NMiNE

Home Assignment

LAB 1

By
Faisal Ahmed Moshiur
Date: 31/01/2023

Q1:

Answers

1a: $x = 2$

1b: $x = 4$

1c: Three solutions from visual inspection of the graph shown below:

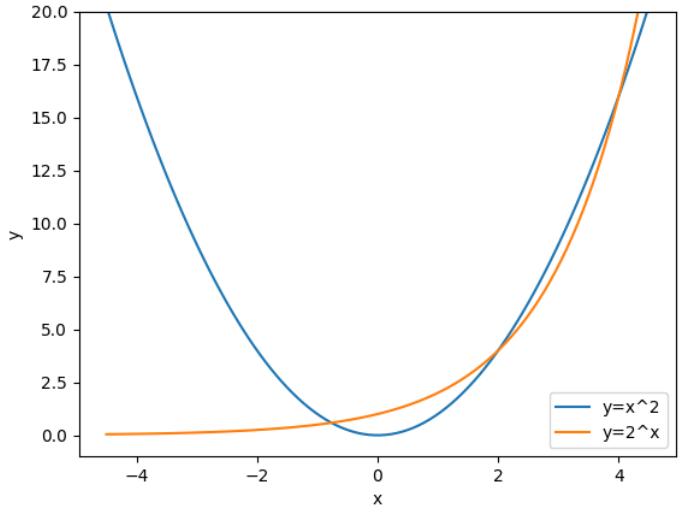

Fig 1c: Solution for equation $x^2 = 2^x$ depicted by points of intersections between the curves $y = x^2$and $y = 2^x$.

1d: Other roots are not possible because the two equations $y = x^2$and $y = 2^x$ represent curves having different rates of change of x with respect to y i.e., dy/dx. Therefore, the curves will deviate from each other and point of intersection i.e., their solutions are not possibles in the interval of numbers in real number domain.

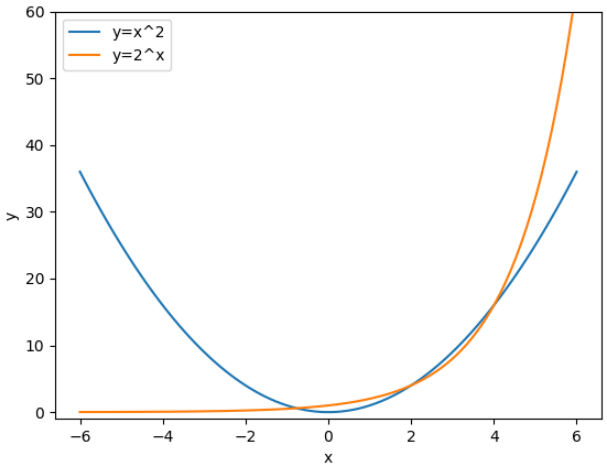

Fig 1d: Curves for $y = x^2$and $y = 2^x$ in a different scale to depict their deviation from each other beyond the three intersection points.

Q2:

Answers

2a: `[-sqrt(2), sqrt(2)]`

2b: `[2, 4, -2*LambertW(log(2)/2)/log(2)]`

Q3:

Answers

3a: The equation $sgn(x)2^{\left(\frac{x}{2}\right)} = 0$ can have either 0 or 2 solutions, as sgn(x) can only be 1, 0, or -1, and $2^{\left(\frac{x}{2}\right)}$ can either be positive or negative. However, the number of solutions of this equation depends on the domain of x that you are considering.

3b: The equation $sgn(x)2^{\left(\frac{x}{2}\right)} = 0$ *has two solutions, x = 2 and x = 4. To check* the convergence condition, we need to calculate the derivative of the function $f(x) = sgn(x)2^{\left(\frac{x}{2}\right)}$ at the two solutions and determine if the derivative is less than 1 in magnitude. If the magnitude of the derivative is less than 1, it is a sufficient condition for the fixed-point iteration method to converge.

At $x = 2$, the derivative can be calculated as:

$$df/dx = d\left(sgn(x)2^{\left(\frac{x}{2}\right)}\right)/dx$$

$$= \left(d\left(sgn(x)\right)/dx\right)\left(2^{\left(\frac{x}{2}\right)}\right) + sgn(x)\left(d\left(2^{\left(\frac{x}{2}\right)}\right)/dx\right)$$

$$= (0)\left(2^{(2/2)}\right) + sgn(2)\left(2^{(2/2)}(1/2)ln(2)\right)$$

$$= 2ln(2)$$

Since the magnitude of the derivative at $x = 2$ is greater than 1, the fixed-point iteration method may not converge at this solution.

At $x = 4$, the derivative can be calculated as:

$$df/dx = d\left(sgn(x)\left(2^{(x/2)}\right)\right)/dx$$

$$= \left(d\left(sgn(x)\right)/dx\right)\left(2^{(x/2)}\right) + sgn(x)\left(d\left(2^{(x/2)}\right)/dx\right)$$

$$= (0)\left(2^{(4/2)}\right) + sgn(4)\left(2^{(4/2)}(1/2)ln(2)\right)$$

$$= 4ln(2)$$

Since the magnitude of the derivative at $x = 4$ is greater than 1, the fixed-point iteration method may not converge at this solution.

Therefore, based on the convergence condition, the fixed-point iteration method may not converge at the solutions $x = 2$ and $x = 4$ for the equation $sgn(x)2^{\left(\frac{x}{2}\right)} = 0$.

3c: The convergence condition cannot be determined for the negative root of the equation $sgn(x)2^{\left(\frac{x}{2}\right)} = 0$ because $sgn(x)$ is not defined for negative numbers. As per the definition provided, $sgn(x)$ is equal to 0 when $x = 0$ and is equal to -1 when $x < 0$, but it is not defined for negative x.

3d: At $x_0 = 4.5$,

```
    return sgn(x) * (2**(x/2))
                     ~^^~~~~~
OverflowError: (34, 'Result too large')
```

3e: At $x_0 = 4$,

```
The result after fixed-point iteration is: 4.0
```

3f: At $x_0 = 2.5$,

```
The result after fixed-point iteration is: 2.0002178009941285
```

3g: At $x_0 = -1.5$,

```
The result after fixed-point iteration is: -0.7666811662504153
```

Q4:

Answers

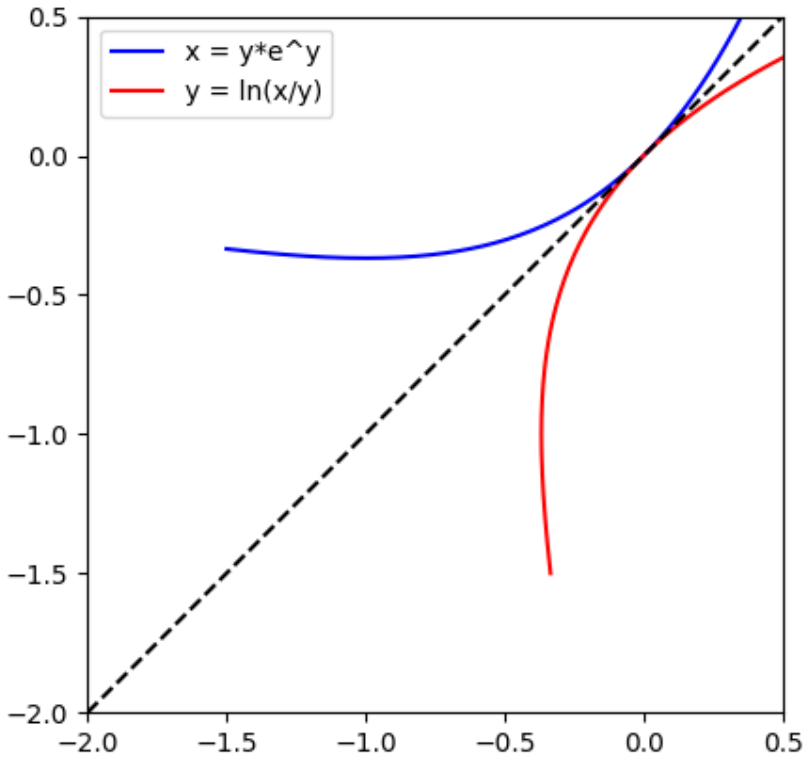

Fig 1: The function $x = ye^y$, and its inverse plotted with the bisector y = x.

Q5:

Answers

5a: The derivative of the function $\mathrm{x}(y) = \mathrm{y}e^{y}$ can be calculated analytically using the product rule of differentiation:

$$x'(y) = (ye^{y})' = e^{y} + \mathrm{y}(e^{y})' = e^{y} + \mathrm{y}e^{y} = (1 + y)e^{y}$$

The derivative $\mathrm{x}'(\mathrm{y})$ is positive in the region where $1 + \mathrm{y} > 0$, or equivalently, $\mathrm{y} > -1$. In this region, $\mathrm{x}(\mathrm{y})$ is increasing. The derivative $\mathrm{x}'(\mathrm{y})$ is equal to zero when $1 + \mathrm{y} = 0$, or $\mathrm{y} = -1$. Finally, $\mathrm{x}'(\mathrm{y})$ is negative in the region where $1 + \mathrm{y} < 0$, or $\mathrm{y} < -1$. In this region, $\mathrm{x}(\mathrm{y})$ is decreasing.

5b: The function $\mathrm{x}(y) = \mathrm{y}e^{y}$ is increasing when $\mathrm{y} \geq -1/\mathrm{e}$ and decreasing when $\mathrm{y} < -1/\mathrm{e}$. Hence, the monotonicity regions of $\mathrm{x}(\mathrm{y})$ can be labeled as $R_0 = [-1/e, \infty)$ and $R_{-1} = (-\infty, -1/e)$.

Since $\mathrm{x}(y)$ is a one-to-one function, it has exactly one inverse for each monotonicity region. Therefore, the real argument Lambert function W has two branches, one for each monotonicity region. The principal branch W_0 is defined as the inverse of $\mathrm{x}(y)$ over R_0, while the complementary branch $W_{-1}(x)$ is defined as the inverse of $\mathrm{x}(y)$ over R_{-1}.

5c: The minimum value and corresponding argument of $\mathrm{x}(y) = \mathrm{y}e^{y}$ can be found by setting the derivative of $x(y)$ to zero and solving for y.

The derivative of $\mathrm{x}(y)$ is given by:

$x'(y) = (ye^{y})' = (1 + y)e^{y}$

Setting $x'(y) = 0$, we have:

$$(1 + y)e^{y} = 0$$

Dividing both sides by e^{y}, we get:

$$1 + y = 0$$

Solving for y, we find that $y = -1$.

Substituting $y = -1$ into $x(y)$, we find that the minimum value is:

$$x_{min} = \mathrm{x}(-1) = -e^{-1}$$

So the minimum value of x(y) is $x_{min} = -e^{-1}$ and the corresponding argument is $y_{min} = -1$.

5d: Therefore, the function $\mathrm{x}(y) = \mathrm{y}e^{y}$ is increasing in the interval $(-1, +\infty)$, and is decreasing in the interval $(-\infty, -1)$.

The real argument Lambert function W has two branches: the main branch, $W_0(x)$, which is increasing for $\mathrm{x} > -1/\mathrm{e}$ and the complementary branch, $W_{-1}(x)$, which is decreasing for $\mathrm{x} < -1/\mathrm{e}$.

5e:

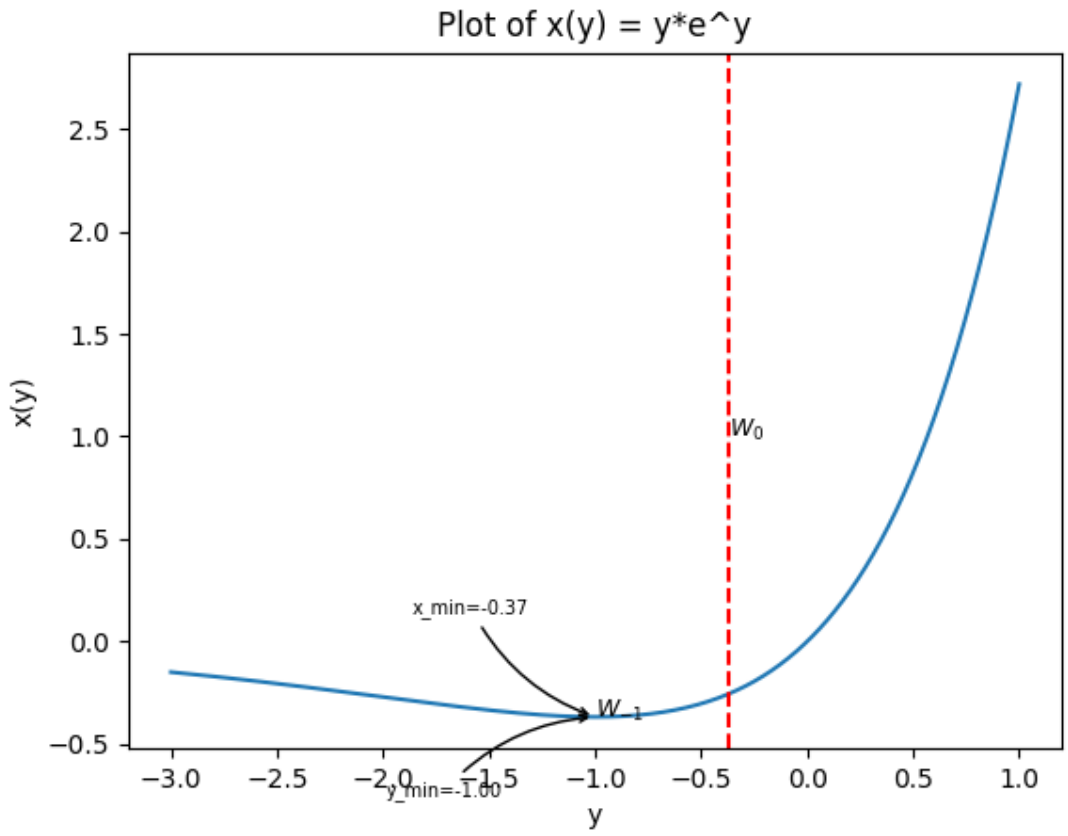

Fig 1: Plot of $x(y) = ye^{y}$ with the characteristic properties shown in the labelling.

Q6:

Answers

6a:

```
Number of iterations: 4
Actual absolute error: 1.5947243525715749e-12
```

6b:

```
Result: 0.7549945974339535
Actual absolute error: 0.06184741687400819
Number of iterations: 100
```

6c:

```
Number of iterations: 6
Actual absolute error: 4.21610160114827e-05
```

6d:

```
Solution for the equation with y0=-2 : -1.4142135623746899
```

6e:

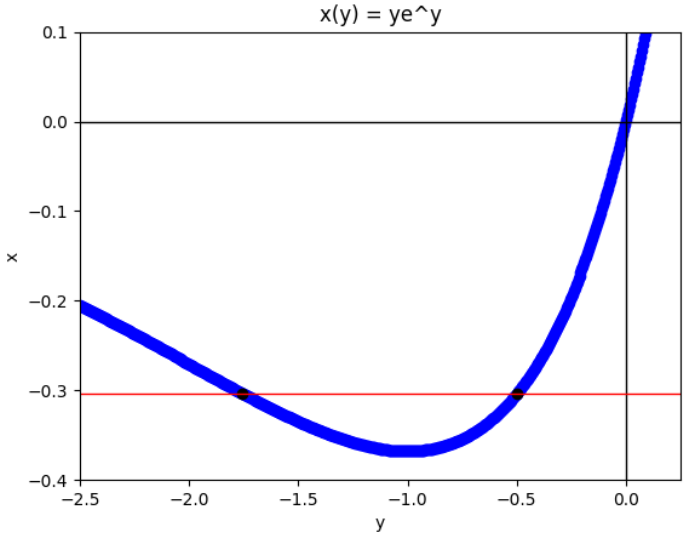

Fig 1: Graph for function, $x(y) = ye^{y}$ with its solution marked by black contour.

Q7:

Answers

Newton's method for a general nonlinear equation $f(y) = 0$ can be derived as follows:

Given a starting guess y_0, the iteration formula for Newton's method is given by:

$$y_{n+1} = y_n - f(y_n)/f'(y_n)$$

where $f'(y_n)$ is the derivative of $f(y)$ evaluated at y_n.

On the other hand, Halley's method is a higher-order version of Newton's method. It is based on a second-order Taylor series expansion of $f(y)$ near y_n:

$$f(y) = f(y_n) + f'(y_n)(y - y_n) + (1/2)f''(y_n)(y - y_n)^2$$

The iteration formula for Halley's method is given by:

$$y_{n+1} = y_n - 2f(y_n)/\big(2f'(y_n) + f''(y_n)(y_n - y_n)\big)$$

Adapting Halley's method to the Lambert equation, $ye^y = x$, we need to first define the function $f(y)$ as:

$$f(y) = ye^y - x$$

The derivatives of f(y) are given by:

$$f'(y) = e^y + ye^y$$

$$f''(y) = 2e^y$$

Substituting these into the Halley's iteration formula, we obtain:

$$\begin{aligned} y_{n+1} &= y_n - 2f(y_n)/\big(2f'(y_n) + f''(y_n)(y_n - y_n)\big) \\ &= y_n - 2ye^y/(2(e^y + ye^y) + 2e^y) \\ &= y_n - 2ye^y/(2e^y + 2ye^y + 2e^y) \\ &= y_n - (2y)/(2 + 2y + 2) \\ &= y_n - y/(1 + y) \end{aligned}$$

7a: Newton's method is a widely used iterative method for solving nonlinear equations. Given an initial guess x_0, it is based on the idea that an equation $f(x) = 0$ can be locally approximated as a straight line. The slope of this line is given by the derivative of $f(x)$ at x_0, and the intersection of this line with the x-axis gives an improved estimate of the solution x. The process is repeated until convergence.

The explicit form of Newton's method is given by:

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

where x_n is the n-th approximation to the solution and $f'(x_n)$ is the derivative of $f(x)$ evaluated at x_n.

It is important to note that the convergence of Newton's method depends on the initial guess x_0 being close enough to the solution and the function $f(x)$ being well-behaved (i.e., having a continuous derivative that doesn't change sign over the interval containing the solution).

7b: Halley's method is an extension of Newton's method and is a second-order Householder's method with a convergence rate that is cubic. It can be derived explicitly as follows:

Let $f(x)$ be a real-valued function and let x_n be the current approximation of the root of the equation $f(x) = 0$.

1. Compute the first and second derivatives of the function: $f'(x)$ and $f''(x)$, respectively.
2. Update the approximation using the formula:
$$x_{n+1} = x_n - 2f(x_n)f'(x_n)/\big(2f'(x_n)^2 - f(x_n)f''(x_n)\big)$$
3. Repeat the process until the desired tolerance is reached.

The convergence rate of Halley's method is cubic, which means that the number of correct digits is multiplied by 3 in each iteration. In addition, Halley's method is relatively insensitive to the choice of the initial guess and tends to converge faster than Newton's method when the derivative information is accurate.

7c: The Lambert equation, $ye^y = x$, can be transformed to the general form, $f(y) = 0$, by defining $f(y)$ as:

$$f(y) = ye^y - x$$

The first derivative of $f(y)$ is:

$$f'(y) = e^y + ye^y$$

The second derivative of f(y) is:

$$f''(y) = 2e^y + ye^y$$

The ratio $f(y)/f'(y)$ is:

$$f(y)/f'(y) = (ye^y - x)/(e^y + ye^y)$$

The ratio $f''(y)/f'(y)$ is:

$$f''(y)/f'(y) = (2e^y + ye^y)/(e^y + ye^y) = 2 + y$$

These ratios are used to calculate the updated value of y in Halley's method, which is a Householder's method of the second order, $d = 2$.

7d: Halley's method is a second-order Householder's method, which means that it considers both the first and second derivatives of the equation being solved. To formulate an explicit version of Halley's method adapted to the Lambert equation, $ye^y = x$, we first need to transform the equation into the general form, $f(y) = 0$. This can be done by defining $f(y) = ye^y - x$.

The first derivative of $f(y)$ can be calculated as:

$$f'(y) = e^y + ye^y$$

And the second derivative of f(y) can be calculated as:

$$f''(y) = 2e^y$$

Next, we calculate the ratios of $f(y)$ to $f'(y)$ and $f''(y)$ to $f'(y)$ that are used in Halley's method:

$$f(y)/f'(y) = (ye^y - x)/(e^y + ye^y)$$

$$f''(y)/f'(y) = 2e^y/(e^y + ye^y)$$

Finally, we can use these ratios in Halley's method to find an explicit formula for the solution:

$$y_{n+1} = y_n - 2 * f(y_n)/\left(2 * f'(y_n) - f(y_n)f''(y)/f'(y_n)\right)$$

Where y_n is an initial guess for the solution and y_{n+1} is the improved estimate at each iteration.

Q8:

```
import numpy as np

def Lambert_W(x, branch=0, tol=1e-8, init=1):
    # Check if the input arguments are consistent
    if x < 0:
        if branch == 0:
            print("Error: x < 0 and branch = 0. Choose branch = -1 instead.")
            return None, None
    if x >= np.exp(-1):
        if branch == -1:
            print("Error: x >= e^(-1) and branch = -1. Choose branch = 0 instead.")
            return None, None
    if tol <= 0 or tol > 1e-1:
        print("Error: Tolerance must be 0 < tol < 1e-1.")
        return None, None
    if init != 0 and init != 1:
        print("Error: init must be either 0 or 1.")
        return None, None

    # Define the maximum number of iterations
    itMax = 30


    # Initial guess for y_0
    if branch == 0 and init == 0:
        if x == -1 / np.exp(1):
            y = -1

        elif -1 / np.exp(1) < x < 0:
            y= (np.exp(1) * x / (1 + np.exp(1) * x + np.sqrt(1 + np.exp(1) * x))) * np.log(1 +
np.sqrt(1 + np.exp(1) * x))

        elif 0 < x < np.exp(1):
            y = x / np.exp(1)

        else:
            y = np.log(x) - np.log(np.log(x))

    if branch == -1 and init == 1:
        if x == -1 / np.exp(1):
            y = -1

        elif -1 / np.exp(1) < x < -1 / 4:
            y = -1 - np.sqrt(2 * (1 + np.exp(1) * x))

        elif -1 / 4 < x < 0:
            y = np.log(-x) - np.log(-np.log(-x))


    # Implement Halley's method
    it = 0
    while it < itMax:
        f = y * np.exp(y) - x
        f_prime = np.exp(y) * (y + 1)
        f_double_prime = np.exp(y) * (2 * y + 2)
        y_new = y - 2 * f * f_prime / (2 * f_prime**2 - f * f_double_prime)

        # Check if the tolerance is reached
        if abs(y_new - y) <= tol:
            break
```

```
        y = y_new
        it += 1

    if it == itMax:
        print("Warning: Maximum number of iterations reached. Tolerance may not be achieved.")

    return y, it

# Evaluate the omega constant
omega, it = Lambert_W(1, branch=0, tol=1e-8, init=0)
print(f"Omega obtained via calculation:{omega}\nDiffrence between the omega obtained in this calculation and Wikipedia value:{abs(omega - 0.56714329040978387299968662210)}")
print("Number of iterations:", it)
```

Answer:

```
Omega obtained via calculation:0.567143290389849
Diffrence between the omega obtained in this calculation and Wikipedia value:1.9934831563261923e-11
Number of iterations: 3
```

Q10:

Answers

Seven years later, in 1900, Max Planck derived Planck's Law, which describes the spectral density of electromagnetic radiation from a black body, formulated as:

$$E(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}$$

--- (1)

Planck's Law produces a continuous function unique to each black body temperature. Wien's Law determines the wavelength of peak emission, so deriving Wien's Law involves finding the maximum value of Planck's Law as a function of temperature.

The first step is to take the partial derivative of Planck's Law (1) with respect to wavelength, λ.

$$\frac{\partial \mathrm{E}}{\partial \lambda} = \frac{2hc^2}{\lambda^6 \left(e^{\frac{hc}{\lambda k_B T}} - 1 \right)} \left(\frac{e^{\frac{hc}{\lambda k_B T}}}{e^{\frac{hc}{\lambda k_B T}} - 1} - 5 \right)$$

---(2)

Next, setting (2) equal to zero and simplifying:

$$\frac{hc}{\lambda k_B T} \left(\frac{e^{\frac{hc}{\lambda k_B T}}}{e^{\frac{hc}{\lambda k_B T}} - 1} - 5 \right) = 0$$

---(3)

Defining $x \equiv \frac{hc}{\lambda k_B T}$, equation (3) becomes:

$$\frac{xe^x}{e^x - 1} - 5 = 0$$

---(4)

Rearranging equation (4) gives:

$$e^x(x-5)+5=0$$

---(5)

This is the said compact transcendental equation from which Wien's displacement constant b, can be calculated as given in question

$$\lambda_{max}=\frac{b}{T}$$

Since, $x \equiv \frac{hc}{\lambda k_B T}$, we can get the value of b from numerically calculated value of x, by

$$b=\lambda_{max}T=\frac{hc}{xk_B}$$

NIST database value for the following constants:

$h = 6.62607015 \times 10^{-34}\, Js$

$c = \ 299792458\, ms^{-1}$

$k_B = 1.380649 \times 10^{-23}\, JK^{-1}$

Newton-Raphson's method is used to solve (5) and use that value of x to obtain value of b to 10 decimal places:

```
The solution is: x = 4.965114231744276
The value of b is: b = 0.0028977720
```