NMiNE

Home Assignment

LAB 1

By
Faisal Ahmed Moshiur
Date: 19/02/2023

Q1:

<u>Answers</u>

1a: $x\ = 2$

1b: $x = 4$

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

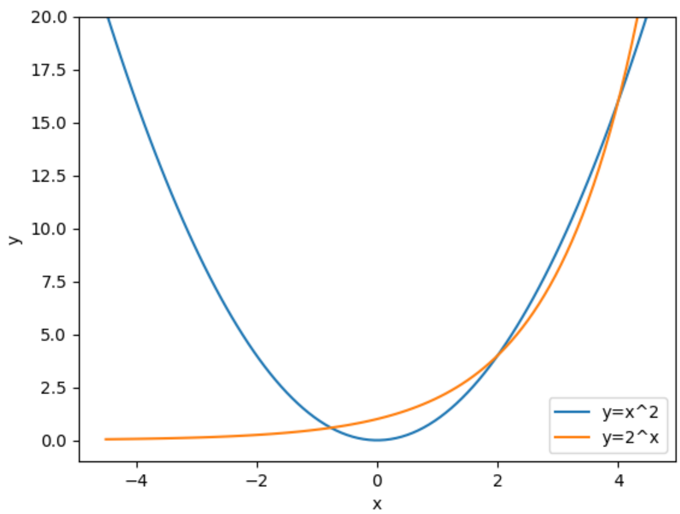


Fig 1c: Solution for equation $x^2 = 2^x$ depicted by points of intersections between the curves $y = x^2$and $y = 2^x$.

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.

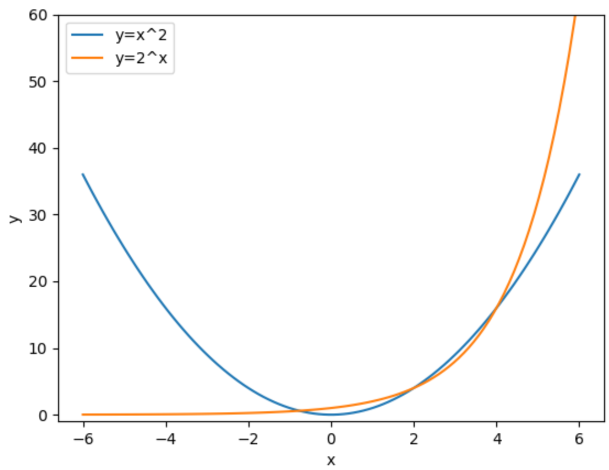


Fig 1d: Curves for $y = x^2$and $y = 2^x$ in a different scale to depict their deviation from each other beyond the three intersection points.

Q2:

<u>Answers</u>

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

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

Q3:

Answers

3a: For $x > 0$, $sgn(x) = 1$ so we have two solutions vis 2 and 4.

For $x = 0$, $sgn(x) = 0$ so we have one solution i.e., 0.

However, for $x < 0$, $sgn(x) = -1$ so we have $x = -2\frac{W_0\frac{ln(2)}{2}}{ln(2)}$.

3b: To check the convergence condition, we need to calculate the absolute value of 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:

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

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

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

$$= ln(2) < 1$$

Since the magnitude of the derivative at $x = 2$ is less than 1, the fixed-point iteration method converges at this solution.

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

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

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

$$= 2ln(2) > 1$$

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

3c: For negative roots,

$$\frac{d}{dx}\left[-2^{\frac{x}{2}}\right]$$

$$= -\frac{d}{dx}\left[2^{\frac{x}{2}}\right]$$

$$= -\ln(2) \cdot 2^{\frac{x}{2}} \cdot \frac{d}{dx}\left[\frac{x}{2}\right]$$

$$= -\ln(2) \cdot 2^{\frac{x}{2}} \cdot \frac{1}{2} \cdot \frac{d}{dx}[x]$$

$$= -\ln(2) \cdot 1 \cdot 2^{\frac{x}{2}-1}$$

$$= -\ln(2) \cdot 2^{\frac{x}{2}-1}.$$

Hence, for the negative values will always give absolute value of derivatives less than 1 which means as shown above it will converge.

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

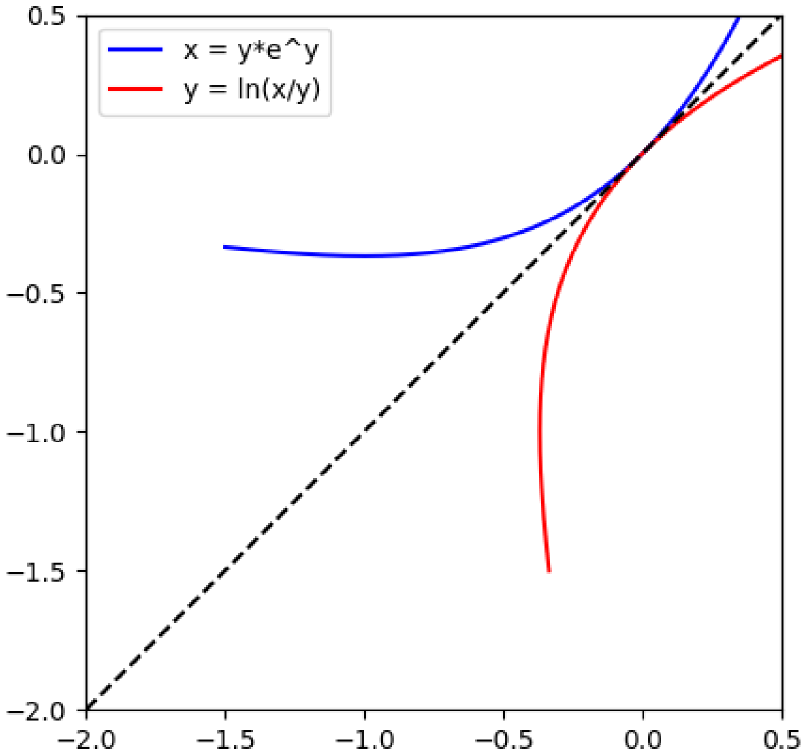


Fig 1: The function $x = ye^{y}$, and its inverse plotted with the bisector y = x.

Q5:

Answers

5a: The derivative of the function $x(y) = ye^y$ can be calculated analytically using the product rule of differentiation:

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

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

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

Since $x(y)$ 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 $x(y)$ over R_0, while the complementary branch $W_{-1}(x)$ is defined as the inverse of $x(y)$ over R_{-1}.

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

The derivative of $x(y)$ 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 x $> -1/\mathrm{e}$ and the complementary branch, $W_{-1}(x)$, which is decreasing for x $< -1/\mathrm{e}$.

5e:

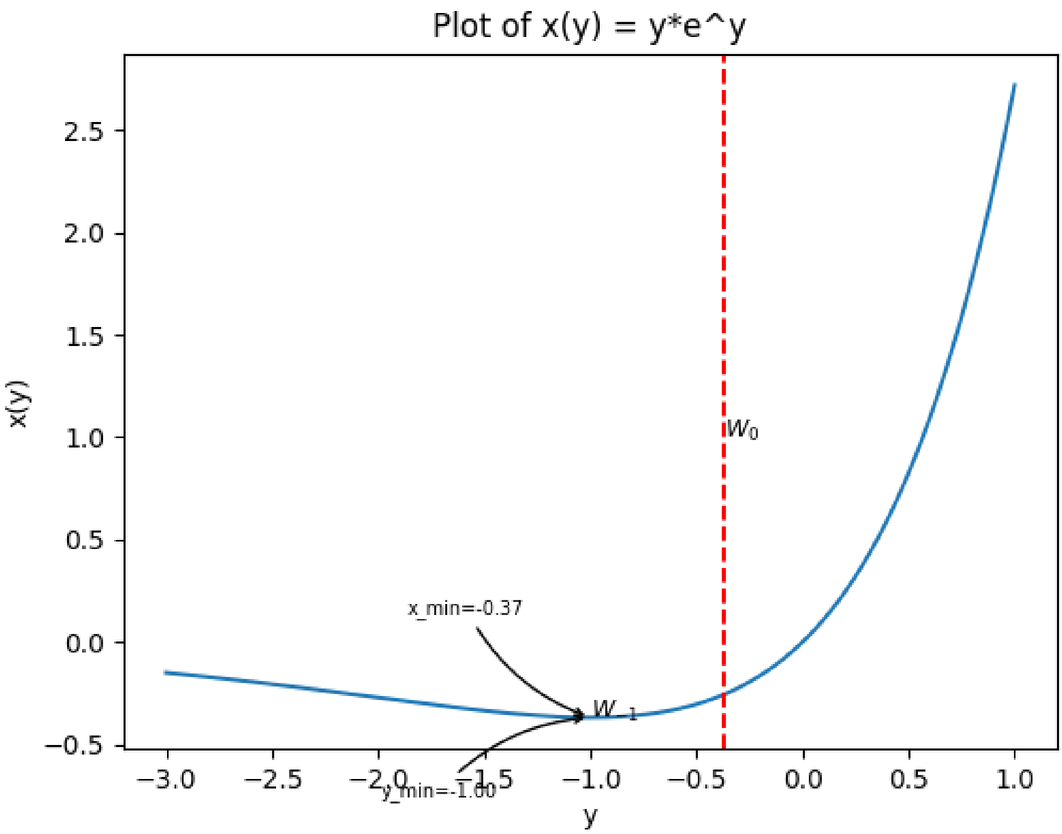


Fig 1: Plot of $x(y) = ye^{y}$ with the characteristic properties shown in the labelling.

Q6:

Answers

6a:

```
Number of iteration: 4,
Actual absolute error: 1.5947243525715749e-12,
Value of square root of 2: 1.4142135623746899
```

6b:

```
Newton's method solution:
 y = 0.6931471805963518,
 absolute error = 1.2328316145726603e-10,
 iterations = 4
```

6c:

```
Newton's method solution:
 y = -0.4999999998556274,
 absolute error = 4.3783199288327523e-11,
 iterations = 7
```

6d:

```
Newton's method solution:
 y = -1.7564312086263951,
 absolute error = 2.942091015256665e-14,
 iterations = 3
```

6e:

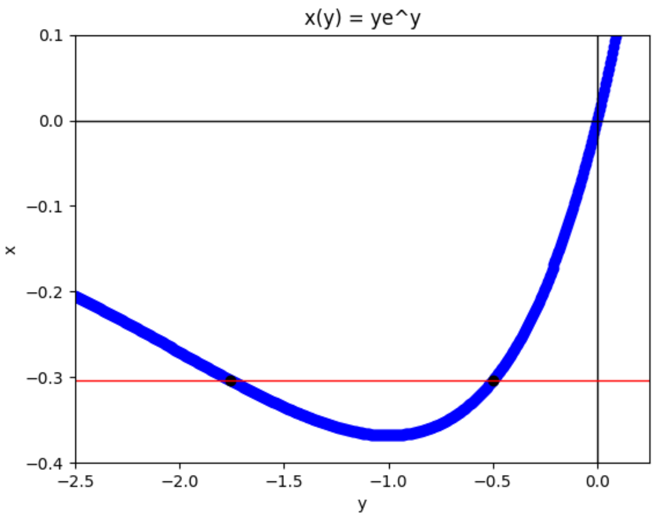


Fig 1: Graph for function, $x(y) = ye^y$ with its solution marked by black contour.

Q7:

Answers

7a: Householder's method is a numerical algorithm for solving the nonlinear equation $f(x) = 0$. In this case, the cubic convergence rate of one real variable. The method consists of a sequence of iterations

$$x_{n+1} = x_n + d\,\frac{(1/f)^{(d-1)}(x_n)}{(1/f)^{(d)}(x_n)}$$

For $d = 1$, due to Newton's method from Householder's method:

$$x_{n+1} = x_n + 1\,\frac{(1/f)(x_n)}{(1/f)^{(1)}(x_n)}$$

$$= x_n + \frac{1}{f(x_n)} \cdot \left(\frac{-f^{'}(x_n)}{f(x_n)^2}\right)^{-1}$$

$$= x_n - \frac{f(x_n)}{f^{'}(x_n)}$$

7b: For $d = 2$, due to Halley's method from Householder's method:

$$(1/f)^{'}(x) = -\frac{f^{'}(x)}{f(x)^2}$$

And

$$(1/f)^{''}(x) = -\frac{f^{''}(x)}{f(x)^2} + 2\,\frac{f^{'}(x)^2}{f(x)^3}$$

Therefore,

$$x_{n+1} = x_n + 2\,\frac{(1/f)^{'}(x_n)}{(1/f)^{''}(x_n)}$$

$$= x_n + \frac{-2f(x_n)f^{'}(x_n)}{-f(x_n)f^{''}(x_n) + 2[f^{'}(x_n)]^2}$$

$$= x_n - \frac{f(x_n)f^{'}(x_n)}{f^{'}(x_n)^2 - \frac{1}{2}f(x_n)f^{''}(x_n)}$$

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:

$$f'(y) = (1 + y)e^y$$

The second derivative of f(y) is:

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

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

$$f(y)/f'(y) = (ye^y - x)/(1 + y)e^y$$

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

$$f''(y)/f'(y) = (2 + y)e^y/(1 + y)e^y = (2 + y)/(1 + \mathrm{y})$$

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:

We can rewrite the form of equation of Halley’s method in terms of fraction of the derivatives:

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{'}(x_n) - \frac{f(x_n)}{f^{'}(x_n)} \frac{f^{''}(x_n)}{2}}$$

$$= x_n - \frac{f(x_n)}{f^{'}(x_n)} \left[1 - \frac{f(x_n)}{f^{'}(x_n)} \cdot \frac{f^{''}(x_n)}{2f^{'}(x_n)} \right]^{-1}$$

From the values obtained in previous exercises, we can now show that

$$y_{n+1} = y_n - \frac{y_\mathrm{n} e^{y_\mathrm{n}} - x}{(1 + y_\mathrm{n})e^{y_\mathrm{n}}} \left[1 - \frac{(y_\mathrm{n} e^{y_\mathrm{n}} - x)}{(1 + y_\mathrm{n})e^{y_\mathrm{n}}} \frac{(2 + y_\mathrm{n})}{2(1 + y_\mathrm{n})} \right]^{-1}$$

$$= y_\mathrm{n} - \frac{2(1 + y_\mathrm{n})(y_\mathrm{n} e^{y_\mathrm{n}} - x)}{2e^{y_n}(1 + y_n)^2 - [(y_\mathrm{n} e^{y_\mathrm{n}} - x)(2 + y_\mathrm{n})]}$$

Q8:

```python
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
```

```python
        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 - 0.567143290409783872999968662210)}")
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
```

Q9:

<u>Answers:</u>

9a & 9b:

Table 1. Comparison of Newton's and Halley's methods for evaluating W_0(x).									
	Exact:	y = -1		y = -1+2^-10		y = -1/2		y = 8	
Method		Err	Iter	Err		Err		Err	
Newton	y_0 = 1	6.85403323608113E-06	20	2.6406619413332777e-07	16	9.81057742477364E-06	7	0.9997669027670781	100
	y_0 = opt	6.85403323608113E-06	20	4.419464361138381e-07	14	1.9162112323001246e-06	5	1.850441471162867e-07	4
Halley	y_0 = 1	5.272811389526445e-06	10	5.272835717664775e-06	10	4.389252938294686e-06	10	1.49569440921482E-06	12
	y_0 = opt	zero division error		4.651583286588906e-06	5	6.26816346419639E-06	9	7.48312286447117E-06	7

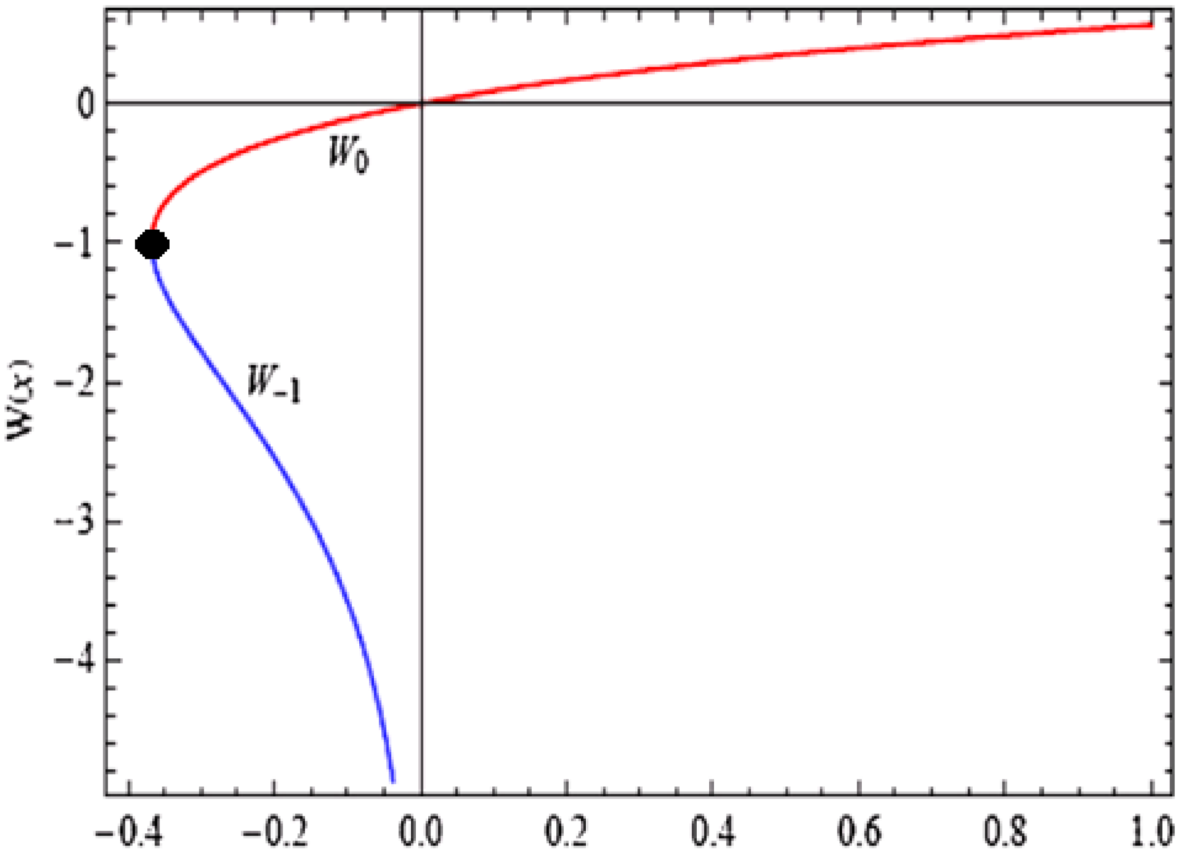
Table 2. Comparison of Newton's and Halley's methods for evaluating W_{-1}(x).									
	Exact:	y = -1		y = -1-2^-10		y = -1.5		y = -8	
Method		Err	Iter	Err		Err		Err	
Newton	y_0 = -2	7.099431064139239e-06	16	3.619090791495694e-07	12	5.587808349361012e-08	4	2.8619249192729512e-09	9
	y_0 = opt	7.099431064139239e-06	16	3.178784027113579e-07	1	6.338421268958783e-06	3	1.2281669636848846e-07	4
Halley	y_0 = -2	6.17934179682159E-06	24	6.917772354589452e-06	24	3.9312309428995e-06	20	1.73401522976181e-07	13
	y_0 = opt	zero division error		4.3133640301703555e-06	52	5.950298098493807e-06	18	2.06528825444302E-07	15

Table 1 shows that usage of the optimized value for Newton's method increases efficiency, especially as the value of y increases. However, the optimized value for Halley's methods shows some fluctuations in the result in terms of error and iteration count. However, in table 2, i.e., in the $W\_\{-1\}$ region, Halley's method shows fluctuations in terms of error. Still, the number of iterations drops gradually as we decrease the value of y further.

Moreover, Newton's method showed significant fluctuation regarding the number of iterations needed for certain tolerance to be met. The zero-division error might occur as I have used computer value in consequence of using "if" conditional for choosing "y0" values if specific x is chosen where the computer might have classed the y_{exact}=-1 less than the -1/e value.

$$\begin{cases} y = \varnothing & a < -1/e \\ y = W_{-1}(-1/e) = W_0(-1/e) = -1 & a = -1/e \\ y_1 = W_{-1}(a) < y_2 = W_0(a) & -1/e < a < 0 \\ y = W_0(a) & a \geq 0 \end{cases}$$

9c:



The intersection point (x = -1/e, y = -1) is where the two branches meet. This can be seen from the fact that for this value of x, both branches have the same y-value of -1. The Lambert function has two branches because it is a multi-valued function, and the two branches meet at the point of intersection where they are equal. The blue branch, W_0(x), is the principal branch with y >= -1 for x >= -1/e, and the red branch, W_{-1}(x), is the second branch with y <= -1 for x <= -1/e.

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.

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

---(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)

```
LambertW(-5*exp(-5)) + 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

$$b = \lambda_{max} T = \frac{hc}{x k_B}$$

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
```