Numer Algor (2011) 57:27–34
DOI 10.1007/s11075-010-9409-6

ORIGINAL PAPER

The Lambert W-functions and some of their integrals: a case study of high-precision computation

Walter Gautschi

Received: 18 February 2010 / Accepted: 27 June 2010 /
Published online: 16 July 2010

Abstract The real-valued Lambert W-functions considered here are $w_0(y)$ and $w_{-1}(y)$, solutions of $we^w = y, -1/e < y < 0$, with values respectively in $(-1, 0)$ and $(-\infty, -1)$. A study is made of the numerical evaluation to high precision of these functions and of the integrals $\int_1^\infty [-w_0(-xe^{-x})]^\alpha x^{-\beta} \mathrm{d}x, \alpha > 0, \beta \in \mathbb{R}$, and $\int_0^1 [-w_{-1}(-xe^{-x})]^\alpha x^{-\beta} \mathrm{d}x, \alpha > -1, \beta < 1$. For the latter we use known integral representations and their evaluation by nonstandard Gaussian quadrature, if $\alpha \neq \beta$, and explicit formulae involving the trigamma function, if $\alpha = \beta$.

Keywords Lambert W-functions · Integrals of Lambert W-functions · Nonstandard Gaussian quadrature · Variable-precision computation

Mathematics Subject Classifications (2010) 33B99 · 33F05 · 65D20 · 65D30

1 Introduction

The (real-valued) Lambert W-functions are solutions of the nonlinear equation

$$we^w = y, \quad y \in \mathbb{R}. \tag{1.1}$$

If $y > 0$, there is a unique real solution, $w(y)$, satisfying $0 < w(y) < \infty$. If $-1/e \leq y < 0$, there are exactly two real solutions, $w_0(y)$ and $w_{-1}(y)$, satisfying respectively $-1 \leq w_0(y) < 0$ and $-\infty < w_{-1}(y) \leq -1$. Clearly, $w(0+) = 0$, $w_0(0-) = 0$, and $w_{-1}(0-) = -\infty$, while $w_0(-1/e) = w_{-1}(-1/e) = -1$. For

W. Gautschi (✉)
Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398, USA
e-mail: wxg@cs.purdue.edu

$y < -1/e$, there are no real solutions of (1.1). For a discussion of the various branches of the Lambert W-functions, also in the complex plane, see [3].

We note that

$$-w_0(-xe^{-x}) \equiv x \quad \text{for } 0 \le x \le 1, \tag{1.2}$$

and likewise

$$-w_{-1}(-xe^{-x}) \equiv x \quad \text{for } 1 \le x < \infty. \tag{1.3}$$

Indeed, by definition,

$$-xe^{-x} \equiv w_0(-xe^{-x})e^{w_0(-xe^{-x})}, \quad 0 \le x \le 1.$$

This identity remains valid if $w_0(-xe^{-x})$ at both occurrences is replaced by $-x$, which is in $[-1, 0]$. By uniqueness of w_0, there follows (1.2). The identity (1.3) is proved similarly.

Our interest here is in the computation (to high precision) of the three Lambert functions and of the integrals

$$I_{0,[1,\infty]}(\alpha,\beta) = \int_1^\infty f_0(x;\alpha,\beta)\mathrm{d}x \quad \text{and} \quad I_{1,[0,1]}(\alpha,\beta) = \int_0^1 f_1(x;\alpha,\beta)\mathrm{d}x, \tag{1.4}$$

where

$$\begin{aligned} f_0(x;\alpha,\beta) &= [-w_0(-xe^{-x})]^\alpha x^{-\beta},\ \alpha > 0; \\ f_1(x;\alpha,\beta) &= [-w_{-1}(-xe^{-x})]^\alpha x^{-\beta},\ \alpha > -1, \beta < 1. \end{aligned} \tag{1.5}$$

Both integrals present numerical difficulties because of singularities of the integrands at the upper resp. lower end point of integration.

For $\alpha = \beta$, the integrals are explicitly known [8],

$$\begin{aligned} I_{0,[1,\infty]}(\alpha,\alpha) &= \alpha\psi_1(\alpha) - 1,\ \alpha > 0; \\ I_{1,[0,1]}(\alpha,\alpha) &= \alpha\psi_1(1-\alpha) + 1,\ |\alpha| < 1, \end{aligned} \tag{1.6}$$

where ψ_1 is the trigamma function. Their sum equals

$$\alpha(\psi_1(\alpha) + \psi_1(1-\alpha)) = \alpha\left[\frac{\pi}{\sin(\alpha\pi)}\right]^2, \tag{1.7}$$

by the reflection formula for ψ_1 (cf. [1, Eq. 6.4.7]). The Matlab routines[1] `sI01infaa.m`[2] and `sI101aa.m`[2] implement (1.6) in variable-precision arithmetic.

[1] All Matlab routines referred to in this paper can be downloaded from the web site http://www.cs.purdue.edu/archives/2002/wxg/codes/LAMBERTW.html. They make use of additional routines in the packages `OPQ`, `SOPQ` found on the same web site.

[2] This routine requires Matlab Version 7.8 (R2009a) or later.

If $\alpha \neq \beta$, then [8]

$$I_{0,[1,\infty]}(\alpha,\beta) = \frac{1}{\alpha-\beta+1}\left[-1+\alpha\int_0^1 u^{\alpha-1}\left(\frac{\ln(1/u)}{1-u}\right)^{\alpha-\beta+1}\mathrm{d}u\right], \quad \alpha > 0, \tag{1.8}$$

which for $\alpha = \beta$ reduces to (1.6) in view of [7, Eq. 4.251.4]. Similarly [8],

$$I_{1,[0,1]}(\alpha,\beta) = \frac{1}{\alpha-\beta+1}\left[1+\alpha\int_1^\infty u^{\alpha-1}\left(\frac{\ln u}{u-1}\right)^{\alpha-\beta+1}\mathrm{d}u\right], \; \alpha > -1, \; \beta < 1. \tag{1.9}$$

Both formulae lend themselves to numerical evaluation by appropriate (non-standard) Gaussian quadrature; see Sections 3, 4 for details.

Although the Lambert functions will not be used explicitly in what follows, it may be of interest to briefly consider computational methods for their evaluation. This is done in Section 2.

2 Computing the Lambert W-functions

There are of course many possible ways of solving the equation (1.1).[3] A simple and generally reliable method is Newton's method which, by choosing the initial approximations judiciously, allows us to compute all three Lambert functions defined in Section 1 (only the last two of them being of interest here). For y near and above $-1/e$, Newton's method, however, suffers from loss of accuracy and consequent slow, or even lack of, convergence. In this case a power series expansion method is proposed. For y near and below zero, Newton's method for w_{-1}, while numerically stable, may take many iterations (some 30 in Matlab double precision, when $y = -10^{-10}$) to converge.

2.1 Newton's method

The Newton iteration for (1.1) is

$$w^{[\nu+1]} = \frac{(w^{[\nu]})^2 + ye^{-w^{[\nu]}}}{1+w^{[\nu]}}, \qquad \nu = 0, 1, 2, \ldots, \tag{2.1}$$

where as initial value $w^{[0]}$ we take

$$w^{[0]} = \begin{cases} y & \text{if } 0 < y < e, \\ \ln(y/\ln y) & \text{if } y \geq e \end{cases} \tag{2.2}$$

for $w(y)$, and

$$w^{[0]} = \begin{cases} 0 & \text{for } w_0(y), \\ -2 & \text{for } w_{-1}(y). \end{cases} \tag{2.3}$$

[3] For a recent discussion of numerical methods, see [2].

The choice in (2.2), when $y \geq e$, is motivated by the asymptotic behavior of $w(y)$ as $y \to \infty$, and the choice in (2.3) by our desire to have monotone convergence. The latter, in theory, is guaranteed (cf. [5, Example 6.4, p. 233]) by the convexity/concavity properties of the function we^w and the fact that $w = -2$ is an inflection point of the curve $y = we^w$.

Near the point $(w, y) = (-1, -1/e)$, where the graph of $y = we^w$ has a horizontal tangent, Newton's method converges only linearly and may even fail to converge because of cancellation errors in the denominator of (2.1). Indeed, $w^{[\nu]}$ comes arbitrarily close to -1 when y approaches $-1/e$ from above. To avoid this difficulty, we use an appropriate power series expansion (cf. Section 2.2).

2.2 Power series solution

Let

$$y = -\frac{1}{e} + x^2, \quad x > 0. \tag{2.4}$$

Then the solution w of (1.1) admits an expansion in powers of x,

$$w = -1 + c_1 x + c_2 x^2 + c_3 x^3 + \cdots . \tag{2.5}$$

Matching the power series of we^w against the (finite) series (2.4), we find

$$c_1 = \pm\sqrt{2e}, \tag{2.6}$$

and, with the help of Maple, that

$$\begin{aligned}
c_2 &= -\frac{1}{3}\, c_1^2, \\
c_3 &= -\left(c_2 c_1^2 + \frac{1}{8}\, c_1^4 + \frac{1}{2}\, c_2^2\right) / c_1, \\
c_4 &= -\left(c_3 c_2 + c_3 c_1^2 + c_1 c_2^2 + \frac{1}{2}\, c_2 c_1^3 + \frac{1}{30}\, c_1^5\right) / c_1, \\
c_5 &= -\left(c_4 c_2 + c_4 c_1^2 + \frac{1}{2}\, c_3 c_1^3 + \frac{3}{4}\, c_2^2 c_1^2 + \frac{1}{6}\, c_2 c_1^4\right. \\
&\qquad \left. + \frac{1}{2}\, c_3^2 + \frac{1}{3}\, c_2^3 + \frac{1}{144}\, c_1^6 + 2 c_1 c_2 c_3\right) / c_1, \\
c_6 &= -\left(2 c_4 c_1 c_2 + \frac{3}{2}\, c_3 c_2 c_1^2 + \frac{1}{840}\, c_1^7 + c_5 c_2 + c_4 c_3\right. \\
&\qquad + c_5 c_1^2 + \frac{1}{2}\, c_4 c_1^3 + c_1 c_3^2 + c_3 c_2^2 + \frac{1}{6}\, c_3 c_1^4 \\
&\qquad \left. + \frac{1}{2}\, c_1 c_2^3 + \frac{1}{3}\, c_2^2 c_1^3 + \frac{1}{24}\, c_2 c_1^5\right) / c_1, \\
&\cdots \qquad\qquad \cdots
\end{aligned} \tag{2.7}$$

Clearly, for $w = w_{-1}$ we must select the minus sign in (2.6), and for $w = w_0$ the plus sign. Substituting (2.6) in (2.7), we then get, again with the help of Maple,

$$w_{-1}(x) = -1 - \sqrt{2e}\,x - \frac{2}{3}\,e\,x^2 - \frac{11}{36}\sqrt{2e^3}\,x^3 - \frac{43}{135}\,e^2\,x^4 \\ - \frac{769}{4320}\sqrt{2e^5}\,x^5 - \frac{1768}{8505}\,e^3\,x^6 + \cdots \tag{2.8}$$

and

$$w_0(x) = -1 + \sqrt{2e}\,x - \frac{2}{3}\,e\,x^2 + \frac{11}{36}\sqrt{2e^3}\,x^3 - \frac{43}{135}\,e^2\,x^4 \\ + \frac{769}{4320}\sqrt{2e^5}\,x^5 - \frac{1768}{8505}\,e^3\,x^6 + \cdots . \tag{2.9}$$

A series expansion closely related to (2.8) is the power series expansion of $-w_{-1}(-\exp(-1 + z^2/2))$ in [4, Eq. (48)]; other related series can be found in [4, Section 3.2]. Both expansions (2.8) and (2.9) are appropriate for, say, $x^2 \leq .5 \times 10^{-4}$, while Newton's method is adequate in all other cases. In symbolic routines, using variable-precision artithmetic, only Newton's method needs to be used, together with appropriate precautions near the branch point $(w, y) = (-1, -1/e)$ (cf. Section 2.3).

2.3 Matlab implementation

The procedures described in Sections 2.1 and 2.2 are implemented in the Matlab routines `wofy.m`, `wofy0.m`, `wofy1.m`. The respective symbolic analogues are `swofy.m`, `swofy0.m`, `swofy1.m`. These use only Newton's method; to counteract the loss of accuracy in `swofy0.m` and `swofy1.m` when y is near and above $-1/e$, the working precision in these two routines must be selected sufficiently larger than the target precision. For `wofy0.m` one needs 4, 6, 10 more digits than in the target precision when the distance of y from $-1/e$ is respectively 10^{-5}, 10^{-10}, and 10^{-20}; for `swofy1.m` the numbers are 4, 7, and 12 digits, respectively.

3 The integrals $I_{0,[1,\infty]}(\alpha, \beta)$ and $I_{0,[0,1]}(\alpha, \beta)$

For the evaluation of $I_{0,[1,\infty]}(\alpha, \beta)$, $\alpha \neq \beta$, we use the integral representation (1.8). In view of the logarithmic/algebraic singularity at the lower limit of the integral in (1.8), and its regularity at the upper limit, we decompose the integral into two parts: the first extended from 0 to $1/e$, the second from $1/e$ to 1. The former is written as

$$I_{[0,1/e]} = \int_0^{1/e} (1 - x)^{-(\alpha-\beta+1)} \cdot x^{\alpha-1}[\ln(1/x)]^{\alpha-\beta+1}\mathrm{d}x, \tag{3.1}$$

the second factor being treated as a weight function,

$$v(x; \alpha, \beta) = x^{\alpha-1}[\ln(1/x)]^{\alpha-\beta+1}, \quad 0 < x \leq 1/e, \tag{3.2}$$

with the intention of applying Gauss quadrature relative to this weight function. The second part,

$$I_{[1/e,1]} = \int_{1/e}^{1} u^{\alpha-1} \left(\frac{\ln(1/u)}{1-u} \right)^{\alpha-\beta+1} \mathrm{d}u,$$

after the change of variable $u = 1 - x(1 - 1/e)$, becomes

$$I_{[1/e,1]} = (1-1/e) \int_0^1 [1-(1-1/e)x]^{\alpha-1} \left(\frac{-\ln(1-(1-1/e)x)}{(1-1/e)x} \right)^{\alpha-\beta+1} \mathrm{d}x \tag{3.3}$$

and is amenable to Gauss–Legendre quadrature on [0, 1].

With regard to Gauss quadrature for the (nonstandard) weight function (3.2), we generate the relevant orthogonal polynomials (see [6] for details) by the variable-precision Chebyshev algorithm from the moments

$$\begin{aligned}\mu_k(v; \alpha, \beta) &= \int_0^{1/e} x^{k+\alpha-1} [\ln(1/x)]^{\alpha-\beta+1} \mathrm{d}x = \int_1^{\infty} t^{\alpha-\beta+1} e^{-(k+\alpha)t} \mathrm{d}t \\ &= \frac{1}{(k+\alpha)^{\alpha-\beta+2}} \Gamma(\alpha-\beta+2, k+\alpha), \quad k = 0, 1, 2, \ldots .\end{aligned}$$

These are generated (in variable-precision arithmetic) by the Matlab routine `smomvab.m`[2]. The Matlab routine `sr_vab.m` then generates the first N recurrence coefficients of the required orthogonal polynomials and stores them in the $N \times 2$ array `abv`. These in turn allow us to generate the desired N-point Gaussian quadrature rule using the `SOPQ` routine `sgauss.m`. The evaluation of $I_{0,[1,\infty]}(\alpha, \beta)$ from (1.8), using (3.1) and (3.3), is implemented in the Matlab routine `sI01infab.m`.

We ran this procedure in 32-digit arithmetic for $\alpha = 2, 1, \frac{1}{2}$ ($\alpha = 0$ is trivial), and for each of these values for $\beta = 2, 1, \frac{1}{2}, 0, -\frac{1}{2}, -1, -2$ ($\alpha = 1$, $\beta = 2$ being trivial). The most time-consuming part of these calculations is the generation of the N recurrence coefficients by the routine `sr_vab`, which, when $N = 50$, took about 12 minutes or less on the Sun Ultra 5 workstation, assuming a good estimate of `dig0`—the initial number of digits used in the routine `sr_vab`. Tables of these coefficients can be found on the web site http://www.cs.purdue.edu/archives/2001/wxg/tables in files whose names start with "`abv`".

A sample of results is shown in Table 1, where n denotes the number of quadrature points needed for 32-digit accuracy and `dig` the number of digits required in the routine `sr_vab` to obtain the recurrence coefficients accurate to 32 digits. The number n is seen to be less than 30, which is remarkably small for this type of accuracy. When $\alpha = \beta$, there is agreement with the 32-digit results obtained by the routine `sI01infaa.m` except for an endfigure discrepancy of one unit in the case $\alpha = \beta = 2$.

The integral $I_{0,[0,1]}(\alpha, \beta)$, by (1.2), is equal to $1/(\alpha - \beta + 1)$ if $\alpha - \beta + 1 > 0$.

Table 1 The integral $I_{0,[1,\infty]}(\alpha, \beta)$ for selected values of α, β

α	β	n	`dig`	$I_{0,[1,\infty]}(\alpha, \beta)$
2	2	27	115	0.28986813369645287294483033329204
	0	26	115	0.55242099404393096881202067693593
	−2	25	110	1.5857382583390739261863136404274
1	1	29	115	0.64493406684822643647241516664603
	0	26	115	1.1449340668482264364724151666646
	−1	28	110	2.5136576366744873885388199948242
$\frac{1}{2}$	2	27	110	0.56150165251555182684424279164016
	0	28	110	2.3338359155089014467610187032541
	−2	27	110	38.727633569979724008308403165634

4 The integrals $I_{1,[0,1]}(\alpha, \beta)$ and $I_{1,[1,\infty]}(\alpha, \beta)$

In order to evaluate $I_{1,[0,1]}(\alpha, \beta)$ from the integral representation (1.9), we first make the change of variable $u = 1/x$ in the integral of (1.9) to write it as

$$\int_0^1 x^{-\beta} \left(\frac{\ln(1/x)}{1-x} \right)^{\alpha-\beta+1} \mathrm{d}x.$$

This has the same form as the integral in (1.8). Hence, we deal with it in the same way as was done in Section 3, i.e., decompose it into two integrals analogous to (3.1) and (3.3). The first is calculated by Gauss quadrature with respect to the weight function

$$u(x; \alpha, \beta) = x^{-\beta}[\ln(1/x)]^{\alpha-\beta+1}, \quad 0 < x \le 1/e, \tag{4.1}$$

the second by Gauss–Legendre quadrature of

$$(1-1/e) \int_0^1 [1-(1-1/e)x]^{-\beta} \left(\frac{-\ln(1-(1-1/e)x)}{(1-1/e)x} \right)^{\alpha-\beta+1} \mathrm{d}x.$$

The moments of the weight function (4.1) are given by

$$\mu_k(u; \alpha, \beta) = \frac{1}{(k-\beta+1)^{\alpha-\beta+2}} \Gamma(\alpha-\beta+2, k-\beta+1), \quad k = 0, 1, 2, \dots,$$

and are evaluated by the routine `smomuab.m`[2]. The Matlab routine `sr_uab.m` then generates the recurrence coefficients for the required orthogonal polynomials and stores them in the array `abu`. The integral $I_{1,[0,1]}(\alpha, \beta)$ itself is computed by the routine `sI101ab.m`.

The procedure was run in 32-digit arithmetic for $\alpha = 2, 1, \frac{1}{2}, -\frac{1}{2}$ and for each of these values for $\beta = \frac{1}{2}, 0, -\frac{1}{2}, -1, -2$ (the case $\alpha = -\frac{1}{2}$, $\beta = \frac{1}{2}$ being trivial). The first 50 recurrence coefficients required in these computations, generated by the routine `sr_uab.m`, are retrievable on the web site mentioned in Section 3 from files whose names start with "`abu`".

Selected results are shown in Table 2 in the same format as used in Table 1.

Table 2 The integral $I_{1,[0,1]}(\alpha, \beta)$ for selected values of α, β

α	β	n	`dig`	$I_{1,[0,1]}(\alpha, \beta)$
2	$\frac{1}{2}$	27	110	33.343927985540712124255844272507
	0	27	110	6.0273152733489747770776399896483
	−2	25	110	0.66470507420927905363933223999206
$\frac{1}{2}$	$\frac{1}{2}$	26	110	3.467401100272339654708622749969
	0	26	115	1.4200196887885673611304689259528
	−1	25	110	0.61417487418179378390030116175849
$-\frac{1}{2}$	0	26	115	0.74291550121126488688927157124286
	−1	25	115	0.41704512121160039084979057685156
	−2	24	115	0.29216513963802369184942362241138

When $\alpha = \beta$, the results are in complete agreement with those furnished by the routine `sI101aa.m` with `dig` $= 32$.

The integral $I_{1,[1,\infty]}(\alpha, \beta)$, by virtue of (1.3), exists only if $\beta - \alpha - 1 > 0$, and then equals $1/(\beta - \alpha - 1)$.

Acknowledgements The computational problem in the case $\alpha = \beta$ was brought to the author's attention by Tony Tam. The author is indebted to Robert M. Corless for the References [2–4], the first of which, but not the others, having been familiar to the author at the time of writing this paper.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of mathematical functions. National Bureau of Standards, Applied Mathematics Series 55, U.S. Government Printing Office, Washington, DC (1964)
2. Barry, D.A., Li, L., Jeng, D.-S.: Comments on numerical evaluation of the Lambert W-functions and application to generation of generalized Gaussian noise with exponent 1/2. IEEE Trans. Signal Process. **52**, 1456–1458 (2004)
3. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W-functions. Adv. Comput. Math. **5**, 329–359 (1996)
4. Corless, R.M., Jeffrey, D.J., Knuth, D.E.: A sequence of series for the Lambert W-functions. In: Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI), pp. 197–204. ACM, New York (1997, electronic)
5. Gautschi, W.: Numerical Analysis: An Introduction. Birkhäuser, Boston (1997)
6. Gautschi, W.: Variable-precision recurrence coefficients for non-stand-ard orthogonal polynomials. Numer. Algorithms **52**, 409–418 (2009)
7. Gradshteyn, I.S., Ryzhik, I.M.: Table of integrals, series, and products, 7th edn. Elsevier/Academic Press, Amsterdam (2007)
8. Yu, Y.: Personal communication, October (2009)