The Lambert Function

Laboratory Exercise in Numerical Methods

Vasily Arzhanov, Nuclear Engineering, KTH

Abstract

The Lambert W function, also known as the omega function or the product logarithm, is a multivalued function occurring in various areas of mathematics and physics. When viewed as a complex-valued function of one complex argument, the Lambert function has infinitely many branches, $W_k(z)$, for each integer k. The function W_0 is known as the principal branch. Two branches, namely $W_0(x)$ and $W_{-1}(x)$, sometimes also denoted [7] as $W_\mathrm{p}(x)$ and $W_\mathrm{m}(x)$, will be enough when working in the real domain, $x \in \mathbb{R}$.

The Lambert W function cannot be given in terms of elementary functions. It is useful to solve various equations involving exponentials for example when finding the maxima of the Plank, Bose-Einstein, and Fermi-Dirac distributions. This function occurs also when solving delay differential equations. It finds its applications in biochemistry and combinatorics.

This laboratory exercise focuses on studying, evaluating, and applying the Lambert function in various situations. While performing this laboratory exercise, the students are supposed to answer the questions marked with **blue** and submit a (short) written report. The layout of the present guidelines may serve as a template for writing the final report.

Key words: Lambert function, principal branch, Halley's method, Wien's displacement law.

Preamble

The Lambert W function often appears when solving various equations involving the unknown variable both in the base and in the exponent, or both inside and outside of a logarithm. As a starting example, consider the following nice-looking equation

$$x^2 = 2^x . \tag{1}$$

Exercise 1 [4p].

(a) **[1p]** Consider Eq.(1). With a sharp eye, suggest an obvious integer solution.
(b) **[1p]** Once more, suggest another obvious integer solution to this equation.
(c) **[1p]** Plot the following two curves, $y = x^2$ and $y = 2^x$, over the interval $-4.5 \le x \le 4.5$ and restrict the y-axis between $-1 \le y \le 20$. By visual inspection, how many solutions can be identified?
(d) **[1p]** Are there other solutions? Explain.

Appreciably, the symbolic toolboxes in both MATLAB and Python know about the third solution and give it in terms of the Lambert W function.

Exercise 2 [2p].

Read about the symbolic toolbox in MATLAB or Python then do the following.
(a) **[1p]** As a warm-up, solve the equation, $x^2 = 2$, in MATLAB or Python symbolically and report the answer.
(b) **[1p]** Solve the equation, $x^2 = 2^x$, in MATLAB or Python symbolically and report the answer.

Introduction

Johann Heinrich Lambert, 1728 – 1777, a Swiss polymath known for the first proof that π is irrational, found a series representation for the solution to the equation $y = y^m + q$. Leonard Euler, 1707 – 1783, converted this equation to a more compact one, $\ln y = cy^a$, which in turn may be further transformed to the standard form of the Lambert function, which is defined as the solution y to the transcendental equation when x is given

$$ye^y = x . \tag{2}$$

Clearly, the solution y depends on x and may be denoted as $y = y(x)$. Thus, by definition, $W(x) \equiv y(x)$. Here, $y(x)$ may also be referred to as the inversion of Eq.(2). Similar to the fundamental logarithmic identity, $b^{\log_b x} = x$, the corresponding principal identity for the Lambert W function, which follows from the very definition, reads as

$$W(x)e^{W(x)} = x$$

Several other obvious identities are immediately deduced, for example,

$$e^{W(x)} = \frac{x}{W(x)},\ e^{-W(x)} = \frac{W(x)}{x} \text{ and } e^{nW(x)} = \left(\frac{x}{W(x)}\right)^n .$$

One more identity turns out to be useful. By definition, $W(x) = y$, where y is a solution to Eq.(2). Setting $x = ye^y$ in the definition, $W(x) = y$, gives the following identity

$$W(ye^y) = y \tag{3}$$

1. Negative Solution

This chapter explains why the symbolic toolbox gives the negative solution to Eq.(1) in terms of the Lambert W function. Applying the square root on both sides of Eq.(1) results in two equations

$$x = \pm 2^{x/2}$$

The plus sign leads to the "positive" equation $x = 2^{x/2}$, which clearly has only positive roots. A visual inspection quickly reveals the two obvious solutions, namely $x = 2$ and $x = 4$, whereas the negative root belongs to the "negative" equation

$$x = -2^{x/2} \tag{4}$$

Figure 1 shows that the "positive" equation (in red) has two positive roots whereas the "negative" equation does have a negative root somewhere in the range $-1 < x < 0$.

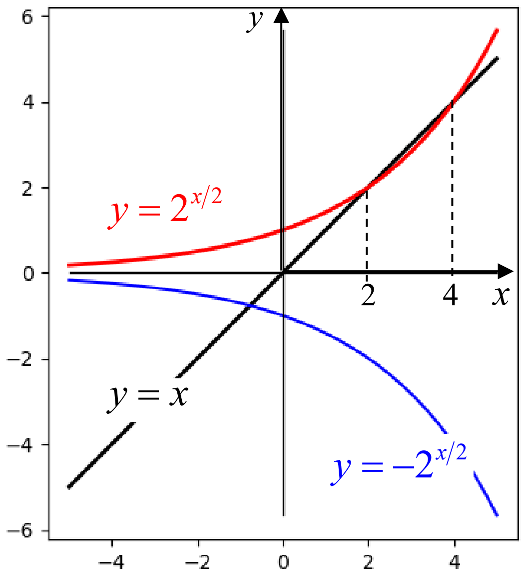

Figure 1. Roots for "positive" (red) and "negative" (blue) equations.

The sign function of a real argument x is a piecewise function that can be defined as follows.

$$\operatorname{sgn} x = \operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

It is often referred to as the signum (Latin for sign) function to avoid confusion with the sine function. The signum function is an odd function satisfying $x = |x| \operatorname{sgn} x$ and $|x| = x \operatorname{sgn} x$.

The two equations, $x = \pm 2^{x/2}$, can now be conveniently written in a compact form as

$$x = \operatorname{sgn}(x) 2^{x/2} \tag{5}$$

This equation is given in the form readily suited for applying the fixed-point iteration, FPI.

Exercise 3 [12p].

Write a computer code that solves Eq.(5) by the fixed-point iteration, $x_{n+1} = \phi(x_n)$, with $\phi(x) = \text{sgn}(x)2^{x/2}$. Set the absolute tolerance, tol = 10^{-4}, and the maximum possible number of iterations to be 50, which prevents your code from eventually entering an infinite loop, then do the following.

(a) **[4p]** How many solutions does Eq.(5) have? Think carefully.
(b) **[1p]** Check the convergence condition for the solutions, $x = 2$ and $x = 4$.
(c) **[3p]** Check the convergence condition for the negative root.
(d) **[1p]** Run FPI with an initial guess, $x_0 > 4$, and report the result.
(e) **[1p]** Run FPI with an initial guess, $x_0 = 4$, and report the result.
(f) **[1p]** Run FPI with an initial guess, $0 < x_0 < 4$, and report the result.
(g) **[1p]** Run FPI with an initial guess, $x_0 < 0$, and report the result.

We proceed to study the negative solution to Eq.(4), which is now rewritten as

$$-x = 2^{x/2}$$

Using the fundamental logarithmic identity, $2 = e^{\ln 2}$, the above equation transforms as

$$-x = 2^{x/2} = e^{x \cdot \ln 2/2}$$

Setting, $x \cdot \ln 2/2 \equiv -y$, gives $-x = y \cdot 2/\ln 2 = e^{-y}$, which is finally converted to

$$ye^y = \ln 2/2$$

By definition, the solution is given by $y = W\left(\ln 2/2\right)$, which finally leads to the answer

$$x = -\frac{2}{\ln 2} W\left(\frac{\ln 2}{2}\right) \tag{6}$$

Strangely enough, we still do not know, what the $W(x)$ function is, but were able to express the negative solution to Eq.(1) in terms of $W(x)$. In particular, we still do not know if $W(x)$ is at all defined at ln2/2.

2. Domain and Range of the Lambert W Function

As was defined earlier, the Lambert W function is an inverse of the function $x(y) = ye^y$, which can logically be denoted as $y(x)$, see also Eq.(2). Simply plotting the inverse $y(x)$ might well offer a first insight as to the character and extent of the Lambert function. We just recall that, by definition, $W(x) = y(x)$.

The following simple example illustrates how we can plot an inverse function without actually inverting the direct function. Let, for instance, plot the natural logarithmic function, $y = \ln x$, treating it as an inverse to the exponential function, $x = e^y$. To this end, we can do the following.

1) Cover a specified interval with equidistant points, for example as `y = linspace(-2.5,1.7)`.
2) Generate the corresponding x-values, for example, `x = exp(y)`.
3) Plot the exponential function, for example, `plot(y,x)`.
4) Plot the logarithmic function as `plot(x,y)`.

The result is illustrated in the next figure.

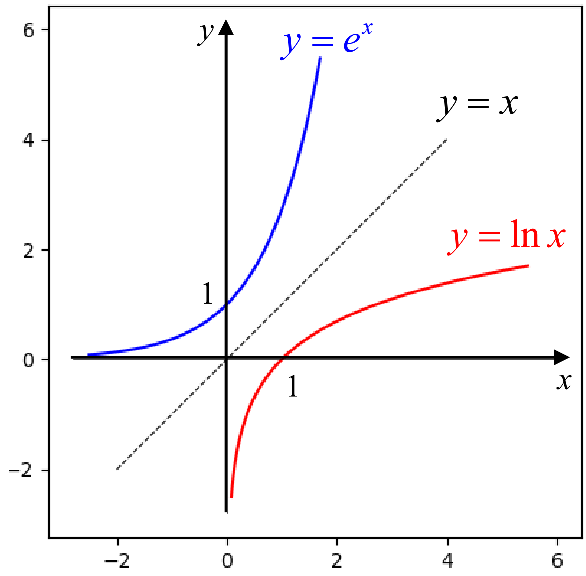

Figure 2. Plotting $\ln(x)$, in red, as inverse of $\exp(x)$, in blue.

Great, this little programming trick works well. Let's continue in the same spirit. As a first glance at the Lambert W function, do Exercise 4.

Exercise 4 [3p].

Using the ideas demonstrated previously in items 1) to 4), plot in one and the same window the direct function, $x = f(y) \equiv ye^y$, and its inverse $y = f^{-1}(x)$, which is expected to be the Lambert function $y = W(x)$. To this end,

(a) **[1p]** Cover the y-interval [-1.5, 0.4] with 129 equidistant points and plot the direct function in the blue colour.
(b) **[1p]** In the same window, plot the inverse function in the red colour.
(c) **[1p]** Plot the bisector, $y = x$, as a black broken line and set the aspect ratio to one.

Looking at the red curve, which is supposed to be the Lambert function, it is suspected that the red line cannot represent a function at all because there are numbers c around -0.2 for which the vertical line, $x = c$, crosses the red curve in two points, which in turn is in fundamental contradiction to the concept of a function. It is recalled, in mathematics, a function maps each element in the domain to exactly one element in the co-domain. To clarify the situation, we plot the red curve in a separate window and select different limits for the x- and y-axis. The next figure confirms our fears.

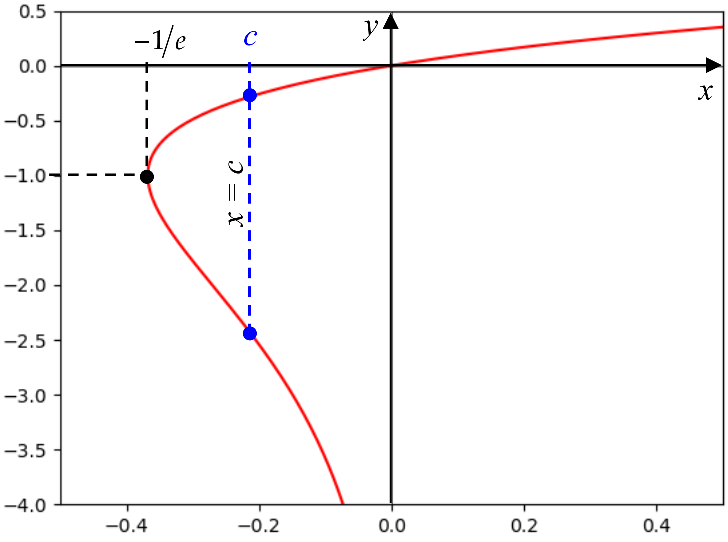

Figure 3. The red curve magnified.

The revealed situation suggests that the Lambert W function is a multi-valued function i.e., composed of several, at least two, branches. It also calls for an accurate definition of the domain and co-domain for the Lambert function. In general, the domain of a function $f(x)$, denoted as dom(f), is the set of all values x for which the function is defined, and the co-domain (also range or image) of the function is the set of all values that $f(x)$ attains.

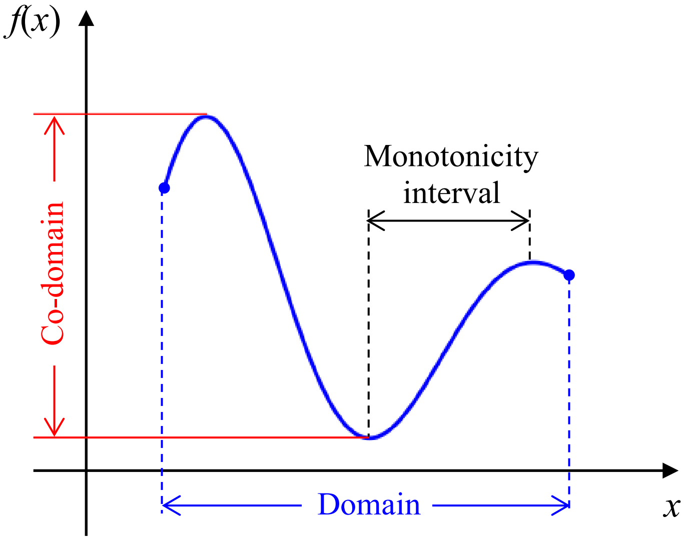

Figure 4. Domain and Co-domain (Range) of a function.

Definition: A function $f(x)$ is said to be strictly monotonically increasing if for any $x_1 < x_2$, it holds $f(x_1) < f(x_2)$. Accordingly, a strictly monotonically decreasing function is defined.

A function with either property (either strictly increasing or decreasing) is called strictly monotonic. Functions, that are strictly monotonic, are one-to-one and hence each such function has its inverse typically denoted as $f^{-1}(x)$. One way of studying monotonicity properties is through the derivative. To find monotonicity regions, or intervals, for the "direct" function, $x(y) = ye^y$, do the next exercise.

Exercise 5 [11p].

It is recalled, the Lambert function is defined as an inverse to the "direct" function $x(y) = ye^y$. Investigate this function in detail. With this in mind, carry out the following.

(a) **[2p]** Calculate the derivative analytically and find the regions (or equivalently intervals) where $x'(y) > 0$, $x'(y) = 0$, and $x'(y) < 0$.

(b) **[3p]** Determine the (strict) monotonicity regions, $y \in R_k$, for the function $x(y)$. Label the regions where $x(y)$ is increasing with non-negative subscripts beginning with $k = 0$. Label the regions of decreasing $x(y)$ with negative subscripts beginning with, $k = -1$. Deduce how many branches the real argument Lambert function W has.

(c) **[2p]** Find the minimum value, $x_{\min}$, and the corresponding argument, $y_{\min}$, $x_{\min} = x(y_{\min})$.

(d) **[2p]** Determine the domain and the co-domain for each branch of the Lambert function. Select one of the branches to be the principal (main) branch and denote it as $W_0(x)$. Denote and label the complementary branch(es) accordingly.

(e) **[2p]** Plot the function, $x(y) = ye^y$, and label appropriately characteristic regions and values.

3. Newton's Method

Many algorithms for solving the non-linear equation, $ye^y = x$, in y may be applied to evaluate numerically the Lamber function, $W(x) = y$. We start with Newton's method, also known as Newton-Raphson, because of its quadratic convergence, which is considered to be very fast. To this purpose, we first transform the equation to its standard form with x is fixed

$$f(y) = ye^y - x = 0 \tag{7}$$

Newton's method starts with an initial guess, y_0, and then generates, hopefully, a converging number sequence

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

The derivative is found easily $f'(y) = (1+y)e^y$, which casts Newton's method to the explicit form

$$y_{n+1} = y_n - \frac{y_n e^{y_n} - x}{(1+y_n)e^{y_n}} = y_n - \frac{y_n - xe^{-y_n}}{1+y_n} \tag{8}$$

It is instructive to calculate also the second derivative, $f''(y) = (2+y)e^y$, which vanishes when $y = -2$ i.e., $f''(-2) = 0$, hence, this is a point of inflection, where the curvature changes its sign. When $x \geq 0$, there is only one solution to Eq.(7) as contrast to the interval, $-1/e < x < 0$, for which there exist two solutions as Figure 3 suggests. Therefore, it is of paramount importance to find conditions under which Newton's algorithm converges to one root or to the other. A small numerical experiment might be helpful here.

We note first, when $y = -1/8$, the expression ye^y evaluates to $-1/8e^{-1/8}$ hence if we set the constant x to this value, $x = -1/8e^{-1/8} \approx -0.11$, one of the solutions to the equation, $ye^y - x = 0$, is known, namely $y = -1/8 = -0.125$. Clearly, x belongs to the interval, $-0.37 \approx -1/e < x < 0$. Next, to clarify the situation, we display a plot of the function, $f(y) = ye^y - x$, in Figure 5. As expected, one root, or zero, is $-1/8$, denoted as w_0 (belongs to the branch W_0). The other one is denoted as w_{-1} (belongs to the branch W_{-1}). It is noted, $f'(y) < 0$ when $y < -1$; and $f'(y) > 0$ when $y > -1$. In other words, a tangent line at y has a negative slope when $y < -1$, and a positive slope when $y > -1$. Thus, selecting $w_{-1} < y_0 \le -2$ produces the first iterate y_1, Eq.(8), such that $w_{-1} < y_1 < y_0$ and so on i.e., we move monotonically to the root w_{-1}.

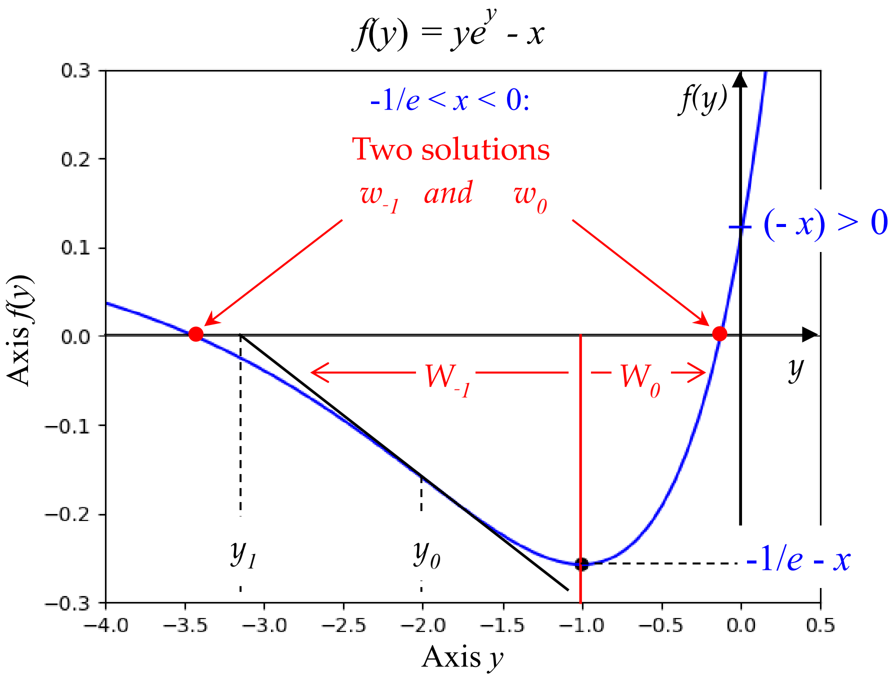

Figure 5. Two roots of the equation $f(y) = ye^y - x = 0$

More detailed analysis reveals that Newton's method converges to the left root, w_{-1}, when the starting approximation (guess) belongs to some interval $[a,b]$ such that, $b \le y_0 \le a < -1$, and on contrary, Newton's method converges to the right root, w_0, when $y_0 > -1$. Somewhat simplified, it is recommended to use $y_0 = 1$ for evaluating the principal branch, $W_0(x)$, when $-1/e < x < \infty$; and start Newton's method with $y_0 = -2$ for evaluating the complementary branch, $W_{-1}(x)$, when $-1/e < x < 0$. In the latter case, it is recalled that $y_0 = -2$ is a point of inflection representing a kind of characteristic (average) slope coefficient in the region, $-\infty < y < -1$. It is noted the tangent line at $y_0 = -2$ in Figure 5 is below the plot when $-2 < y < -1$, and just the opposite, the tangent line is above the plot when $-\infty < y < -2$. The above observation summarizes as follows.

$$y_0 = y_0(x) = \begin{cases} 1: & y_n \to W_0(x) \quad \text{for } -1/e < x < \infty \\ -2: & y_n \to W_{-1}(x) \quad \text{for } -1/e < x < 0 \end{cases} \tag{9}$$

It should be stressed the abscissa $x = -1/e$ is a critical point, where the two branches meet each other, as is shown in Figure 3

$$W_0(-1/e) = W_{-1}(-1/e) = -1.$$

This is a direct consequence of identity (3) when setting $y = -1$. The next Exercise 6 is designed to verify some of the above observations.

Exercise 6 [12p].

Write a computer code that implements Newton's method for solving the general non-linear equation, $f(x) = 0$, when the function and its derivative are known analytically. A template for a general Newton-Raphson solver is found in the Appendix. Then do the following.

(a) **[2p]** As a warm-up find $\sqrt{2}$ using the starting value, $x_0 = 1$, and the absolute tolerance, tol = 10^{-3}. Report the number of iterations and the actual absolute error.

(b) **[2p]** Find numerically ln2 by noting that when $y = \ln 2$, it holds $ye^y = 2\ln 2$. In other words, when $x = 2\ln 2$, the solution to the Lambert equation, $ye^y = x$, is $y = \ln 2$. To this end, set the initial guess, $y_0 = 1$, and the absolute tolerance, tol = 10^{-3}. Report the number of iterations and the actual absolute error.

(c) **[2p]** Note that when $y = -1/2$, the function, ye^y, evaluates to $x = -0.5/\sqrt{e}$. Run your computer code with this x-value and the absolute tolerance, tol = 10^{-3} using the initial guess, $y_0 = 1$. Report the number of iterations and the actual absolute error.

(d) **[3p]** Find numerically the second solution starting with, $y_0 = -2$.

(e) **[3p]** Plot the function $x(y) = ye^y$ in blue using the horizontal axis for y and the vertical axis for x. Plot the x- and y-axes in black. Plot also the horizontal line, $x = -0.5/\sqrt{e}$, in red. Using the scatter the function in Matlab or Python, plot two markers at the location of the first solution i.e., at $\left(y = -1/2, x = -0.5/\sqrt{e}\right)$, and at the location of the second solution; use black colour. Make sure the markers display the position where the horizontal line cuts the curve of the function $x(y) = ye^y$. It is recommended to limit the y-variable as `xlim([-2.5, 0.25])` and to limit the x-variable as `ylim([-0.4, 0.1])`. It is recalled that the role of the x- and y-axes is interchanged.

4. Halley's Method

Generally speaking, Newton's method generates iterations that approach a zero of a nonlinear function with the second order of convergence, which is considered to be very fast. It is tempting to ask if there are even faster root-finding algorithms. The answer is positive, for example Halley's method, whose order of convergence is three. The method is named after Edmond Halley, 1656 – 1742, the British astronomer, who is better known for discovering a comet.

Halley's algorithm comes second after Newton's one in the class of Householder's methods. Conceptually, Halley's approach is to find the roots of a linear-to-linear Padé approximation to the function in question as compared to the Secant or Newton technique, which approximate the function linearly, or Muller's method that approximates the function quadratically. In simpler words, Halley's algorithm replaces the function with a hyperbola having the same slope and curvature and then it steps to the nearest zero of this approximation.

The class of Householder's methods consists of root-finding algorithms used for functions of one real variable with continuous derivatives up to some order $d + 1$. Each such method is characterized by the number d which is referred to as the order of the method. The algorithm is iterative and has an order of convergence of $d + 1$. For a fixed positive integer $d \geq 1$, the method is used for solving the nonlinear equation, $f(y) = 0$. Starting with an initial guess, y_0, the algorithm generates a sequence of iterations

$$y_{n+1} = y_n + d\frac{\left(1/f\right)^{(d-1)}(y_n)}{\left(1/f\right)^{(d)}(y_n)}.$$

Here, $f^{(n)}(x)$ stands for the n-*th* derivative of the function $f(x)$ that is

$$f^{(0)}(x) = f(x),\ f^{(1)}(x) = f'(x), \text{ and } f^{(2)}(x) = f''(x).$$

Setting the starting value for y_0 might pose some problem. The simplest solution is our observation expressed in Eq.(9) i.e., to use

$$y_0 = y_0(x) = \begin{cases} 1: & \text{for } W_0(x) \\ -2: & \text{for } W_{-1}(x) \end{cases} \tag{10}$$

More accurate formulas are suggested by several authors. They are summarized as follows. When calculating the principal branch $W_0(x)$, it is recommended to set the initial value y_0 as dependent on x:

$$y_0 = y_0(x) = \begin{cases} -1 & x = -1/e \\ \dfrac{ex}{1+ex+\sqrt{1+ex}}\log\left(1+\sqrt{1+ex}\right) & -1/e < x < 0 \\ x/e & 0 \le x \le e \\ \log(x) - \log\left(\log(x)\right) & e < x < \infty \end{cases} \tag{11}$$

When calculating the complementary branch $W_{-1}(x)$, it is recommended to set the initial value y_0 as dependent on x:

$$y_0 = y_0(x) = \begin{cases} -1 & x = -1/e \\ -1-\sqrt{2(1+ex)} & -1/e < x \le -1/4 \\ \log(-x) - \log\left(-\log(-x)\right) & -1/4 < x < 0 \end{cases} \tag{12}$$

Halley's method becomes particularly efficient if the ratios $f(y)/f'(y)$ and $f''(y)/f'(y)$ happen to take a simple form.

Exercise 7 [18p].

Derive an explicit form of Newton's and Halley's methods for the general nonlinear equation $f(y) = 0$ assuming existence of suitably many continuous derivatives of $f(y)$ and adapt Halley's method to the Lambert equation, $ye^y = x$. With this purpose, do the following.

(a) **[4p]** Derive explicitly Newton's method for the general nonlinear equation, $f(x) = 0$, by recalling that it is a Householder's method of the first order, $d = 1$.

(b) **[6p]** Derive explicitly Halley's method for the general nonlinear equation, $f(x) = 0$, by recalling that it is a Householder's method of the second order, $d = 2$.

(c) **[3p]** Transform the Lambert equation, $ye^y = x$, to the general form, $f(y) = 0$, and calculate explicitly the ratios $f(y)/f'(y)$ and $f''(y)/f'(y)$.

(d) **[5p]** Formulate an explicit version of Halley's method that is adapted to the Lambert equation. Arrange emerging algebraic expressions in terms of suitable ratios such that the final formula takes a simple form.

The next exercise aims at developing a computer code that evaluates the Lambert W function with Halley's method.

Exercise 8 [8p].

Write a computer code that implements Halley's method for computing the two branches, $W_0(x)$ and $W_{-1}(x)$, of the Lambert function. Note the argument y in $f(y)$, which is consistent with the previous notation, and x plays a role of a parameter. To this end, do the following.

(a) **[2p]** Write a computer code that evaluates the Lambert function, $W(x)$, at a given point x. The input arguments are supposed to be as follows:

- `x`: for the argument of the Lambert W function.
- `branch` = 0 for selecting $W_0(x)$ and
  = -1 for selecting $W_{-1}(x)$.
- `tol`: for absolute tolerance i.e., stopping when $|y_{n+1} - y_n| \le \text{tol}$.
- `init` = 0 for selecting not optimized initial value y_0, Eq.(11) and
  = 1 for selecting optimized initial value y_0, Eqs.(12) or (13).

(b) **[1p]** Set x to be the only required argument and the rest to be optional with the following default values: `branch = 0`, `tol = 1e-8`, `init = 1`.

(c) **[1p]** The code must return two values, `y` = $W(x)$ and the number of iterations, say `it`.

(d) **[2p]** The code must check the input arguments for consistency i.e., that x belongs to the domain of the selected branch; the tolerance is a positive and reasonably small number, say 0 < `tol` < 10^{-1}; `init` is either 0 or 1.

(e) **[1p]** Define an internal parameter that restricts the maximum number of iterations, say `itMax = 30`. This measure prevents the code from eventually entering an infinite loop. Issue a warning if the required tolerance is not achieved within the allowed maximum number of iterations.

(f) **[1p]** Include your code in an appendix of your report; evaluate the omega constant using an absolute tolerance of 10^{-8} and compare it with the value found in Wikipedia.

The omega constant is defined as

$$\Omega e^{\Omega} = 1$$

It can also be regarded as the solution to the following equations

$$\Omega = e^{-\Omega} \text{ or } \Omega = -\ln\Omega$$

Hence, the omega constant, Ω, may be characterized as an invariant point of the functions e^{-x} and -ln$x$. It is easy to show that the fixed-point iteration, $\Omega_{n+1} = e^{-\Omega_n}$, converges, which also characterizes the omega constant as a point of attraction (an attractive fixed point) of the function e^{-x}. The omega constant is listed in the On-Line Encyclopedia of Integer Sequences (OEIS) as sequence A030178. It appears in several identities, for example

$$\int_{-\infty}^{\infty} \frac{\mathrm{d}t}{\left(e^t - t\right)^2 + \pi^2} = \frac{1}{1+\Omega}.$$

Other representations are as follows

$$\Omega = \frac{1}{\pi}\mathrm{Re}\int_0^{\pi} \ln\frac{e^{e^{it}} - e^{-it}}{e^{e^{it}} - e^{it}}\mathrm{d}t$$

and

$$\Omega = \frac{1}{\pi}\mathrm{Re}\int_0^{\pi} \ln\left(1 + \frac{\sin t}{t} e^{t\cot t}\right)\mathrm{d}t.$$

The next exercise is designed to deepen the theoretical knowledge in Halley's method.

Exercise 9 [18p].

Set an "exact" (suggested) value for y; evaluate $x = ye^y$ then solve the Lambert equation, $ye^y = x$, for y using Newton's or Haley's methods with an absolute tolerance of tol = 10^{-5}. In the end, calculate the absolute error i.e., the difference between the exact and computed solutions. Finally, fill out Tables 1 and 2. More in detail, do the following.

(a) **[8p]** Fill out **Table 1** in the suggested format for the branch $W_0(x)$ and make a comparative comment about efficiency of Newton's and Halley's methods with simplified and optimized starting values.

(b) **[8p]** Fill out **Table 2** in the suggested format for the branch $W_{-1}(x)$ and make a comparative comment about efficiency of Newton's and Halley's methods with simplified and optimized starting values.

(c) **[2p]** Using your code, plot the branch $W_0(x)$ in blue and the branch $W_{-1}(x)$ in red. Add separately a black marker at the location ($x = -1/e$, $y = -1$). Explain why this position is the point where the two branches meet each other. In the plot, show explicitly the domain and co-domain for each branch.

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	Iter	Method		Err	Iter
Newton	$y_0 = 1$	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Newton	y_0 = opt	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Halley	$y_0 = 1$	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Halley	y_0 = opt	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10

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	Iter	Err	Iter	Err	Iter
Newton	$y_0 = -2$	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Newton	y_0 = opt	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Halley	$y_0 = -2$	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10
Halley	y_0 = opt	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10	1.23×10^{-10}	10

5. Applications

An article in the issue of FOCUS [2], the newsletter of the Mathematical Association of America, advocates the Lambert W function as a promising candidate for a new elementary function worth of studying in schools and to be included in textbooks. The article argues; this function is radically different from traditional elementary functions of polynomials, rationals, exponentials, logarithmics and trigonometrics. Moreover, its calculus leads to many interesting and powerful applications. Some of them are found in theoretical and statistical mechanics; in atomic, nuclear, and optical physics, in general relativity, and quantum chromodynamics.

From a mathematical point of view, the Lambert W function appears to be very useful when solving transcendental equations in which the unknown is found both outside and inside an exponential or logarithmic function. Except for special cases, such equations cannot be solved explicitly in terms of algebraic combinations of exponentials and logarithms, for example, the equation we started with i.e., Eq.(1). Other elementary examples are

$$2x+3=e^{x} \text{ or } x=\ln(3x).$$

The general strategy to solve such equations is to cast them to the standard form

$$ye^{y}=a.$$

In the most of physical applications, we can restrict ourselves to the real variables and numbers. Note a slight change in the notation; the character a is now used instead of the variable x to stress it is a constant typically composed of physical coefficients and parameters of the problem in question. By definition, the answer is then given as $y = W(a)$. Another useful way of thinking is to recall the identity, $y=W\left(ye^{y}\right)$. Then the solution of the standard equation "appears" naturally

$$y=W\left(ye^{y}\right)=W(a).$$

However, care must be taken to write down the answer. There can be no solution, one or two solutions depending on the value of a. The situation is summarized as follows.

$$\begin{cases} y=\varnothing & a<-1/e \\ y=W_{-1}(-1/e)=W_{0}(-1/e)=-1 & a=-1/e \\ y_{1}=W_{-1}(a)<y_{2}=W_{0}(a) & -1/e<a<0 \\ y=W_{0}(a) & a\geq 0 \end{cases} \tag{13}$$

Possibly, the first problem in physics that was solved explicitly in terms of the Lambert W function, was one in which the exchange forces between two nuclei of the hydrogen ion H_2^+ were calculated [4]. Several other examples are related to viscous flow, generalized Gaussian noise, solar wind, crystal growth, black holes, general relativity, exact solutions of the Schrödinger equation, quantum chromodynamics, statistical mechanics, chemical engineering, epidemiology, Stirling's formula for $n!$, Bernoulli numbers and Todd genus, enumeration of trees in combinatorics, water-wave heights in oceanography etc.

D. Kalman gives an interesting counterpart of the Lambert W function in [5], where he defines a (generalized logarithmic) function, $y = \text{glog}(x)$, as the solution to the equation

$$e^{y}/y=x.$$

The glog function is very similar to W(x), possessing similar properties and useful common applications. It is easy to show that the two functions are intimately related by the identities

$$W(x)=-\text{glog}(-1/x) \text{ and } \text{glog}(x)=-W(-1/x).$$

6. Wien's Displacement Law

The law is named after Wilhelm Wien (1864–1928), a German physicist, who derived it in 1893 based on a thermodynamic argument. Wien's displacement law states that the black-body radiation for different temperatures will peak at different wavelengths that are inversely proportional to the temperature

$$\lambda_{\max} = b/T\,.$$

Here, T is the absolute temperature and b is a constant of proportionality commonly referred to as Wien's wavelength displacement law constant. The National Institute of Standards and Technology (NIST) gives its numerical value [6]

$$b = 2.897771955\times 10^{-3}\,\mathrm{m\cdot K}$$

Wien's displacement law can easily be deduced from Plank's law, which states that every physical body spontaneously and continuously emits electromagnetic radiation moreover the energy E radiated per unit volume and per unit wavelength interval by a cavity of a black body is given by Planck's spectral distribution

$$E(\lambda,T) = \frac{8\pi hc/\lambda^5}{e^{hc/(\lambda kT)}-1} \tag{14}$$

Exercise 10 [12p].

Derive Wien's displacement law and find an analytical expression for Wien's displacement constant b starting with Plank's spectral distribution. Use the NIST Reference on Constants [6] as a database of the most accurate physical data. In steps, do the following.

(a) **[4p]** Assuming a fixed temperature T, derive an equation for finding $\lambda_{\max}$.
(b) **[2p]** Suggest a substitution that transforms the derived equation to a compact transcendental equation.
(c) **[3p]** Find all the solutions (roots) of the derived compact equation.
(d) **[3p]** Give an analytic expression for Wien's displacement constant b and calculate its numerical value with 10 decimal places.

References

[1] Corless R.M., Gonnt G.H., Hare D.E.G., Jeffrey D.J., Knuth D.E., *On the Lambert W Function*, Adv. Comput. Math. 5, 329-359 (1996).
[2] Gouvea F., Ed., *Time for a New Elementary function?*, FOCUS (Newsletter of Mathematics Association of America), vol. 20, p. 2 (2000).
[3] Dence T.P., *A Brief Look into the Lambert W Function*, Applied Mathematics, vol. 4, p. 887-892, (2013).
[4] Scott T.G., Babb J.F., Dalgarno A. And Morgan III J.D. *Resolution of a paradox in the calculation of exchange forces for* H_2^+, Chemical Physics Letters 203, 175–183 (1993).
[5] Kalman D., *A Generalized Logarithm for Exponential Linear Equations*, The College Mathematical Journal, 32, 1, 2–14 (2001).
[6] NIST Reference on Constants, Units, and Uncertainty, https://physics.nist.gov/cuu/Constants/ (2022-12-04).
[7] NIST Digital Library of Mathematical Functions, https://dlmf.nist.gov/ , (2022-12-08)

Appendix

Templates for non-linear solvers

Fixed Point Interation

```
def fpi(phi,x0,tol=1e-4,itMax=50):
   d = float('inf'); it = 0;
   while (abs(d) > tol) & (it < itMax):
      it = it + 1
      x  = phi(x0)
      d  = x - x0
      x0 = x
   if it == itMax:
      print('\nNo convergence within itMax = %2d  x = %g
             phi(x)  = %g\n'%(itMax,x,phi(x)))
   return x,it
```

Newton-Raphson algorithm

```
def NewRap(f,fp,x0,tol=1e-8,itMax=30):
   d = float('inf')
   x = x0; it = 0;
   while (abs(d)>tol) & (it<itMax):
      it = it + 1
      d  = f(x)/fp(x)
      x  = x - d
   if it == itMax:
      print('\nNewRap: No convergence within itMax = %d\n'%itMax)
   return x,it
```