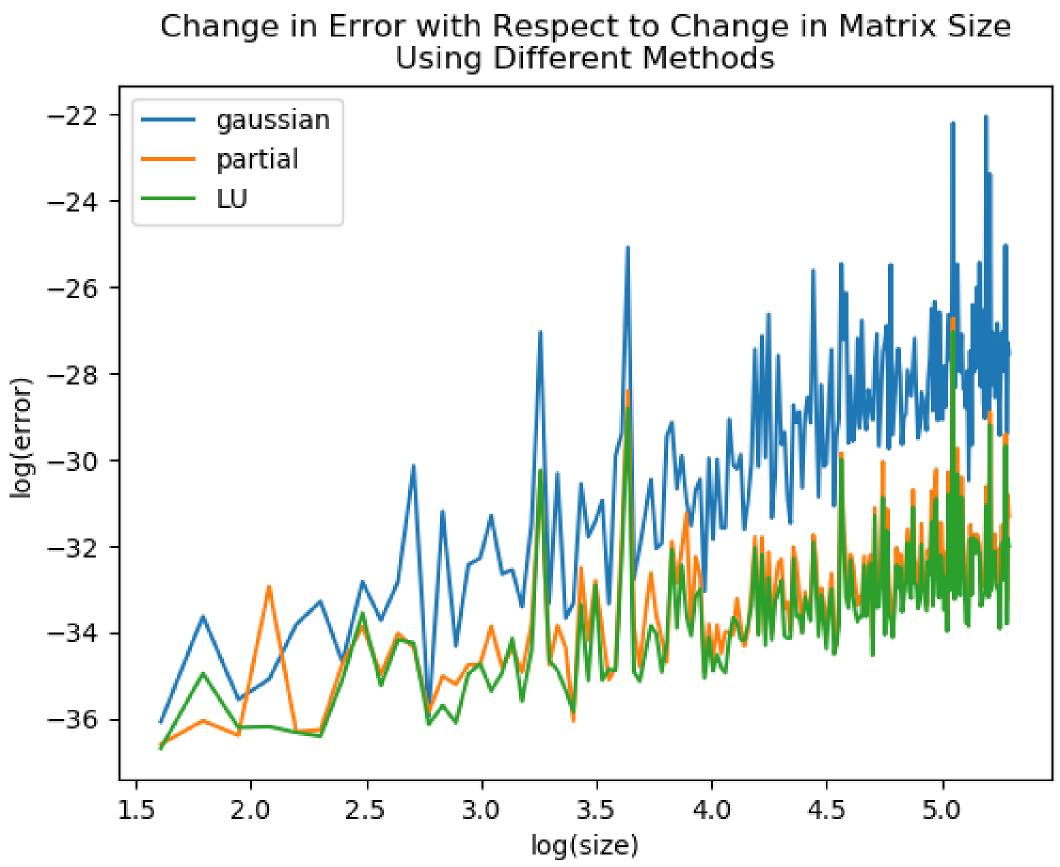


Figure 2: Log(error) vs log(size) with methods; gaussian elimination, partial pivoting and LU decomposition of matrices from size 5 to 200

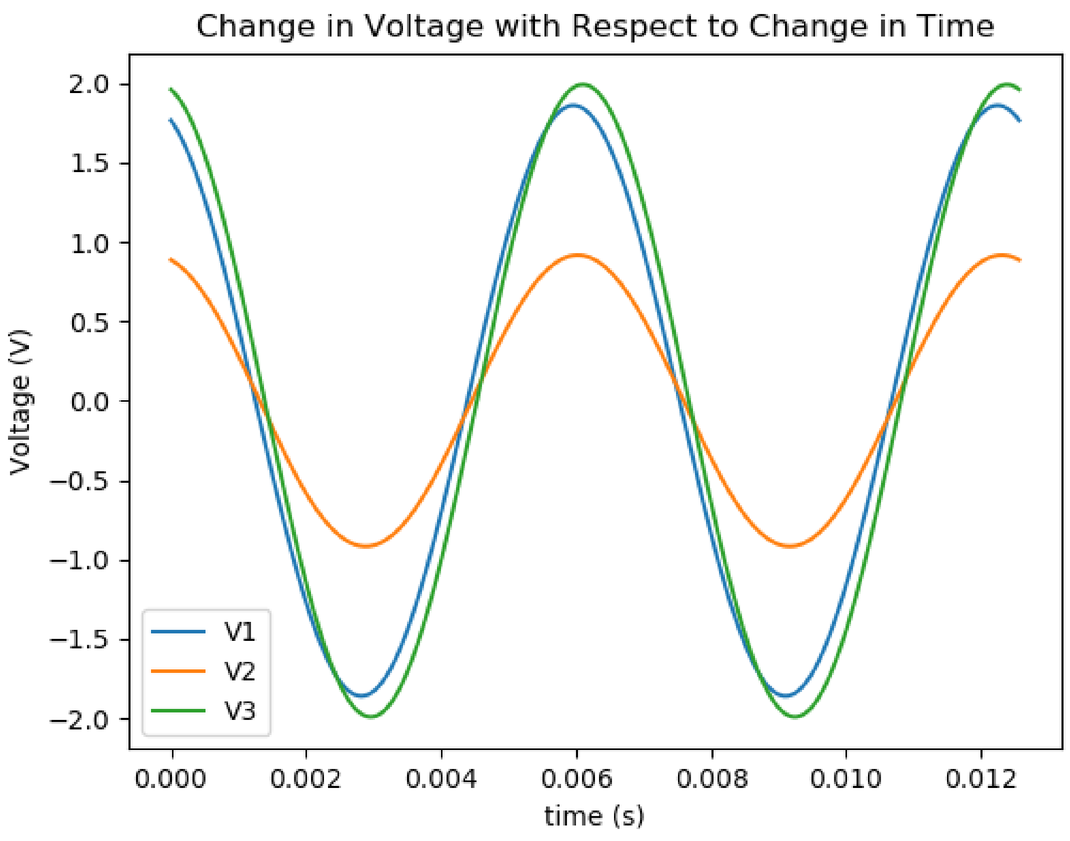
The matrix size vs error is plotted in a logarithmic scale, and this graph shows there exists a linear relation between log of error and log of size, therefore the relation between error and size is exponential. Noticeably, the error for partial pivoting and LU decomposition overlapped in many parts, it shows these two methods produce very similar results in python.

Based on the graphs of run time and error, it is obvious that LU decomposition is the best method to use in python when solving a matrix, it takes least time to run and produces least error.

Part c

Using matrix solving methods above, a complex circuit problem can be solved. The following is the graph of the voltage vs time graph in circuit mentioned in textbook problem (6.5):

Figure 3: Voltage vs time of for V1, V2 and V3 of two oscillations period.



This figure shows the change in voltage in two oscillations. They differ in amplitude and starting position. But the period is the same.

Q2

The question investigates the behaviour of a wave function in an infinite square well with non-constant potential across the well.

Part c

The state energy can be found from the eigenvalue of Hamiltonian function. However, the Hamiltonian function is infinitely large in theory, so this program can only approximate the Hamiltonian function. We first approximate it by taking the matrix size to be 10 by 10, and the ground state energy is 5.83354094 eV.

Part d
For a better Hamiltonian function approximation, the matrix size increases to 100 by 100, and the first ten energy levels of this system are; 5.83354054 eV, 11.1730178 eV, 18.64463974eV, 29.11155717eV, 42.60398334eV, 59.11159469eV, 78.62908518eV, 101.15392918eV, 126.6848395eV, 155.22111166eV.

The ground state energy for Hamiltonian matrices of size 10 and 100 are every closed, so, matrix size of 100 can be considered a very closed approximation of the true value.

Part e

The ground state, and first two excited states of wave function is shown as following:

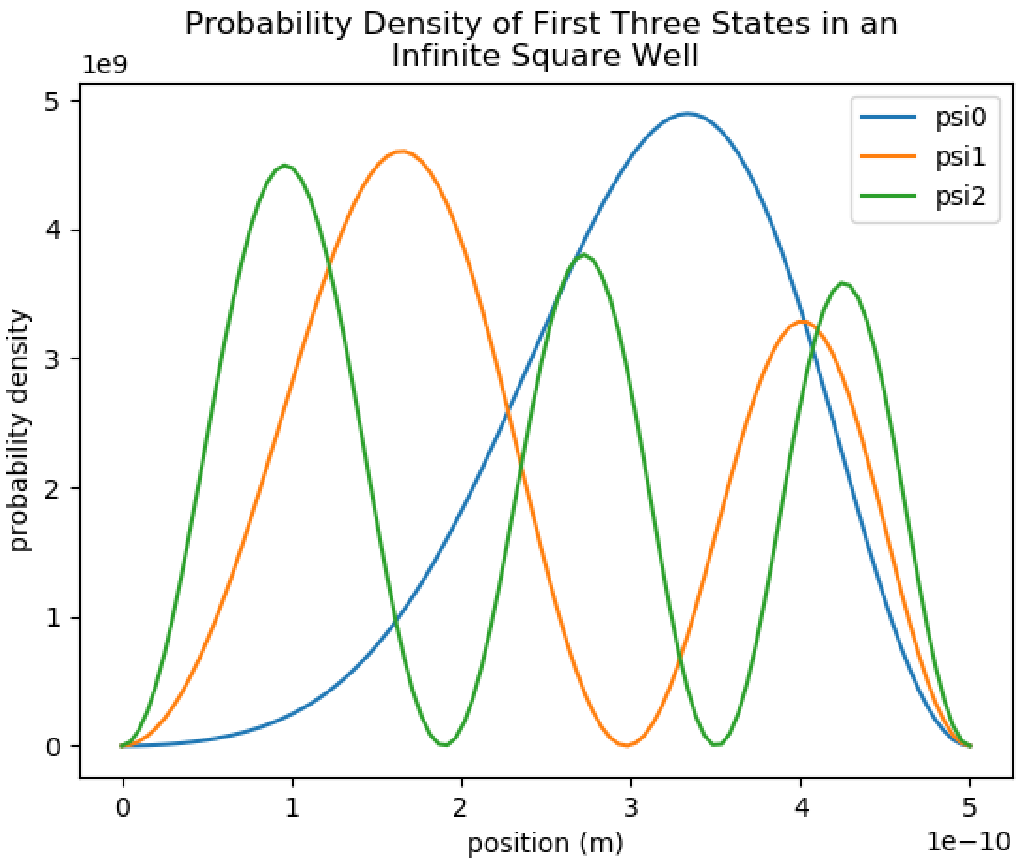


Figure 4: Probability density function of first three states in an infinite square well with varied potential inside the well.

In the traditional infinite square well problem, the probability density exhibits the property of symmetry in the middle of the well. However, this graph shows that when the potential is not constant across the well, probability density is not symmetrical. Where ground state probability density is tilted to right size and maxima of excited state probability density decreases from left to right.
One of the properties of probability density is that the integral should equal 1, but in this case, the integral for psi 1, psi 2 and psi 3 are 3, 113 and 239. They are not 1, that is due to the rounding error made by python.

Q3

Part a - Exercise 6.10 parts (a,b)

The relaxation method, which is an recursive algorithm used to obtain the converging value of a non-linear equation of a single variable that cannot be easily solved, was used to find the values of ‘x’ that satisfies the equation 1 below.

$$x = 1 - e^{-cx} \qquad \text{[Eq 1]}$$

where c is a known parameter and x is an unknown parameter.

With the initial guess of ‘x’ and the value of ‘c’ set to 1.0 and 2.0 respectively, the values of ‘x’ converged to 0.7968121300200202. Note that this value is presented with more than three significant figures to show the exact value when converged. The raw output of the iterations can be found in Appendix A.

In addition, the values of ‘x’ obtained by ranging the values of c from 0 to 3 in steps of 0.01 were plotted in Figure 1.

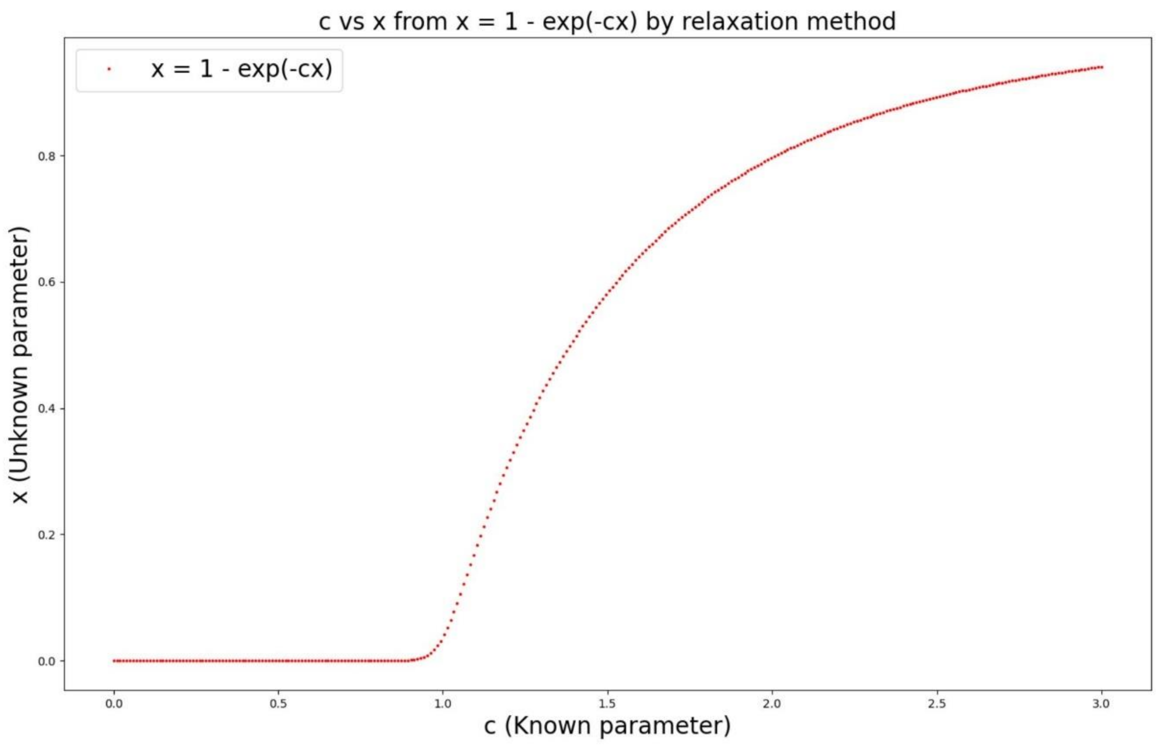


Figure 5: The values of 'c' versus the values of 'x' in Equation 1. The values of 'c' were set to range from 0 to 3.0 in steps of 0.01. The initial guess of 'x' was 1.0.

As specified in the exercise 6.10 part b, the phase transition that distinguishes a regime in which x = 0 to a regime of nonzero x was observed, near c = 1.0. This type of phase transition in Figure 1 is typically called "percolation transition" in physics.

Part b – Exercise 6.11 parts (b, c, d)

An overrelaxation method is a relaxation method that introduces an additional parameter $'w'$ such that $x' = x + (1+w)\Delta x$, where x' = f(x) and Δx is the difference between the original value and the value in the next iteration. The overrelaxation method is, in some cases, capable of solving a non-linear equation in fewer steps than the original relaxation method. Rearranging the expression with the overrelaxation parameter, the formulas for the overrelaxation method are given by:

$$x' = (1+w)f(x) - wx \qquad \text{[Eq 2]}$$

$$\epsilon' \cong \frac{x - x'}{1 - 1/[(1+w)f'(x) - w]} \qquad \text{[Eq 3]}$$

where ϵ' and f'(x) refer to the error on x' and the derivative of the original non-linear equation, respectively. The detailed derivation has been omitted as it can be found in the textbook.

Using Equation 2 and 3 on Equation 1, which means applying the overrelaxation method, the number of iterations taken to converge a solution accurate to 10^{-6} and the relative error were obtained with the same initial guess of 'x' and the value of 'c' as in Part a, which are shown in Table 1. The detailed output can be found in Appendix B.

	Number of Iterations	Error *(Three sig. fig.)*
Relaxation Method (c = 2)	14	$-5.01 * 10^{-7}$
Overrelaxation (c = 2, w = 0.5)	4	$-2.43 * 10^{-7}$
Overrelaxation (c = 2, w = 1.0)	5	$4.15 * 10^{-7}$
Overrelaxation (c = 2, w = 1.5)	7	$-5.20 * 10^{-7}$
Overrelaxation (c = 2, w = 2.0)	8	$-8.38 * 10^{-7}$

Table 1: The number of iterations taken to converge to a solution accurate to 10^{-6} and the relative error.

In general, the overrelaxation methods took fewer iterations than the relaxation method in converging to solution accurate to 10^{-6}. For example, the relaxation method took 14 iterations, whereas the overrelaxation method with w = 0.5 took only 4. Therefore, the result agrees with the theoretical expectation that the overrelaxation method is faster than the relaxation method.

Although only positive values of 'w' were tested for the overrelaxation method, there also exist certain circumstances when a negative value of 'w' is faster in finding a solution. Recall the expression $x' = x + (1 + w)\Delta x$. When $w > 0$, $x' > x$. Therefore, the positive 'w' is better when 'x' converges to a greater value than the initial guess. On contrary, the negative 'w' is more suitable when 'x' converges to a smaller value than the initial guess because $x' < x$ when $w < 0$.

Part c - Exercise 6.13 parts (b, c)

Binary Search is an algorithm to find a match by setting an interval of values and narrowing it down by splitting it into half until the desired match is found. Just as the relaxation method or overrelaxation method discussed above, it can also be used to find a solution to a non-linear equation. Although binary search is considered to be fast and powerful, it has a drawback that it must specify two initial points that include, or "bracket" in other words, the solution. If two initial points are set in a way that the interval between them does not contain a solution, then the binary search is unable to return the solution.

Newton's method, however, does not require such specification of an interval that potentially includes the solution. It rather explains the relationship between values with the concept of derivatives, which is also a drawback of Newton's method because the exact form of the derivative must be known in order to apply the Newton's method. Equation 4 and 5 below express the value in the next iteration and the relative error in Newton's method:

$$x' = x - \frac{f(x)}{f'(x)} \qquad \text{[Eq 4]}$$

$$\epsilon' = \frac{-f''(x)}{2f'(x)}\epsilon^2 \qquad \text{[Eq 5]}$$

In practice, Equation 6 below is more often used than Equation 5 for the relative error.

$$x' - x \cong \epsilon \qquad \text{[Eq 6]}$$

In Exercise 6.13, the Planck's radiation law was reduced to a single-variable non-linear equation and solved by using the binary search, Newton's method and as well as the relaxation method. The required expressions can be found below.

Planck's radiation law explains the intensity of radiation per unit area and per unit wavelength λ from a black body at temperature T as follows:

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{\frac{-hc}{\lambda k_B T}}-1} \qquad \text{[Eq 7]}$$

Equation 7 can be reduced to the following equation in terms of the wavelength λ.

$$5e^{\frac{-hc}{\lambda k_B T}} + \frac{hc}{\lambda k_B T} - 5 = 0 \qquad \text{[Eq 8]}$$

Equation 5 can also be reduced to a single-variable equation by making the substitution $x = \frac{hc}{\lambda k_B T}$.

$$5e^{-x} + x - 5 = 0 \qquad \text{[Eq 9]}$$

This substitution is called "Wien displacement law"

$$\lambda = \frac{b}{T} \qquad \text{[Eq 10]}$$

where $b = \frac{hc}{k_B x}$ is called "Wien displacement constant".

As a result, Equation 9 was solved to an accuracy of 10^{-6}. The position of 'x', the number of iterations taken to reach the position, and the relative error to the exact value by each method are summarized in Table 2.

	Position of x	# of iterations	Relative Error
Binary Search (x_1 = 2.0, x_2 = 8.0)	4.9651144742965700	23	$7.15 * 10^{-7}$

Relaxation Method (x = 8.0)	4.9651142800940375	5	-4.83* 10^{-8}
Newton's Method (x = 8.0)	4.9651142317442770	4	-3.57 * 10^{-12}

Table 2: Position of 'x' to an accuracy of 10^{-6} and the number of iterations taken to reach the position were obtained by three methods in the first column. Note that the position of 'x' is presented with more than three significant figures to show the differences among them. The interval for the binary search was [2.0, 8.0], whereas the initial guess was 8.0 for both relaxation and Newton's method.

As expected in the textbook theoretically, the Newton's method was the fastest among three, with only 4 iterations. The relaxation method is relatively slower than Newton's method, which required 5 iterations. However, the binary search underwent a far greater number of iterations, requiring 23 iterations. Therefore, the most and least efficient method were Newton's method and Binary Search, respectively.

Using the relation $b = \frac{hc}{k_B x}$ and Equation 10, Wien displacement constants and the surface temperatures of the Sun were obtained and listed in Table 3.

	Wien displacement constant [meter*Kelvin]	Surface temperature of the Sun [Kelvin]
Binary Search	0.002899778859 ±0.000000000442	5776.451910 ±0.000881
Relaxation Method	0.002899778972 ±0.000000000448	5776.452136 ±0.000892
Newton's Method	0.0028997790009 ±0.0000000000008	5776.452193 ±0.000010

Table 3: Wien displacement constant and surface temperature of the Sun for each method. Note that the values are presented with more than three significant figures for better accuracy in comparison.

As the relative error was the smallest for Newton's method, the uncertainties obtained for Wien displacement constant and the surface temperature of the Sun by Newton's method were also the smallest among three methods. However, all three methods succeeded to obtain the similar results in comparison of each other.

Appendix A – Output of Relaxation Method

```
Exercise 6.10 Part a -  1 th iteration: 0.8646647167633873

Exercise 6.10 Part a -  2 th iteration: 0.8225966691808597

Exercise 6.10 Part a -  3 th iteration: 0.8070247503317746

Exercise 6.10 Part a -  4 th iteration: 0.8009201942366403
Exercise 6.10 Part a -  5 th iteration: 0.7984747083247583
Exercise 6.10 Part a -  6 th iteration: 0.7974866394625985
Exercise 6.10 Part a -  7 th iteration: 0.797086049491047
Exercise 6.10 Part a -  8 th iteration: 0.796923413762181
Exercise 6.10 Part a -  9 th iteration: 0.7968573480008734
Exercise 6.10 Part a -  10 th iteration: 0.7968305046795713
Exercise 6.10 Part a -  11 th iteration: 0.7968195968986895
Exercise 6.10 Part a -  12 th iteration: 0.7968151643557075
Exercise 6.10 Part a -  13 th iteration: 0.7968133630966887
Exercise 6.10 Part a -  14 th iteration: 0.7968126311118457
Exercise 6.10 Part a -  15 th iteration: 0.7968123336514794
Exercise 6.10 Part a -  16 th iteration: 0.7968122127708882
Exercise 6.10 Part a -  17 th iteration: 0.7968121636479626
Exercise 6.10 Part a -  18 th iteration: 0.7968121436855996
Exercise 6.10 Part a -  19 th iteration: 0.7968121355733799
Exercise 6.10 Part a -  20 th iteration: 0.7968121322767707
Exercise 6.10 Part a -  21 th iteration: 0.7968121309371088
Exercise 6.10 Part a -  22 th iteration: 0.7968121303927027
Exercise 6.10 Part a -  23 th iteration: 0.7968121301714692
Exercise 6.10 Part a -  24 th iteration: 0.7968121300815654
Exercise 6.10 Part a -  25 th iteration: 0.7968121300450306
Exercise 6.10 Part a -  26 th iteration: 0.7968121300301837
Exercise 6.10 Part a -  27 th iteration: 0.7968121300241503
Exercise 6.10 Part a -  28 th iteration: 0.7968121300216985
Exercise 6.10 Part a -  29 th iteration: 0.7968121300207021
Exercise 6.10 Part a -  30 th iteration: 0.7968121300202972
Exercise 6.10 Part a -  31 th iteration: 0.7968121300201326
Exercise 6.10 Part a -  32 th iteration: 0.7968121300200658
Exercise 6.10 Part a -  33 th iteration: 0.7968121300200386
Exercise 6.10 Part a -  34 th iteration: 0.7968121300200276
Exercise 6.10 Part a -  35 th iteration: 0.796812130020023
Exercise 6.10 Part a -  36 th iteration: 0.7968121300200213

Exercise 6.10 Part a -  37 th iteration: 0.7968121300200206

Exercise 6.10 Part a -  38 th iteration: 0.7968121300200203

Exercise 6.10 Part a -  39 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  40 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  41 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  42 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  43 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  44 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  45 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  46 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  47 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  48 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  49 th iteration: 0.7968121300200202

Exercise 6.10 Part a -  50 th iteration: 0.7968121300200202

Note that the value of x converges to 0.7968121300200202
```

Appendix B – Output of Overrelaxation Method

```
Exercise 6.11 Part b - Relaxation Method

Exercise 6.11 Part b - The estimate (starting point x = 1.0 , c = 2 ) is 0.7968126311118457

Exercise 6.11 Part b - Starting from x = 1.0 , the error on 14 th iteration is -5.010892644903239e-07

This is close to a solution with an accuracy of less than 10^-6

================================================================================
====================

Exercise 6.11 Part c - Overrelaxation Method

Exercise 6.11 Part c - The estimate (w = 0.5 , starting point x = 1.0 , c = 2 ) is 0.7968123729832619

Exercise 6.11 Part c - Starting from x = 1.0 , the error on 4 th iteration is -2.4295987529549827e-07

This is close to a solution with an accuracy of less than 10^-6

================================================================================
====================

Exercise 6.11 Part c - Overrelaxation Method

Exercise 6.11 Part c - The estimate (w = 1.0 , starting point x = 1.0 , c = 2 ) is 0.7968117147771608
Exercise 6.11 Part c - Starting from x = 1.0 , the error on 5 th iteration is 4.1524622570953174e-07

This is close to a solution with an accuracy of less than 10^-6

================================================================================
====================

Exercise 6.11 Part c - Overrelaxation Method

Exercise 6.11 Part c - The estimate (w = 1.5 , starting point x = 1.0 , c = 2 ) is 0.7968126496332508

Exercise 6.11 Part c - Starting from x = 1.0 , the error on 7 th iteration is -5.196124420355472e-07

This is close to a solution with an accuracy of less than 10^-6

================================================================================
====================

Exercise 6.11 Part c - Overrelaxation Method

Exercise 6.11 Part c - The estimate (w = 2.0 , starting point x = 1.0 , c = 2 ) is 0.796812968245689

Exercise 6.11 Part c - Starting from x = 1.0 , the error on 8 th iteration is -8.382248802459555e-07

This is close to a solution with an accuracy of less than 10^-6
```