Meetings with Lambert W and Other Special Functions in Optimization and Analysis

Jonathan M. Borwein
CARMA
University of Newcastle

Scott B. Lindstrom
CARMA
University of Newcastle

June 13, 2016

Abstract

We remedy the under-appreciated role of the Lambert $\mathcal{W}$ function in convex analysis and optimization. We first discuss the role of little-known special functions in optimization and then illustrate the relevance of $\mathcal{W}$ in a series problem posed by Donald Knuth. We then provide a basic overview of the convex analysis of $\mathcal{W}$ and go on to explore its role in duality theory where it appears quite naturally in the closed forms of the convex conjugates for certain functions. We go on to discover a useful class of functions for which this is the case and investigate their use in optimization, providing some code and examples.

1 Introduction

This paper and its accompanying lecture could be entitled "Meetings with $\mathcal{W}$ and other *too little known* functions." It is provided in somewhat of a tutorial form so as to allow us to meet a wider audience.

Throughout our discussion, we explain the role computer assistance played in our discoveries, with particular attention to our *Maple* package *Symbolic Convex Analysis Tools* and its numerical partner *CCAT*. We also note that, superficially, $\mathcal{W}$ provides an excellent counterexample to Stigler's law of eponomy [25] which states that a scientific discovery is named after the last person to discover it.

For us, the Lambert $\mathcal{W}$ function is the real analytic inverse of $x \mapsto x\exp(x)$. The real inverse is two-valued, as shown in Figure 1, while the complex inverse has countably many branches. We are interested in the principal branch. This is the analytic branch of $\mathcal{W}$ that has the following Taylor series

$$\mathcal{W}(x) = \sum_{k=1}^{\infty} \frac{(-k)^{k-1}}{k!} x^k, \tag{1}$$

with radius of convergence $1/e$. This is the solution that reversion of the series (otherwise called the Lagrange inversion theorem) produces from $x \cdot e^x$. Implicit differentiation leads to

$$\mathcal{W}'(x) = \frac{\mathcal{W}(x)}{x\,(1 + \mathcal{W}(x))}. \tag{2}$$

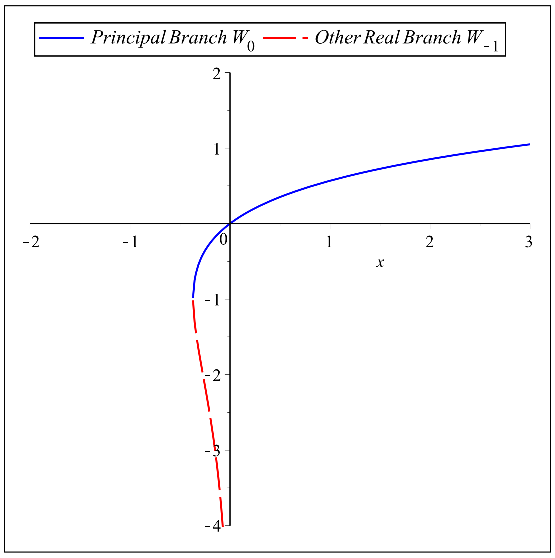


Figure 1: The real branches of the Lambert $\mathcal{W}$ function.

1.1 The power of naming

In the current mathematical world, it matters less what you know about a given function than whether your computer package of choice (say *Maple, Mathematica* or *SAGE*) or online source, say Wikipedia [26] does. We illustrate this first with the *Meijer G function* (see, for example, [27]) before focusing in more detail on $\mathcal{W}$. Our intent in so doing is *not* to wander far afield of our principal focus on $\mathcal{W}$ but *rather* to illuminate the role of computer assisted discovery in the modern research climate by highlighting its important role in solving optimization problems with special functions in general. With this motivation of emphasizing the computational role, we can then begin our serious attack on $\mathcal{W}$.

1.2 Meeting Meijer-G

The Meijer-G function is very useful, if a bit difficult for a human to remember the exact definition for. Often one's computer can help. In 2002, Nick Trefethen, as described in [3], published a list of ten numerical challenge problems in *SIAM Review.* The ninth problem is in optimization.

Example 1 (Trefethen's ninth problem [3])**.** The problem is posed as follows.

> The integral
>
> $$I(\alpha) = \int_0^2 [2 + \sin(10\alpha)]x^\alpha \sin\left(\frac{\alpha}{2-x}\right) \mathrm{d}x$$
>
> depends on the parameter α. What is the value $\alpha \in [0, 5]$ at which $I(\alpha)$ achieves its maximum?

Answer. $I(\alpha)$ is expressible in terms of a *Meijer-G* function: a special function with a solid history that lets us solve the problem. While researchers who have prior experience with these special functions may come to the same conclusions by hand, *Mathematica* and *Maple* will figure it out as well. As in Figure 2, help files, a web search, or *Maple*'s *Function Advisor* then inform the scientist. This is another measure of the changing environment: naiveté need no longer impair to the same extent when the computer may aid in the discovery. It is usually a good idea—and not at all immoral—to data mine. The exact form of $I(\alpha)$ as given by *Maple* is in Equation 3.

$$I(\alpha) = 4\sqrt{\pi}\,\Gamma(\alpha)\,G^{3,0}_{2,4}\left(\frac{\alpha^2}{16}\,\middle|\,\begin{matrix}\frac{\alpha+2}{2},\frac{\alpha+3}{2}\\ \frac{1}{2},\frac{1}{2},1,0\end{matrix}\right)[\sin(10\alpha)+2]\,. \tag{3}$$

If the Meijer-G function is well implemented, one can use any good numerical optimiser. The authors of [19] have written about the use of special functions for integration and credit Michael Monagan and Greg Fee with the original implementation in *Maple.* ◇

Example 2 (Moments of random walks [11])**.** The *moment function* of an $n-$step random walk in the plane is:

$$\mathcal{M}_n(s) \quad = \quad \int_{[0,1]^n}\left|\sum_{k=1}^{n} e^{2\pi x_k i}\right|^s \mathrm{d}(x_1,\ldots,x_{n-1},x_n).$$

The first breakthrough in our study of short random walks [11] is given in Theorem 1.

Theorem 1 (Meijer-G form for $\mathcal{M}_3$)**.** *For s not an odd integer,*

$$\mathcal{M}_3(s) = \frac{\Gamma(1+\frac{s}{2})}{\sqrt{\pi}\;\Gamma(-\frac{s}{2})}\,G^{21}_{33}\left(\begin{matrix}1,1,1\\ \frac{1}{2},-\frac{s}{2},-\frac{s}{2}\end{matrix}\,\middle|\,\frac{1}{4}\right). \tag{4}$$

Equation (4) was first found by Crandall via CAS and proven using residue calculus methods.

Indeed, $\mathcal{M}_3(s)$ is among the first non-trivial higher order Meijer-G functions to be placed in closed form. We were then led to the results of Theorem 2.

Theorem 2 (Meijer form for $\mathcal{M}_4$)**.** *For $\Re s > -2$ and s not an odd integer*

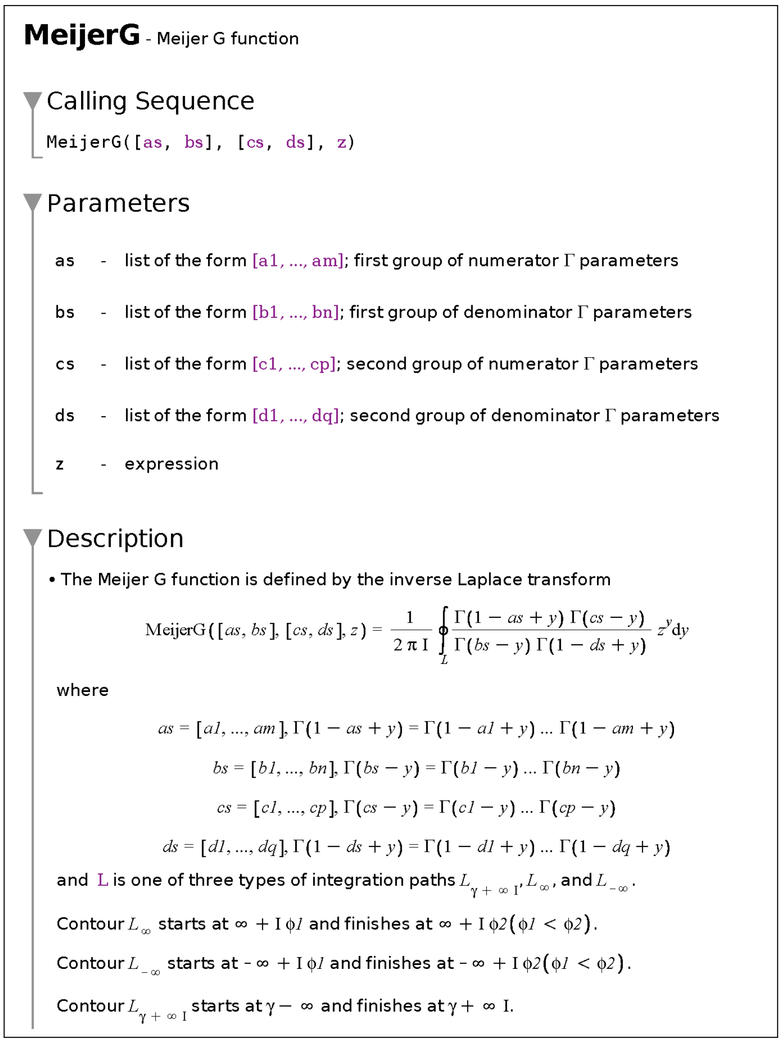
$$\mathcal{M}_4(s) = \frac{2^s}{\pi}\frac{\Gamma(1+\frac{s}{2})}{\Gamma(-\frac{s}{2})}\,G^{22}_{44}\left(\begin{matrix}1,\frac{1-s}{2},1,1\\ \frac{1}{2}-\frac{s}{2},-\frac{s}{2},-\frac{s}{2},-\frac{s}{2}\end{matrix}\,\middle|\,1\right). \tag{5}$$

Armed with these two results we were ready to mount our serious attack on the moment functions and corresponding densities. This led to such useful results as a closed hypergeometric form for the radial density of a three-step walk:

$$p_3(\alpha) = \frac{2\sqrt{3}\alpha}{\pi\;(3+\alpha^2)}\,{}_2F_1\left(\begin{matrix}\frac{1}{3},\frac{2}{3}\\ 1\end{matrix}\,\middle|\,\frac{\alpha^2\left(9-\alpha^2\right)^2}{\left(3+\alpha^2\right)^3}\right).$$

The Meijer-G function has been instrumental in producing a new result on a hundred-year-old topic. ◇

Of course, many special functions are unknown to most of us. We list, for example, the Painlevé transcendents [18], the Lerch transcendent, and the Heun equations. In each of these cases the CAS can enlighten us.


MeijerG - Meijer G function

Calling Sequence

```
MeijerG([as, bs], [cs, ds], z)
```

Parameters

as	-	list of the form [a1, ..., am]; first group of numerator Γ parameters
bs	-	list of the form [b1, ..., bn]; first group of denominator Γ parameters
cs	-	list of the form [c1, ..., cp]; second group of numerator Γ parameters
ds	-	list of the form [d1, ..., dq]; second group of denominator Γ parameters
z	-	expression

Description

• The Meijer G function is defined by the inverse Laplace transform

$$\mathrm{MeijerG}([as, bs], [cs, ds], z) = \frac{1}{2\pi I}\oint_L \frac{\Gamma(1-as+y)\,\Gamma(cs-y)}{\Gamma(bs-y)\,\Gamma(1-ds+y)} z^y \mathrm{d}y$$

where

$$as = [a1, ..., am], \Gamma(1-as+y) = \Gamma(1-a1+y) \ldots \Gamma(1-am+y)$$

$$bs = [b1, ..., bn], \Gamma(bs-y) = \Gamma(b1-y) \ldots \Gamma(bn-y)$$

$$cs = [c1, ..., cp], \Gamma(cs-y) = \Gamma(c1-y) \ldots \Gamma(cp-y)$$

$$ds = [d1, ..., dq], \Gamma(1-ds+y) = \Gamma(1-d1+y) \ldots \Gamma(1-dq+y)$$

and L is one of three types of integration paths $L_{\gamma+\infty I}$, L_{∞}, and $L_{-\infty}$.

Contour L_{∞} starts at $\infty + I\,\phi1$ and finishes at $\infty + I\,\phi2 (\phi1 < \phi2)$.

Contour $L_{-\infty}$ starts at $-\infty + I\,\phi1$ and finishes at $-\infty + I\,\phi2 (\phi1 < \phi2)$.

Contour $L_{\gamma+\infty I}$ starts at $\gamma - \infty$ and finishes at $\gamma + \infty$ I.

Figure 2: What *Maple* knows about Meijer-G.

2 Knuth's Series Problem: Experimental mathematics and $\mathcal{W}$

We continue with an account of the solution in [5], to a problem posed by Donald E. Knuth of Stanford in the November 2000 issues of the *American Mathematical Monthly*. See [21] for the published solution. We initially follow the discussion in [5] quite closely.

Problem 10832 Evaluate

$$S = \sum_{k=1}^{\infty} \left(\frac{k^k}{k!e^k} - \frac{1}{\sqrt{2\pi k}} \right).$$

Solution: We first attempted to obtain a numerical value for S. Using *Maple*,

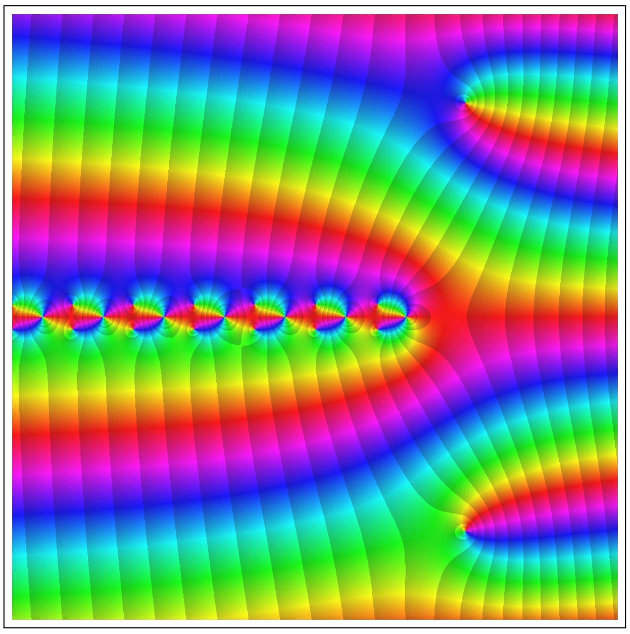


Figure 3: The complex moment function $\mathcal{M}_4$ as drawn from (5) in the Calendar *Complex Beauties 2016*.

we produced the approximation

$$S \quad \approx \quad -0.084069508727655996 46.$$

Based on this numerical value, the Inverse Symbolic Calculator, available at the URL http://isc.carma.newcastle.edu.au/, with the "Smart Lookup" feature, yielded the result

$$S \quad \approx \quad -\frac{2}{3} - \frac{1}{\sqrt{2\pi}}\zeta\left(\frac{1}{2}\right).$$

Calculations to even higher precision (50 decimal digits) confirmed this approximation. Thus within a few minutes we "knew" the answer.

Why should such an identity hold? One clue was provided by the surprising speed with which *Maple* was able to calculate a high-precision value of this slowly convergent infinite sum. Evidently, the *Maple* software knew something that we did not. Peering under the hood, we found that *Maple* was using the Lambert W function, which, as we know, is the functional inverse of $z \to ze^z$.

Another clue was the appearance of $\zeta(1/2)$ in the above experimental identity, together with an obvious allusion to Stirling's formula in the original problem. This led us to conjecture the identity

$$\sum_{k=1}^{\infty}\left(\frac{1}{\sqrt{2\pi k}} - \frac{(1/2)^{\overline{k-1}}}{(k-1)!\sqrt{2}}\right) \quad = \quad \frac{1}{\sqrt{2\pi}}\zeta\left(\frac{1}{2}\right),$$

where $(x)^{\overline{n}}$ denotes $x(x+1)\cdots(x+n-1)$, we say "x to the n rising," [20] and where the binomial coefficients in the LHS of (6) are the same as those of the function $1/\sqrt{2-2x}$. Moreover, *Maple* successfully evaluated this summation, as shown on the RHS and as is further discussed in Remark 1. We now needed to establish that

$$\sum_{k=1}^{\infty}\left(\frac{k^k}{k!e^k}-\frac{(1/2)^{\overline{k-1}}}{(k-1)!\sqrt{2}}\right) \quad = \quad -\frac{2}{3}.$$

Guided by the presence of the Lambert $\mathcal{W}$ function, as in (1),

$$\mathcal{W}(z) \quad = \quad \sum_{k=1}^{\infty}\frac{(-k)^{k-1}z^k}{k!},$$

an appeal to Abel's limit theorem suggested the conjectured identity

$$\lim_{z\to 1}\left(\frac{d\mathcal{W}(-z/e)}{dz}+\frac{1}{\sqrt{2-2z}}\right) \quad = \quad \frac{2}{3}. \tag{6}$$

Here again, *Maple* was able to evaluate this limit and establish the identity (6) which relies on the following reversion [16]. Let $p=\sqrt{2(1+ez)}$ with $z=\mathcal{W}e^{\mathcal{W}}$, so that

$$\frac{p^2}{2}-1=\mathcal{W}\exp(1+\mathcal{W})=-1+\sum_{k\geq 1}\left(\frac{1}{k!}-\frac{1}{(k-1)!}\right)(1+\mathcal{W})^k$$

and revert to

$$1+\mathcal{W}=p-\frac{p^2}{3}+\frac{11}{72}p^3+\ldots$$

for $|p|<\sqrt{2}$. Now (2) lets us prove (6). □

As can be seen from this account, the above manipulation took considerable human ingenuity, in addition to computer-based symbolic manipulation. We include this example to highlight a challenge for the next generation of mathematical computing software—these tools need to more completely automate this class of operations, so that similar derivations can be accomplished by a significantly broader segment of the mathematical community.

Remark 1 ($\zeta(s)$ for $0<s<\infty, s\neq 1$)**.** More generally, for $0<\operatorname{Re} s<1$ in the complex plane, we discovered empirically that

$$\sum_{k=1}^{\infty}\left(\frac{1}{k^s}-\frac{\Gamma(k-s)}{\Gamma(k)}\right)=\zeta(s).$$

Now *Maple*'s summation tools can reduce this to

$$\sum_{k=1}^{N}\frac{1}{k^s}-\frac{\Gamma(N+1-s)}{(1-s)\Gamma(N)}\to\zeta(s).$$

For any given rational $s\in(0,\infty)$ *Maple* will evaluate the limit by the Euler-Maclaurin method. Consulting the DLMF at http://dlmf.nist.gov/25.2#E8 we discover

$$\zeta(s)=\sum_{k=1}^{N}\frac{1}{k^s}+\frac{N^{1-s}}{s-1}-s\int_N^{\infty}\frac{x-\lfloor x\rfloor}{x^{s+1}}\,\mathrm{d}x.$$

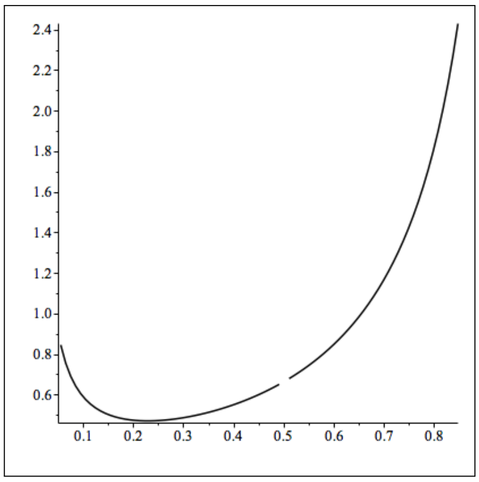


Figure 4: The function κ to the left and right of $s = 1/2$.

.

Since the integral tends to zero for $s > 0$ and

$$\lim_{N\to\infty} \frac{\Gamma(N+1-s)}{(1-s)\,\Gamma(N)} - \frac{N^{1-s}}{1-s} = 0,$$

we can also produce an explicit human proof. $\diamond$

We end this section with an open question:

Question 1. Can one find an extension of (6) for general $s \neq 1/2$ in $(0,1)$? Based on (7) and the Stirling approximation for $\Gamma(k+s) \approx \sqrt{2\pi}\, e^{-k} k^{k+s-1/2}$ we obtain

$$\sum_{k=1}^{\infty} \left(\frac{1}{\sqrt{2\pi} k^s} - \frac{k^{k+1/2-s}}{k!\, e^k} \right) - \frac{\zeta(s)}{\sqrt{2\pi}} = \kappa(s)\,.$$

We have that $\kappa(1/2) = 2/3$, but it remains to evaluate $\kappa(s) \in \mathbb{R}$ more generally, as drawn in Figure 4.

For $s \neq 1/2$ we have not found an analogue to (6), and there is no reason to be sure such an analog exists. Numerically to 25 places we record:

$$\begin{aligned}
\kappa(1/3) &= 0.5051265122136281644488407\\
\kappa(2/3) &= 1.044357456635617976159955\\
\kappa(1/4) &= 0.4742404657846664773555294\\
\kappa(3/4) &= 1.435469800298317747887340\\
\kappa(1/6) &= 0.4899050094518209209997454\\
\kappa(5/6) &= 2.226651254233652670106746.
\end{aligned}$$

Thus our question is closely allied to that of asking whether

$$\mathcal{W}(x;s) = \sum_{k=1}^{\infty} \frac{k^{k+1/2-s}}{k!} x^k$$

for $s \neq 1/2$, can be analysed in terms of $\mathcal{W}$.

3 The convex analysis of $\mathcal{W}$ (as a real function)

The *Maple* package *Symbolic Convex Analysis Tools*, or *SCAT*, which performs convex analysis symbolically, and its partner CCAT (which performs convex analysis numerically) are described in [7] and are available together at `http://carma.newcastle.edu.au/ConvexFunctions/SCAT.ZIP`. We refer to [8, 12, 14] for any background convex analysis not discussed herein.

3.1 Basic properties

1. $\mathcal{W}$ is concave on $(-1/e, \infty)$ and positive on $(0, \infty)$.
2. $(\log \circ \mathcal{W})(z) = \log(z) - \mathcal{W}(z)$ is concave, as $\mathcal{W}$ is log concave on $(0, \infty)$.
3. $(\exp \circ \mathcal{W})(z) = z/\mathcal{W}(z)$ is concave.

In order to prove these, we will make use of the following lemma, which is quite convenient.

Lemma 3. *An invertible real valued function f with domain $X \subset \mathbb{R}$ is concave if its inverse function f^{-1} is convex and monotone increasing on its domain $f^{-1}(X)$.*

Proof. Other variations of this result are readily available (see, for example, [24]), and it is often left as an exercise in texts (see, for example, [15, Exercise 3.3]). Let $x, y \in X$. By the bijectivity of the function f, there exist u, v such that $f^{-1}(u) = x$ and $f^{-1}(v) = y$. Thus we have that

$$f(\lambda x + (1-\lambda)y) = f(\lambda f^{-1}(u) + (1-\lambda)f^{-1}(v)).$$

By the convexity of f^{-1}, we have that

$$\lambda f^{-1}(u) + (1-\lambda)f^{-1}(v) \leq f^{-1}(\lambda u + +(1-\lambda)v) \tag{7}$$

Since f^{-1} is monotone increasing, f is monotone increasing. Using this fact together with Equation 7 shows that

$$f(\lambda f^{-1}(u) + (1-\lambda)f^{-1}(v)) \leq f(f^{-1}(\lambda u + +(1-\lambda)v)) = \lambda u + (1-\lambda)v. \tag{8}$$

Finally, we have

$$\lambda u + (1-\lambda)v = \lambda f(x) + (1-\lambda)f(y).$$

This result greatly simplifies the following propositions. □

Corollary 4. *An invertible real valued function f with domain $X \subset \mathbb{R}$ is concave if its inverse function f^{-1} is convex and monotone decreasing on its domain $f^{-1}(X)$.*

Proof. The proof is the same as that of Lemma 3 with the exception that the direction of the inequality is reversed in Equation (8) beause the inverse is now decreasing instead of increasing. □

Proposition 5. *$\mathcal{W}$ is concave on $(-1/e, \infty)$.*

Proof. Notice that $\mathcal{W}^{-1}((\frac{1}{e},\infty)) = (-1,\infty)$. By Lemma 3, it suffices to show that the inverse of $\mathcal{W}$ is convex and monotone increasing on $(-1,\infty)$. Since the inverse of $\mathcal{W}$ is xe^x, we differentiate; the convexity and monotonicity are clear from the fact that the first two derivatives are both positive on the entire domain. $\square$

Definition 1. A function f is called logarithmically concave if it is strictly greater than zero on its domain and $\log \circ f$ is concave. Similarly f is called logarithmically convex if it is strictly greater than zero on its domain and $\log \circ f$ is convex.[15]

Remark 2. The function $(\log \circ \mathcal{W})(z) = \log(z) - \mathcal{W}(z)$ is concave. This is true since $\mathcal{W}$ is concave and its restriction to $(0,\infty) = \mathbf{dom}(\log \circ \mathcal{W})$ is strictly greater than zero everywhere.

Proposition 6. *The function given by* $(\exp \circ \mathcal{W})(x) = x/\mathcal{W}(x)$ *is concave.*

Proof. Consider the inverse of this function

$$\log(x)e^{\log(x)} = x\log(x).$$

Again, by Lemma 3, it suffices to show that this function is convex and monotone increasing on its domain $(\frac{1}{e},\infty)$. Differentiating, we again find that both the first and second derivatives are strictly greater than zero, showing the result. $\square$

Proposition 7. $(\exp \circ (-\mathcal{W}))(x) = \mathcal{W}(x)/x$ *is convex.*

Proof. By Corollary 4, it suffices to show that the inverse function

$$-\log(x)e^{-\log(x)} = -\log(x)/x$$

is convex and decreasing on its domain $(0,e)$. Indeed, on this interval the first derivative is always negative and the second always positive, showing both traits. $\square$

3.2 The convex conjugate

The convex conjugate — or *Fenchel–Moreau–Rockafellar conjugate* or *dual function* — plays much the role in convex analysis and optimization that the Fourier transform plays in harmonic analysis. For a function $f\colon X \to [-\infty,\infty]$ we define

$$\begin{aligned} &f^* : X^* \to [-\infty,\infty] \;\text{ by} \\ &f^*(y) = \sup_{x\in X}\{\langle y,x\rangle - f(x)\}. \end{aligned} \tag{9}$$

The function f^* is always convex (if possibly always infinite), and if f is lower semicontinuous, convex and proper then $f^{**} = f$. In particular if we show a function $g = f^*$ then g is necessarily convex.

Directly from (9) we have the *Fenchel-Young* inequality, for all y, x

$$f^*(y) + f(x) \geq \langle y,x\rangle.$$

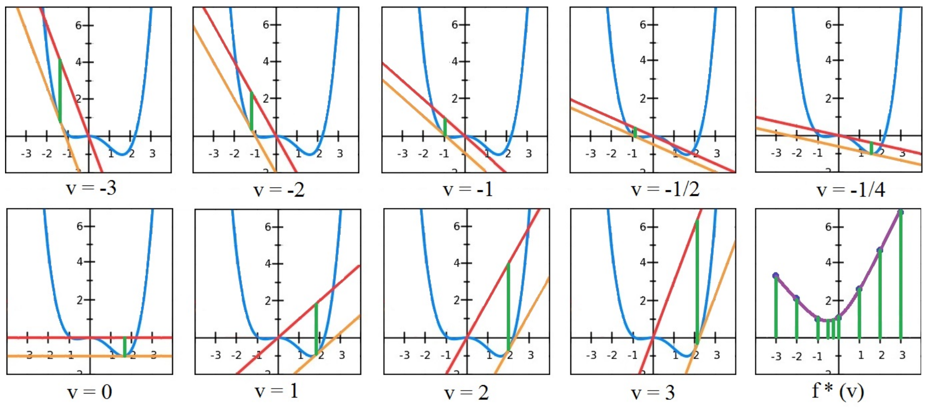


Figure 5: Here the construction of f^* is shown for the function $f(x) = (1/4)x^4 - (1/3)x^3 - (1/2)x^2$. The real-valued inputs of f^* may be thought of as the slopes of the lines through the origin in each image. For each input, we can imagine obtaining the corresponding output by taking a line parallel to the one through the origin and sliding it down as far away from the original line as as it can go while still touching the curve of the function f. The output is represented by the vertical distance between the two lines. This and several other graphical examples may be found in [22].

Thus, for $1/p + 1/q = 1$ with $p, q > 0$, we have that the dual to the function $f(x) = \frac{|x|^p}{p}$ is $f^*(y) = \frac{|y|^q}{q}$. We also have that the *energy* function defined by

$$f(x) = \frac{|x|^2}{2} \tag{10}$$

is the only self-conjugate function.

We also have another dual pair which we will make use of several times. For the function $f(x) = \exp(x)$, we have $f^*(y) = y\log(y) - y$. We call $y\log(y) - y$ the *(negative) Boltzmann-Shannon entropy*. Conversely, since exp is a convex function, we also have that exp is the conjugate of the Boltzmann-Shannon entropy. We may, for the sake of simplicity, express convex conjugacy in the following way:

$$\begin{aligned}(\exp)^*(y) &= y\log(y) - y \\ (x\log(x) - x)^*(y) &= \exp(y).\end{aligned} \tag{11}$$

Here the star indicates that the function under consideration is the convex conjugate of that enclosed in the brackets. Another important example is the *log barrier* $f(x) = -\log x$ for $x > 0$ with conjugate $f^*(y) = -1 - \log y$ for $x < 0$. Notice that this is essentially self dual on changing x to $-x$. These conjugates can be computed directly from the definition, as illustrated below in Example 3, or in the *SCAT* software package.

The convex conjugate produces a duality between compactness of lower level sets and continuity, between strict convexity and differentiability, and also much else [12, Ch. 4]. We illustrate this for the energy (10), negative entropy (11), and log barrier in Figure 6.

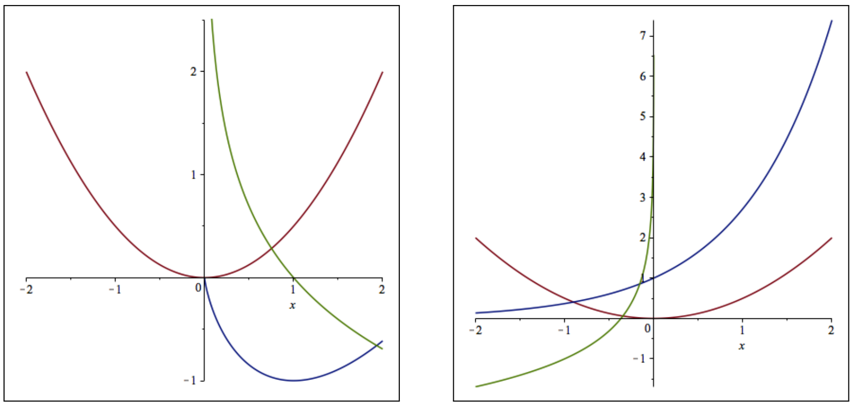


Figure 6: The energy, log barrier and negative entropy (L) and duals (R).

.

Expressed in the same form as specified in Equation (11), an apparently new dual pair is

$$(-\mathcal{W})^*(x) = \begin{cases} \mathcal{W}\left(\frac{1-\mathcal{W}(-e/x)}{x\mathcal{W}(-e/x)}\right) + \frac{1}{\mathcal{W}(-e/x)} - 1 & \text{if } x < 0 \\ \infty & \text{otherwise} \end{cases}.$$

Once a computationally effective closed form is available, all of classical convex duality theory is accessible. This is illustrated for $\mathcal{W}$ in Section 4 and Section 5.

The convex conjugate also exchanges addition of functions with their *infimal convolution*

$$(f \,\square\, g)(y) = \inf_{x \in X}\{f(y-x) + g(x)\}.$$

Indeed $(f \,\square\, g)^* = f^* + g^*$ always holds, and, under mild hypotheses,

$$(f+g)^* = f^* \,\square\, g^*.$$

See [8, 12, 14] for details.

3.3 The convex conjugate for log convex functions

We next consider the conjugate exponents of *log convex* functions. These functions appear frequently in statistical settings. Recalling Definition 1, we may think of a log convex function g as being equal to $\exp \circ f$ for some convex function f. Explicitly,

$$g(x) = e^{f(x)}$$

The Gamma function is a nice example of such a function. Recalling Equation (9), the convex conjugate g^* is just

$$g^*(y) = \sup_{x \in X}\{yx - e^{f(x)}\}. \tag{12}$$

By differentiating the inner term $yx - e^{f(x)}$ and setting equal to zero to find the point at which the function is maximized, we obtain

$$y = f'(x)e^{f(x)}. \tag{13}$$

If we can solve this equation for $x = s(y)$, we will be able to write the conjugate function more clearly as

$$g^*(y) = y \cdot s(y) - g(s(y))$$

How nice the answer is depends on how well this expression simplifies.

Example 3. A first and lovely example is:

$$(\exp \circ \exp)^* (y) = \begin{cases} y \left(\log y - \mathcal{W}(y) - 1/\mathcal{W}(y)\right) & y > 0 \\ -1 & y = 0 \\ \infty & y < 0 \end{cases}$$

We will deduce this in the way described above. ◇

Starting from the definition,

$$g^*(y) = \sup_{x \in \mathbb{R}} \{yx - e^{e^x}\},$$

we take the derivative of the inner term on the right and set equal to zero to obtain

$$y = e^x \cdot e^{e^x}.$$

Notice that what we have here is just $e^x = \mathcal{W}(y)$ and so we may actually solve for x explicitly as follows:

$$x = (\log \circ \mathcal{W})(y).$$

Thus we can substitute back into the original equation and have our closed form solution

$$g^*(y) = y \cdot (\log \circ \mathcal{W})(y) - e^{\mathcal{W}(y)} = y(\log(y) - \mathcal{W}(y)) - y/W(y). \tag{14}$$

Example 4. A second example is related to the normal distribution. We take:

$$g(x) = e^{\frac{x^2}{2}} \text{ for all } x,$$

and we have that

$$g^*(y) = |y| \left(\sqrt{\mathcal{W}(y^2)} - \frac{1}{\sqrt{\mathcal{W}(y^2)}} \right) \text{ for all } y.$$

We derive this below. ◇

We can compute this result in the same way. Starting with the definition

$$g^*(y) = \sup_{x \in \mathbb{R}} \{yx - e^{\left(\frac{x^2}{2}\right)}\}. \tag{15}$$

Differentiating and setting equal to zero, we obtain

$$y = x \cdot e^{\left(\frac{x^2}{2}\right)} \tag{16}$$

Squaring both sides, we have $y^2 = x^2 \cdot e^{x^2}$. We can now see a way to use the Lambert $\mathcal{W}$ function: $x^2 = \mathcal{W}(y^2)$. Thus we arrive at an expression for x:

$$x = \operatorname{sign}(y) \cdot \sqrt{\mathcal{W}(y^2)}.$$

This we can work with. If we instead asked *Maple* to solve Equation (16) for x, it gives us the solution

$$x = y \cdot e^{-\frac{1}{2}W(y^2)}.$$

We can easily check to see this is an equivalent expression. Recall, as first noted in Proposition 6, that $(\exp \circ \mathcal{W})(x) = x/\mathcal{W}(x)$. Thus we may write

$$e^{-\frac{1}{2}W(y^2)} = \left(e^{\mathcal{W}(y^2)}\right)^{-\frac{1}{2}} = \left(\frac{y^2}{\mathcal{W}(y^2)}\right)^{-\frac{1}{2}} = \sqrt{\frac{\mathcal{W}(y^2)}{y^2}} = \frac{\sqrt{\mathcal{W}(y^2)}}{|y|}.$$

Thus we have that

$$y \cdot e^{-\frac{1}{2}W(y^2)} = y \cdot \frac{\sqrt{\mathcal{W}(y^2)}}{|y|} = \operatorname{sign}(y)\sqrt{\mathcal{W}(y^2)}.$$

Since we have an expression for x in terms of y, we can substitute back into the original formulation from Equation (15) to obtain our closed-form expression

$$g^*(y) = |y|\sqrt{\mathcal{W}(y^2)} - \exp\left(\frac{|W(y^2)|}{2}\right). \tag{17}$$

We can simplify this even further. $\mathcal{W}(y^2)$ is always positive since y^2 is always positive, so we can lose the absolute value signs in the right most term and further simplify it as follows:

$$\exp\left(\frac{|W(y^2)|}{2}\right) = \sqrt{e^{\mathcal{W}(y^2)}} = \sqrt{\frac{y^2}{\mathcal{W}(y^2)}} = |y|\frac{1}{\sqrt{\mathcal{W}(y^2)}}$$

Thus Equation (17) simplifies to

$$g^*(y) = |y|\left(\sqrt{\mathcal{W}(y^2)} - \frac{1}{\sqrt{\mathcal{W}(y^2)}}\right). \tag{18}$$

We can check this answer using SCAT. We ask *Maple* to compute

```
n1:=convert(exp((x^2)/2),PWF);Conj(n1,y);
```

which yields the answer

$$\begin{cases} |y|\dfrac{W(y^2)-1}{\sqrt{W(y^2)}} & all(y). \end{cases}$$

This matches our solution for $g^*(y)$ from Equation (18).

3.4 Conjugate of $\exp \circ f$, II

Let us now unpack Examples 3 and 4 in more generality. Suppose we desire to find the convex conjugate of a function of the form $g(x) = (\exp \circ f)(x)$ where f is either invertible (such as in Example 3) or locally invertible (as in Example 4). Then, if we can first solve the equation

$$f'(x)^{\alpha+1} = \gamma f(x) \tag{19}$$

for any α and nonzero γ, we will be able to express g^* in closed form by using the $\mathcal{W}$ function. To see why this is so, recall from Equation (13) that, for a function of the form $g(x) = (\exp \circ f)(x)$, we can obtain a closed-form of the convex conjugate if we can solve for x in the equation $y = f'(x)e^{f(x)}$. Now suppose that we can solve Equation (19). Then we can raise both sides of Equation (13) to the power $\alpha + 1$ to obtain

$$y^{\alpha+1} = f'(x)^{\alpha+1}e^{(\alpha+1)f(x)} = \gamma f(x)e^{(\alpha+1)f(x)}.$$

We can then multiply both sides by $\frac{\alpha+1}{\gamma}$ to obtain

$$(\alpha+1)\frac{y^{\alpha+1}}{\gamma} = (\alpha+1)f(x)e^{(\alpha+1)f(x)}.$$

Now we can see how to use the $\mathcal{W}$ function:

$$(\alpha+1)f(x) = \mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right).$$

We thus arrive at the solution

$$f(x) = \frac{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}{\alpha+1}$$
$$x = b\left(\frac{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}{\alpha+1}, y\right)$$

where $b(z, y) = f^{-1}(z)$ in the case where f is invertible and $b(z, y)$ is the pre-image choice $f^{-1}(z)$ such that $z \cdot y$ is maximized otherwise. In the case where f is convex, there will be at most two such pre-image choices; we are excluding the case where f is a constant function because, in that case, $\gamma = 0$ (note that the convex conjugate is then trivial). We can substitute this back into Equation (12) to obtain:

$$g^*(y) = y \cdot b(d(y), y) - (\exp \circ d)(y) \text{ where} \tag{20}$$
$$d(y) = \frac{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}{\alpha+1}$$

In the case of Example 3, immediately $b(z, y) = f^{-1}(z) = \log(z)$. In the case of $f(x) = \frac{|x|^p}{p}$ (as in Example 4), we have

$$b(z, y) = \begin{cases} (p \cdot z)^{\frac{1}{p}} & \text{if } y \geq 0 \\ -(p \cdot z)^{\frac{1}{p}} & \text{if } y < 0 \end{cases}.$$

We can further simplify Equation (20) by again using the fact that $(\exp \circ \mathcal{W})(x) = x/\mathcal{W}(x)$:

$$(\exp \circ d)(y) = \exp\left(\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)\right)^{\frac{1}{\alpha+1}} = \left(\frac{(\alpha+1)\frac{y^{\alpha+1}}{\gamma}}{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}\right)^{\frac{1}{\alpha+1}}.$$

Thus Equation (20) simplifies to

$$g^*(y) = y \cdot b\left(\frac{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}{\alpha+1}, y\right) - \left(\frac{(\alpha+1)\frac{y^{\alpha+1}}{\gamma}}{\mathcal{W}\left((\alpha+1)\frac{y^{\alpha+1}}{\gamma}\right)}\right)^{\frac{1}{\alpha+1}}. \tag{21}$$

Since this form is quite explicit, we may well ask for what kind of function f we *can* solve Equation (19).

Here *Maple* is again useful, though a pencil-and-paper separation of variables computation will arrive at the same place. We use the built-in differential equation solver *dsolve*, subject to appropriate conditions on the parameters $f(0) = \beta$,

```
dsolve({(diff(f(x), x))^(alpha+1) = gamma*f(x),f(0)=beta}, f(x))
```

and *Maple* provides the result

$$f(x) = \left(\frac{1}{\alpha+1}\left(\alpha\, x\gamma^{(\alpha+1)^{-1}} + e^{\frac{\alpha\ \ln(\beta)}{\alpha+1}}\alpha + e^{\frac{\alpha\ \ln(\beta)}{\alpha+1}}\right)\right)^{\frac{\alpha+1}{\alpha}}.$$

In the limit as $\alpha \to 0$, *Maple* returns $f(x) = \beta(\exp(\gamma x))$ which we recognize as the familiar form of Example 3. Also, if we let

$$\alpha = 1, \text{ and } \gamma = 2$$

and ask *Maple* for the limit as β approaches 0, we recover our familiar function $f(x) = \frac{x^2}{2}$ from Example 4. Thus, we have obtained a large class of closed forms from which $f(x) = \beta \cdot \exp(\gamma x)$ (as in Example 3) and $f(x) = \frac{|x|^p}{p}, (p > 1)$ (as in Example 4) arise as special limiting cases. These are the two cases where the final closed forms of g^* turn out to be particularly clean and pleasant.

We consider first the explicit closed form of the convex conjugates for functions of form $\beta \cdot \exp(\gamma x)$. Because $g(\gamma x)$ has convex conjugate $g^*(\frac{x}{\gamma})$ (see [8, Table 3.2]), it suffices to simply show the form for the case $\gamma = 1$. In this case, $\alpha + 1 = 1$, so the form of the convex conjugate simplifies to

$$g^*(y) = \begin{cases} y\left(\log(y) - \mathcal{W}(y) - \frac{1}{\mathcal{W}(y)} - \log(\beta)\right) & \text{if } y > 0 \\ -1 & \text{if } y = 0 \\ \infty & \text{if } y < 0 \end{cases}.$$

This can be easily compared to Equation (14) from Example 4 in which case we had $\gamma = 1$. Indeed, the conjugate of $\beta f(\gamma x), \beta > 0$ is always easily computed from that of f.

Turning our attention to functions of the general form $f(x) = \frac{|x|^p}{p}, (p > 1)$, we have that $\alpha + 1 = \frac{p}{p-1}$ and $\gamma = p$, so Equation (21) becomes

$$g^*(y) = |y| \left(\left((p-1)\mathcal{W}\left(\frac{|y|^{\frac{p}{p-1}}}{p-1} \right) \right)^{\frac{1}{p}} - \left(\frac{1}{(p-1)\mathcal{W}\left(\frac{|y|^{\frac{p}{p-1}}}{p-1} \right)} \right)^{\frac{p-1}{p}} \right). \quad (22)$$

Compare this more general form of the convex conjugate to that for the specific case $p = 2$ which we saw in Equation (18) from Example 4. We can also rewrite the convex conjugate using the conjugate exponent q. Where $\frac{1}{q} + \frac{1}{p} = 1$, we have $q = \frac{p}{p-1}$ and $\frac{p}{q} = p - 1$, so Equation (22) becomes

$$g^*(y) = |y| \left(\left(\frac{p}{q} \mathcal{W}\left(\frac{q}{p} \, |y|^q \right) \right)^{\frac{1}{p}} - \left(\frac{p}{q} \mathcal{W}\left(\frac{q}{p} |y|^q \right) \right)^{-\frac{1}{q}} \right). \quad (23)$$

These simpler forms make the conjugates much easier to analyse and to compute.

Remark 3. Suppose f is *variable separable*. That is to say that

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} f_j(x_j)$$

where each f_j is convex. Then f is convex and

$$f^*(y_1, y_2, \ldots y_n) = \sum_{j=1}^{n} f_j^*(y_j).$$

From such building blocks and the Fenchel duality theorem [8, 12, 14] many other convex conjugates engaging $\mathcal{W}$ are accessible.

The conjugate has many and diverse uses. For instance, one can establish the convexity of a function through the smoothness of its conjugate. In Proposition 8 we explore one such a situation which arises as a special case of [12, Cor 4.5.2].

Proposition 8. *Suppose $f\colon X \to (-\infty, +\infty]$ is such that f^{**} is proper. Suppose f^* is Fréchet differentiable and f is lower semicontinuous. Then f is convex.*

Example 5. We illustrate Proposition 8 for

$$f(x) = 2\,|x|\,\log(2\,|x|) - 2\,|x| + 1$$

which has convex conjugate

$$f^*(y) = \begin{cases} \exp\left(\frac{y}{2}\right) - 1 & \text{if } y \geq 0 \\ \exp\left(-\frac{y}{2}\right) - 1 & \text{if } y < 0 \end{cases}.$$

More simply, $f^*(y) = \exp(|y|/2) - 1$. This is drawn in Figure 7. The function f^{**} is the convex hull of f. It is zero on the interval $[-1/\sqrt{2}, 1/\sqrt{2}]$ and agrees with f elsewhere. ◇

For more connections between the Boltzmann-Shannon entropy and the Lambert W function, see [10, p. 180].

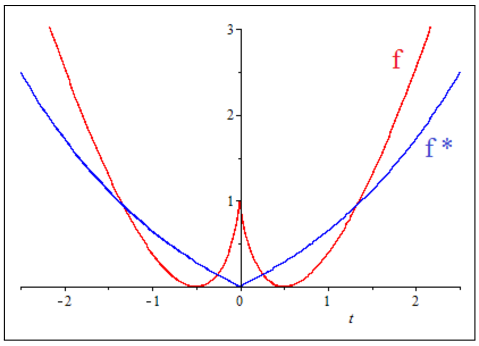


Figure 7: Here Example 5 is illustrated: f is non convex and f^* is not smooth.

4 Occurrences in Composition, Homotopy, and Infimal Convolution

Rather than computing the conjugate *ab initio* we may use one of many convex calculus rules. We choose one such rule from [8, 4.3 Exercise 12] which is especially nice for log convex functions.

Theorem 9 (Conjugates of Compositions [8]). *Consider the convex composition $h \circ g$ of a nondecreasing convex function $h : (-\infty, \infty] \to (-\infty, \infty]$ with a convex function $f \colon X \to (-\infty, \infty]$. We interpret $f(+\infty) = +\infty$, and we assume there is a point $\hat{x}$ in X satisfying $f(\hat{x}) \in \operatorname{int}\operatorname{dom}(h)$. Then for ϕ in X^*,*

$$(h \circ f)^*(\phi) = \inf_{t \geq 0} \left\{ h^*(t) + t f^*\left(\frac{\phi}{t}\right) \right\},$$

where we interpret

$$0 f^*\left(\frac{\phi}{0}\right) = \iota^*_{\operatorname{dom} f}(\phi)$$

in terms of the convex indicator *function $\iota^*_{\operatorname{dom} f}$ which is zero on* dom$f$ *and is $+\infty$ otherwise.*

We may use Theorem 9 with

$$\begin{aligned} h(t) &= \exp(t) \\ h^*(t) &= t \log t - t \text{ (the Boltzmann-Shannon entropy)} \end{aligned}$$

to compute the conjugate for $g(x) = (\exp \circ f)(x)$ with various f. We will do so in the following few examples.

Example 6 (Composition: Conjugate of $\exp \circ f$ III). Consider again the case of the function

$$g(x) = \exp\left(\frac{|x|^p}{p}\right) \text{ for } (p > 1)$$

as in Example 4. Keeping the same notation as Theorem 9, let

$$\begin{aligned} h(x) &= \exp(x) \\ f(x) &= \frac{|x|^p}{p}. \end{aligned}$$

Then we have that

$$h^*(x) = x\log x - x \text{ (the Boltzmann-Shannon entropy)}$$
$$f^*(x) = \frac{|x|^q}{q} \text{ where } \frac{1}{p} + \frac{1}{q} = 1.$$

Since $g = h \circ f$, we may solve for g^* by solving $(h \circ f)^*$. From Theorem 9 we have that, for $\phi \neq 0$,

$$(h \circ f)^*(\phi) = \inf_{t \geq 0} \left\{ h^*(t) + t f^*\left(\frac{\phi}{t}\right) \right\} = \inf_{t \geq 0} \left\{ t \log t - t + t \left(\frac{|\phi|}{t}\right)^q /q \right\}.$$

Thus, if we can find a solution for $t = s(p, \phi)$ which minimizes

$$t \log t - t + t \left(\frac{|\phi|}{t}\right)^q /q, \tag{24}$$

we will be able to substitute $s(p, y)$ for t and obtain a closed form for g^*, namely:

$$g^*(y) = s(p,y) \cdot \left(\log |s(p,y)| - 1 + \frac{1}{q} \left(\frac{|\phi|}{s(p,y)} \right)^q \right).$$

Differentiating Equation (24) with respect to t, setting the differentiated form equal to zero, and solving for t, we arrive at the optimal

$$t = s(p,y) = \exp\left(\frac{\mathcal{W}\left((q-1)|y|^q\right)}{q} \right).$$

We may then substitute this value back into the objective function in Equation (24) to obtain our closed form for g^*. The output from *Maple* appears quite complicated, but this solution may be checked to be equivalent to that expressed in Equation (23). $\diamond$

Any positively p-homogeneous convex function can be similarly treated.

Example 7 (Homotopy)**.** Consider

$$f_t(x) = (1-t)(x \log x - x) + t\frac{x^2}{2} \tag{25}$$

for $0 \leq t < 1$ so that f_0 is the Shannon entropy and f_1 is the energy. We arrive at

$$f_t^*(y) = \frac{(1-t)^2}{2t} \left(\mathcal{W}\left(\frac{t}{1-t} e^{\frac{y}{1-t}} \right) + 2 \right) \mathcal{W}\left(\frac{t}{1-t} e^{\frac{y}{1-t}} \right). \tag{26}$$

In the limit at $t = 1$ we recover the *positive energy* which is infinite for $y < 0$. In the limit at $t = 0$ we reobtain $x \log(x) - x$. $\diamond$

We will return to this function in Section 5.

Example 8 (Infimal Convolution)**.** Consider the infimal convolutions

$$g_\mu = (x \to x\log(x) - x) \,\square \left(x \to \frac{\mu x^2}{2} \right)$$

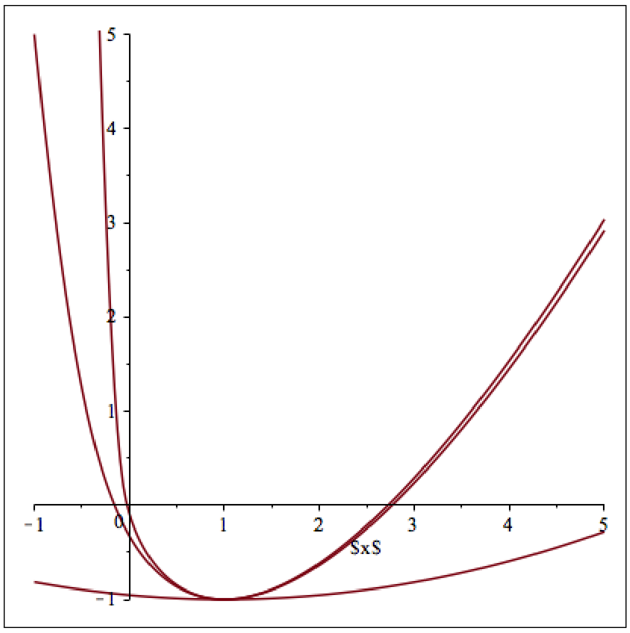


Figure 8: The convolution of entropy $x \log x - x$ and energy $\mu x^2/2$ for $\mu = 1/10, 10$, and 100.

for $\mu > 0$. This family is also called the *Moreau envelope* of $x \log(x) - x$. Then, using the `InfConv` command in SCAT we arrive at

$$g_\mu(y) = \frac{\mu}{2} y^2 - \frac{1}{\mu} \mathcal{W}(\mu e^{\mu y}) - \frac{1}{2\mu} \mathcal{W}(\mu e^{\mu y})^2.$$

and g_μ is fully explicit in terms of $\mathcal{W}$. $\diamond$

In Figure 8 we show how the infimal convolution produces a regularisation of everywhere finite approximations whose epigraphs converge back to that of $x \log x - x$ as $\mu \to +\infty$. This is a special case of the *Moreau-Yosida regularisation* or *resolvent* [14].

Again each time $\mathcal{W}$ enters very naturally indeed.

5 Homotopy and entropy solution of inverse problems

In [14, §4.7] we reprise the entropy solution of inverse problems. Consider the (negative) *entropy functional*[1] defined as follows:

$$I_f : L^1([0,1], \lambda) \to \mathbb{R} \text{ by}$$
$$I_f(x) = \int_0^1 f(x(s))\, \mathrm{d}s$$

where λ is Lebesgue measure and f is a proper, closed convex function.

[1] We chose to solve convex minimization problems rather than maximizing the entropy.

Suppose we wish to minimize I_f subject to finitely many continuous linear constraints of the form

$$\langle a_k, x\rangle = \int_0^1 a_k(s)x(s)\,\mathrm{d}s = b_k$$

for $1 \le k \le n$. We may write this as

$$\begin{aligned}&A : L^1([0,1]) \to \mathbb{R}^n \text{ by}\\ &Ax = \left(\int_0^1 a_1(s)x(s)\,\mathrm{d}s, \ldots, \int_0^1 a_n(s)x(s)\,\mathrm{d}s\right) = b.\end{aligned}$$

Here necessarily $a_k \in L^\infty([0,1], \lambda)$. When f^* is smooth and everywhere finite on the real line, our problem

$$\inf_{x \in L^1} \{I_f(x) | Ax = b\} \tag{27}$$

reduces to solving a finite nonlinear equation

$$\int_0^1 (f^*)'\left(\sum_{j=1}^n \lambda_j a_j(s)\right) a_k(s)\,\mathrm{d}s = b_k \qquad (1 \le k \le n). \tag{28}$$

The details of why all this is true are given in Section 7, and more information can be found in [9] including the matter of primal attainment and of constraint qualification.[2] See also [13], [14, §4.7], [21], and [12, Theorem 6.3.4].

Let us consider a function f_t of the form from Equation (25). If $t = 1$, then $f(x) = x^2/2$ and $(f^*)'(x) = x$ and we are actually solving the classical Gram equations for a least square problem. If $t = 0$, then f is the Shannon Entropy and $(f^*)' = \exp$. Thus we restrict to considering cases where $0 < t < 1$. Note that Equation (28) relies only on $({f_t}^*)'$. Most satisfactorily

$$(f_t^*)'(y) = \frac{(1-t)}{t}\mathcal{W}\left(\frac{t}{1-t}\exp\left(\frac{y}{1-t}\right)\right). \tag{29}$$

This is especially simple when $t = 1/2$ [8, p. 58]. As t tends to 0, we recover

$$\lim_{t\to 0} (f_t^*)'(y) = \exp(y)$$

as in the entropy case. Similarly, when t tends to 1, we obtain

$$\lim_{t\to 1} (f_t^*)'(y) = \max\{y, 0\}$$

which is the conjugate of the positive energy.

5.1 A general implementation

We illustrate by solving Equation (28) and Equation (29) for various values of t in the unit interval. We choose algebraic moments with $a_k(s) = s^{k-1}$ for

[2]When the moments are sufficiently analytic then feasibility assures the quasi-relative-interior CQ.

$1 \leq k \leq 10$ – though our methods work much more generally – and try to approximate $x(s) = \frac{6}{10} + \sin\left(3\pi s^2\right)$ given the algebraic moments

$$b_k = \int_0^1 x(s)a(s)\mathrm{d}s = \int_0^1 \left(\frac{6}{10} + \sin\left(3\pi s^2\right)\right) \cdot s^{k-1}\mathrm{d}s.$$

We do so by solving for $\lambda \in \mathbb{R}^n$ in the dual problem from Equation (28). In other words, we wish to find the values $\lambda_1 \ldots \lambda_{10}$ for which the subgradient values

$$\int_0^1 (f_t^*)' \left(\sum_{j=1}^n \lambda_j a_j(s)\right) a_k(s)\mathrm{d}s - b_k \tag{30}$$

for $k = 1..10$ all evaluate to zero. By Equation (29), our subgradient (dual problem) is represented more explicitly by the set of equations

$$\int_0^1 \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t}\exp\left(\frac{\sum_{j=1}^n \lambda_j s^{j-1}}{1-t}\right)\right) s^{k-1}\mathrm{d}s - b_k = 0. \tag{31}$$

for $k = 1 \ldots 10$. We can solve for λ using any standard numerical solver or, say, by a Newton-type method. We decide, largely for the sake of simplicity, to first use a classical Newton method. Indeed the Hessian computes nicely:

$$\begin{aligned} H(\lambda) &= (h_{i,k}) \\ h_{i,k} &= \int_0^1 \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t}\exp\left(\frac{\sum_{j=1}^n \lambda_j a_j(s)}{1-t}\right)\right) a_k(s)a_i(s)\mathrm{d}s \\ &= \int_0^1 \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t}\exp\left(\frac{\sum_{j=1}^n \lambda_j s^{j-1}}{1-t}\right)\right) s^{k+i-2}\mathrm{d}s. \end{aligned}$$

Our Hessian then turns out to be a Hankel matrix, greatly simplifying the computation. For each iteration, we need only to compute the 19 cases $k+i = 2 \ldots 20$ in order to fully populate our matrix. In fact, the gradient $G(\lambda)$ may be obtained by taking the first row (or column) of the Hessian and subtracting b_k from the kth entry. Thus, we need only to compute the Hessian and we obtain the gradient for free. So in this case there is little extra work in using a second order method. In the case of trigonometric moments we similarly arrive at a Toeplitz matrix.

5.2 Efficient computation of the dual

While there are reduced complexity methods for solving for λ as in Equation (30) with a Hankel matrix, they are less robust than more standard methods. In any event, it is advisable to solve Equation (30) without explicitly taking the inverse because taking the inverse is significantly more computationally expensive for higher order matrices.

The Newton direction is determined by solving

$$H(\lambda^n)(\mu - \lambda^n) + G(\lambda^n) = 0 \tag{32}$$

for μ and then setting $\lambda^{n+1} = \lambda^n + \alpha_n(\mu - \lambda^n)$ for an appropriate step size $\alpha_n > 0$, which in the pure Newton method is $\alpha_n = 1$. See [1] and [2] for more general options.

Since the actual computation of each of the 19 distinct Hessian terms requires numerical integration of the function

$$h_{i,k} = h_{(i+k=\alpha)} = \int_0^1 \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t} \exp\left(\frac{\sum_{j=1}^n \lambda_j s^{j-1}}{1-t}\right)\right) s^\alpha \mathrm{d}s,$$

where only the power α changes from one computation to the next, we can reduce the expense of computation quite easily.

Suppose we adopt a quadrature rule with weights $\{a_l\}_{l=1}^m$ and abcissas $\{x_l\}_{l=1}^m$. Then, where

$$F(x_l) = \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t} \exp\left(\frac{\sum_{j=1}^n \lambda_j x_l^{j-1}}{1-t}\right)\right), \tag{33}$$

for a single iteration of Newton's method we need only use numerical integration on the $\mathcal{W}$ function m times rather than order $m \cdot n$ times.

To see more clearly why this is the case, notice that we can reuse the values $a_l F(x_l)$, $l = 1 \dots m$ as follows:

$$h_{1,1} = \sum_{l=0}^m a_l F(x_l)$$
$$h_{(i+k=\alpha)} = \sum_{l=0}^m a_l F(x_l) x_l^{\alpha-2}.$$

Thus we need only compute each of them once for each iteration. We can also store each values $x_l^{\alpha-2}$ for $l = 1 \dots m$, $\alpha = 2 \dots 20$ in a matrix at the beginning.

Our optimized process is then to take a fixed[3] (Gaussian) quadrature rule and to:

1. Precompute the weights $\{a_l\}_{l=1}^m$, and the abscissas raised to various powers x_l^α, $l = 1 \dots m, \alpha = 0 \dots 18$, storing the weights in a vector and the powers of the abscissas in a matrix.

2. At each step compute the function values $a_l F(x_l)$, $l = 1 \dots m$, storing them in a vector.

3. Compute the necessary 19 Hessian values $\sum_{l=0}^m a_l F(x_l) x_l^{\alpha-2}$, $\alpha = 2 \dots 20$. If we properly create our matrix - of stored abscissa values raised to powers - we will be able to compute the Hessian values by simply multiplying our vector from Step (2) by this matrix.

4. Use the resultant 19 values to build the Hessian and gradient and then solve for the next iterate as in Equation (32).

The primal solution x_t is then given in terms of the optimal multipliers in (31):

$$x_t(s) = \frac{(1-t)}{t} \mathcal{W}\left(\frac{t}{1-t} \exp\left(\frac{\sum_{j=1}^n \overline{\lambda}_j s^{j-1}}{1-t}\right)\right). \tag{34}$$

Note that this provides a functional form for the solution at all s in $[0,1]$, not only for the quadrature points.

[3]We do not wish to allow automated decisions for adaptive methods without our control.

5.3 Some computed examples

The *Maple* code we used for computing the following examples is given in Appendix 8. For the sake of consistency, all examples in this subsection were computed with 24 digits of precision, 20 abscissas, and a Newton step size of 1/2. This reduced step dramatically improved convergence for t near 1. While this precision is higher than would be used in production code, it allows us to see the optimal performance of the algorithm.

Example 9 (Visualization Accuracy)**.** With $t = \frac{1}{2}$ and 8 moments, we ask *Maple* to compute until the error, as measured by the norm of the gradient, is less than 10^{-10}. This error is needed for a reasonable visual fit given the small number of moments used.

At 46 iterations we obtain the following values for λ:

$$[-0.7079161355, 10.64405426, -126.5979784, 656.6020449,\\ -1458.868219, 1329.347874, -299.1180785, -112.3114246]$$

where the error is about $6.84330e-11$. The associated primal solutions (functions) for iterates 6, 12, and 46 are shown in Figure 9. ◇

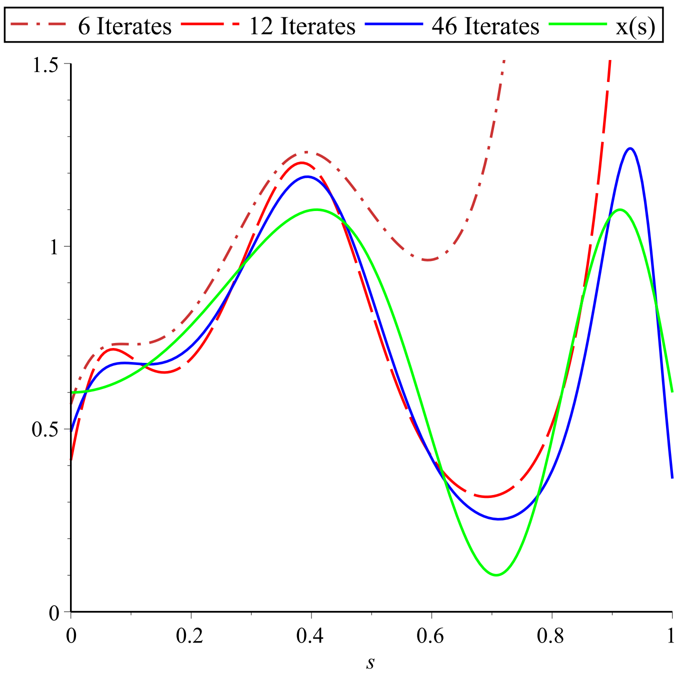


Figure 9: Example 9 illustrated for various iterates.

Example 10 (Variation of t)**.** Next we consider five different possible values for t: $0, .25, .5, .75, 1$. We run Newton's Method for each case until meeting the requirement that the norm of the gradient is less than or equal to 10^{-10}. This yields the following solutions.

t	0	.25	.5	.75	1
λ_1	-.707916	-.404828	-.101065	.204002	.512307
λ_2	10.6440	9.46383	8.23003	6.90162	5.36009
λ_3	-126.597	-114.651	-101.923	-87.8556	-70.8919
λ_4	656.602	605.686	550.755	488.934	412.561
λ_5	-1458.86	-1368.32	-1269.02	-1154.26	-1007.13
λ_6	1329.34	1282.68	1227.95	1157.70	1054.85
λ_7	-299.118	-329.937	-358.596	-381.447	-391.764
λ_8	-112.311	-85.1887	-57.6202	-30.1516	-3.12491
Error	6.84330e-11	9.81661e-11	8.26865e-11	9.6666e-11	7.05698e-11
Iterates	46	46	47	47	47

The associated primal solutions (functions) are shown in Figure 10. Notice that as t increases the visual fit increases substantially. One cannot determine this from looking at the numerical error alone. $\diamond$

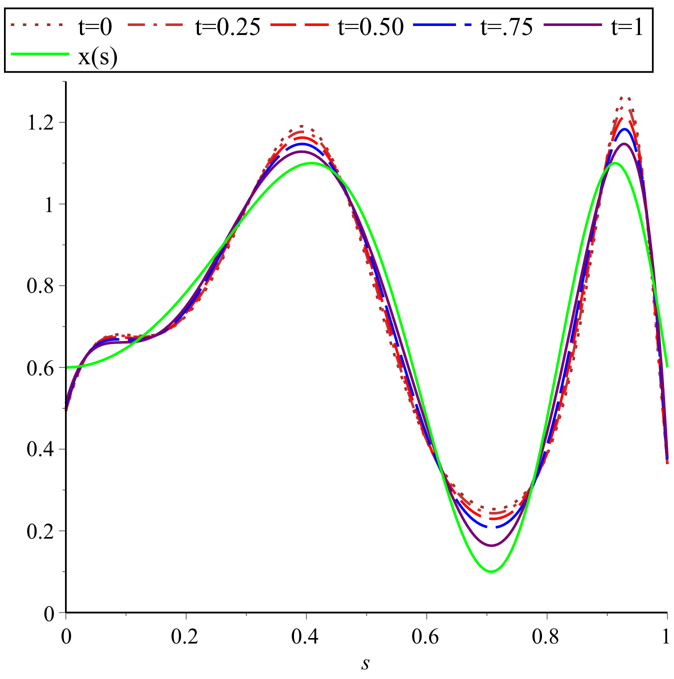


Figure 10: Example 10 is illustrated. Shown are the primal solutions corresponding to various choices of t.

Example 11 (Variation in number of moments)**.** In this example, we consider how the choice for the number of moments affects our results. Specifically, we consider the choices of 4, 8, 12, and 20 moments. We again use $t = \frac{1}{2}$ and run Newton's Method for each case until meeting the requirement that the norm of the gradient is less than or equal to 10^{-10}. It bears noting that, while we used 26 digits of precision for all of these examples in order to be consistent, this was the only case wherein we used 20 moments and so necessitated the employment of such high precision.

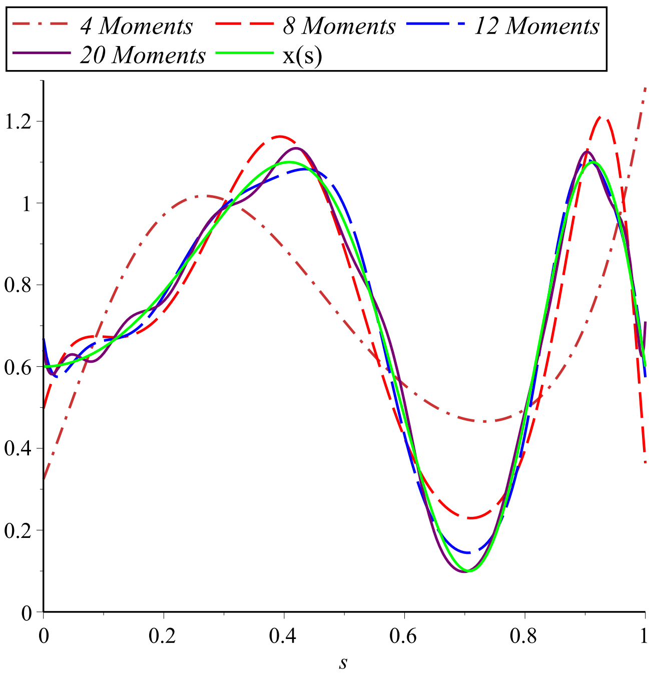


Figure 11: Example 11 is illustrated. Shown are the primal solutions corresponding to various choices for the number of moments.

Example 12 (Reconstructing a pulse)**.** We conclude by computing with a more challenging function, the pulse

$$x(s) = \chi_{[0,\frac{1}{2}]}(s).$$

The pulse is a more computationally challenging example because of its jump discontinuity – which results in a form of the Gibbs Phenomenon – and constancy on an open interval which forces some multipliers to infinity. This slowed the convergence of the gradient when we used more moments, especially for values of t nearer to 1 which more successfully reduce the phenomenon. The desired properties can still be seen visually. We use only 8 moments, and we instruct *Maple* to stop computing once the norm of the gradient is less than 10^{-10} or after reaching 200 iterates, whichever happens first.

t	0	.25	.5	.75	1
Error	6.87225e-11	7.45516e-11	9.69259e-11	1.9136e-11	.21252e-5
Iterates	70	62	55	48	200

In the case of $t = 1$, we reached 200 iterates before the norm of the gradient was less than 10^{-10}, but the primal solution we obtained is still a good proxy for the pulse. This can be seen in Figure 12, where the Gibbs Phenomenon may also be clearly observed for the the other values of t.

For $t = 0$, we see the Gibbs Phenomenon more clearly presented while the primal solution for $t = 1$ further overshoots the pulse on the right side. The other values of t afforded us by the Lambert $\mathcal{W}$ function provide other useful choices for approximating the pulse. Whether one wishes to accentuate the overshooting at the discontinuity or not depends on the application. $\diamond$

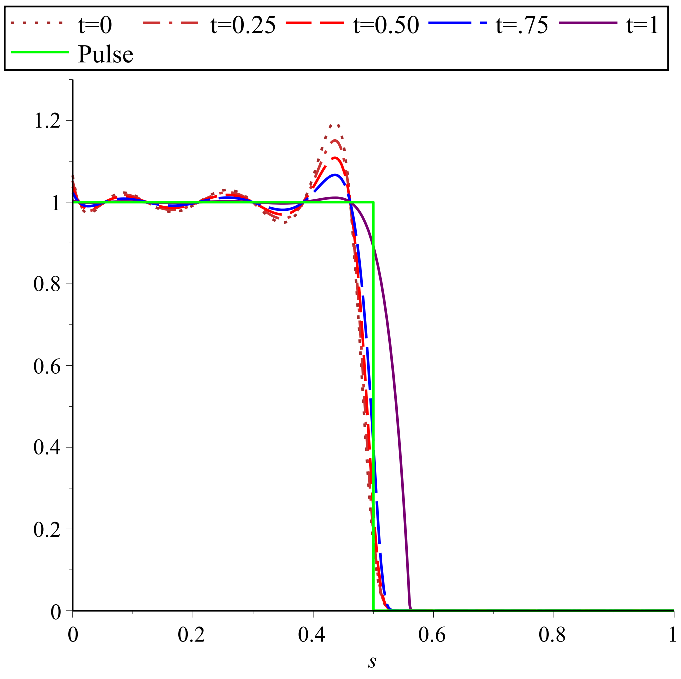


Figure 12: Example 12 is illustrated. Shown are the primal solutions corresponding to various choices for t.

6 Conclusion

We hope that we have made a good advertisement for the value of $\mathcal{W}$ in optimization and elsewhere. We note that, even when one is not able to produce a closed form for a conjugate, SCAT and its numerical partner CCAT may still be very helpful. We illustrate with two such examples.

Example 13 (Without a simple closed form)**.** Consider the function

$$f := (0, \infty) \to \mathbb{R} \text{ by } f(x) = \left(\frac{x}{e}\right)^x .$$

To examine its conjugate, we again make use of *SCAT* with the following code.

```
restart:alias(W=LambertW):
read("scat.mpl"):read("ccat.mpl"):
with(SCAT):with(CCAT):
xx:=PWF([infinity, 0, 1, (y/exp(1))^y], [y], {y::real});
```

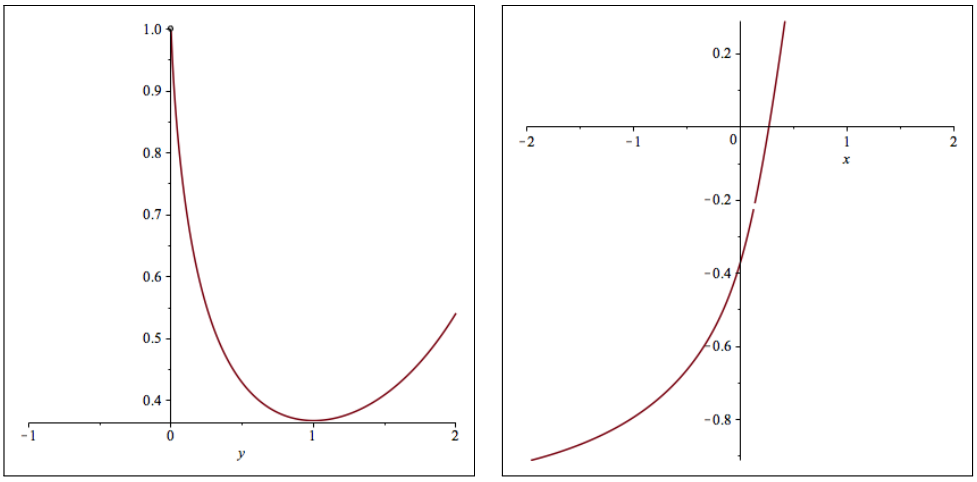


Figure 13: The function $(x/e)^x$ (L) and its conjugate (R).

.

```
Plot(xx,y=-1..2);
yy:=Conj(xx,x);
Plot(yy,x=-2..2);
```

This produces the two plots in Figure 13, even though it returns the conjugate in the unevaluated form

$$x \mapsto xR(x) - \exp(-R(x))R(x)^{R(x)}$$
$$\text{where } R(x) = RootOf(x - \exp(-z)z^z \log(z))$$

Thus *SCAT* allows us to visualize the conjugate even in the case where a closed form is not immediately forthcoming. $\diamond$

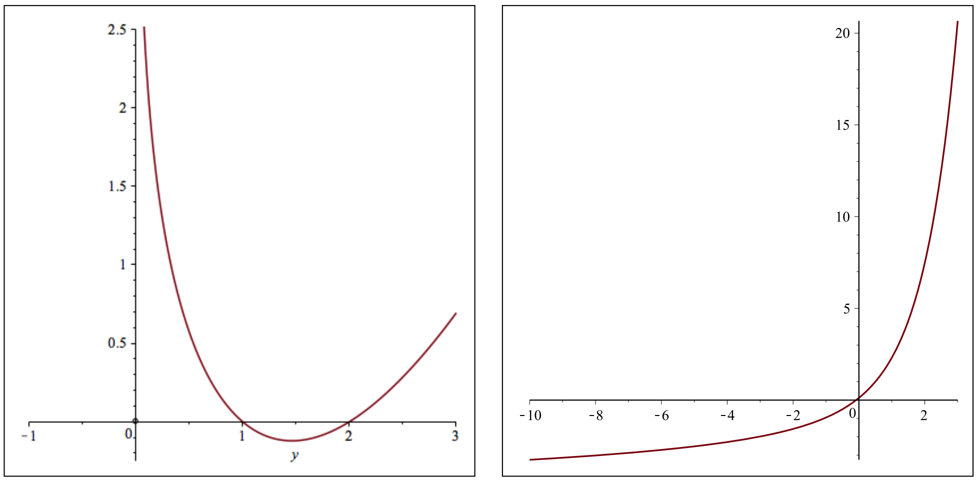


Figure 14: The function $\log \Gamma$ (L) and its conjugate (R).

.

Example 14 ($\log\Gamma$ on $(0,\infty)$ and its conjugate)**.** Likewise, for the conjugate of $\log\Gamma$, SCAT returns

$$\mathit{RootOf}\left(-\Psi\left(\_Z\right)+x\right)x-\log\left(\Gamma\left(\mathit{RootOf}\left(-\Psi\left(\_Z\right)+x\right)\right)\right).$$

Its plot is shown in Figure 14. Here Ψ is the Psi function. The 'noise' on the right is a region in which *Maple*'s built in root finder struggles. This can be obviated be a good Newton solver for a solution $x > 0$ of $\Psi(x) = y$. Set

$$x_0 = \begin{cases} \exp(y) + 1/2 & \text{if } y \geq -2.2 \\ -1/(y - \Psi(1)) & \text{otherwise} \end{cases}$$
$$x_{n+1} = x_n - \frac{\Psi(x_n) - y}{\Psi'(x_n)}.$$

Here Ψ and Ψ' are also known as the digamma and trigamma functions. *Maple* and *Mathematica* both have good built-in polygamma routines. The function and its conjugate are shown in Figure 14. $\diamond$

Finally, we note that the notion of a closed form for a given function is an always-changing issue [6]. Moreover, while $x \exp x$ is elementary $\mathcal{W}(x)$ is not, since arbitrary inversion is not permitted in the definition of *elementary*.

Acknowledgements. The authors thank Paul Vrbik for his computational assistance.

References

[1] J. Barzilai and J.M. Borwein, "Two point step-size methods," *IMA Journal on Numerical Analysis*, **8** (1988), 141–148.

[2] D. Bersetkas, "Projected newton methods for optimization problems with simple constraints, " *SIAM. J. Control and Optimization* **20**(12) (1988), 221–246.

[3] J.M. Borwein, "The SIAM 100 Digits Challenge," Extended review for the *Mathematical Intelligencer*, **27** (4) (2005), 40–48.

[4] J.M. Borwein and D.H. Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century* A.K. Peters Ltd, 2004. ISBN: 1-56881-136-5. Combined Interactive CD version 2006. Expanded Second Edition, 2008.

[5] J.M. Borwein, D.H. Bailey and R. Girgensohn, *Experimentation in Mathematics: Computational Paths to Discovery,* A.K. Peters Ltd, 2004. ISBN: 1-56881-211-6. Combined Interactive CD version 2006.

[6] J.M. Borwein and R.E. Crandall, "Closed forms: what they are and why we care." *Notices Amer. Math. Soc.* **60**:1 (2013), 50–65.

[7] J.M. Borwein and C. Hamilton, "Symbolic convex analysis: algorithms and examples," *Mathematical Programming*, **116** (2009), 17–35.

[8] J.M. Borwein and A.S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples.* Springer, (2000) (2nd Edition, 2006).

[9] J. M. Borwein and A. S. Lewis, "Duality relationships for entropy–like minimization problems." *SIAM Control and Optim.*, **29** (1991), 325–338.

[10] J. M. Borwein, S. Reich, and S. Sabach, "A characterization of Bregman firmly nonexpansive operators using a new monotonicity concept," *Journal of Nonlinear and Convex Analysis.* **12** (2011), 161–184.

[11] J. M. Borwein, A. Straub, J. Wan and W. Zudilin, with an Appendix by D. Zagier, "Densities of short uniform random walks." *Canadian. J. Math.* **64** (5), (2012), 961–990. Available at http://arxiv.org/abs/1103.2995.

[12] J.M. Borwein and J. D. Vanderwerff, *Convex Functions : Constructions, Characterizations and Counterexamples.* Cambridge University Press, (2010).

[13] J.M. Borwein and L. Yao, "Legendre-type integrands and convex integral functions," *Journal of Convex Analysis*, **21** (2014), 264-288.

[14] J.M. Borwein and Q. Zhu, *Techniques of Variational Analysis,* CMS/Springer-Verlag, 2005. Paperback, 2010.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press. 2004.

[16] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth, "On the Lambert W function," *Advances in Computational Mathematics*, **5** (1996), 329-359.

[17] T.P. Dence, "A brief look into the Lambert W function," *Applied Mathematics*, **4** (2013), 887-892.

[18] B. Fornberg and J.A.C. Weideman. "A numerical methodology for the Painlevé equations." *Journal of Computational Physics*, **230** (2011), 5957-5973.

[19] K.O. Geddes, M.L. Glasser, R.A. Moore, T.C. Scott. "Evaluation of classes of definite integrals involving elementary functions via differentiation of special functions," *Applicable Algebra in Engineering, Communication and Computing*, **1** (1990), 149-165.

[20] R.L. Graham, D.E. Knuth, O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science (2nd Edition)*. Addison-Wesley Professional. 1994.

[21] D.E. Knuth, C.C. Rousseau, "A Stirling series: 10832," *American Mathematical Monthly*, **108** (2001), 877–878. http://dx.doi.org/10.2307/2695574.

[22] S.B. Lindstrom *Understanding the quasi relative interior*. Project report on permanent reserve at Portland State University Mathematics Department Libary, (2015). http://docserver.carma.newcastle.edu.au/1682/.

[23] B.S. Mordukhovich and N.M. Nam, *An Easy Path to Convex Analysis*, Morgan & Claypool, 2014. ISBN: 9781627052375.

[24] M. Mršević. "Convexity of the inverse function," *The Teaching of Mathematics*, **XI** (2008), 21-24.

[25] S.M. Stigler, "Stigler's law of eponymy". (F. Gieryn, ed.) "Stigler's law of eponymy." *Transactions of the New York Academy of Sciences* **39** (1980), 147?58. doi:10.1111/j.2164-0947.1980.tb02775.x.

[26] Wikipedia contributors. "Lambert W function," *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Lambert_W_function&oldid=716099641

[27] Wikipedia contributors. "Meijer G-function," *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Meijer_G-function&oldid=707287736

J.M. Borwein, CARMA, University of Newcastle, Callaghan, Australia, 2308
E-mail address, J.M. Borwein: Jonathan.borwein@newcastle.edu.au

S.B. Lindstrom, CARMA, University of Newcastle, Callaghan, Australia, 2308
E-mail address, S.B. Lindstrom: c3231022@uon.edu.au

7 Appendix on Conjugate duality

To see why solving Equation (27) reduces to solving Equation (28), we recall several results. We first recall from [14, Theorem 4.7.1] that:

Proposition 10. *Suppose X is a Banach space, $F : X \to \mathbb{R} \cup \{+\infty\}$ is a lower semicontinuous convex function, $A : X \to Y$ is a linear operator, and $b \in$ core$(A\,\mathrm{dom}F)$. Then*

$$\inf_{x\in X} \{F(x)|Ax = b\} = \max_{\varphi\in\mathbb{R}^N} \{\langle\varphi, b\rangle - (F)^*(A^T\varphi)\}$$

where A^T denotes the adjoint map which satisfies

$$\langle Au, \varphi\rangle_{\mathbb{R}^n} = \langle u, A^T\varphi\rangle_{L^1}. \tag{35}$$

This allows us to reformulate many primal problems as dual problems, making their solutions simpler to compute. In particular, the problem from Equation (27) can be reformulated as

$$\inf_{x\in L^1} \{I_f(x)|Ax = b\} = \max_{\varphi\in\mathbb{R}^N} \{\langle\varphi, b\rangle - (I_f)^*(A^T\varphi)\}. \tag{36}$$

To the end of simplifying this further, we introduce another useful result. We recall from [12, Theorem 6.3.4] that:

Proposition 11. *If I_f is defined as above and $f : \mathbb{R} \to (-\infty, \infty]$ is closed, proper, and convex, we have*

$$(I_f)^* = I_{f^*}.$$

This allows us to express the dual problem more explicitly. In particular, since we have $(I_f)^* = I_{f^*}$, Equation (36) simplifies to

$$\max_{\varphi\in\mathbb{R}^N} \{\langle\varphi, b\rangle - I_{f^*}(A^T\varphi)\} = \max_{\varphi\in\mathbb{R}^N} \left\{\langle\varphi, b\rangle - \int_0^1 f^*(A^T\varphi)\right\} \mathrm{d}s. \tag{37}$$

Now in Equation (35), the inner product on the left is on $\mathbb{R}^n$ while the inner product on the right is the inner product on $L^1([0,1])$. Thus Equation (35) expands to

$$\sum_{k=1}^n \left(\varphi_k \int_0^1 a_k(s)u(s)\,\mathrm{d}s\right) = \int_0^1 (A^T\varphi)u(s)\,\mathrm{d}s.$$

This expansion should make it clear why we may simplify $A^T\varphi$ further by writing

$$A^T\varphi = \sum_{k=1}^n \varphi_k a_k(s).$$

Thus solving Equation (37) amounts to finding $\varphi \in \mathbb{R}^n$ which maximizes

$$\sum_{k=1}^n \varphi_k b_k - \int_0^1 f^*\left(\sum_{k=1}^n \varphi_k a_k(s)\right) \mathrm{d}s. \tag{38}$$

This is an equation which we can subdifferentiate. We maximize it by finding the values of φ_k for which the subdifferential with respect to φ is zero. We first recall an equivalent characterisation for the convex *subgradient*. Recall that $y \in \partial F(x)$ if

$$\langle y, z - x\rangle \leq F(z) - F(x) \text{ for all } z \in X.$$

For more preliminaries on subgradients, see, for Example, [23].

Lemma 12. *For a convex function F, y is a subgradient of F at x if and only if*

$$0 = F(x) + F^*(y) - \langle y, x\rangle.$$

Proof. Taking the negative of both sides, we simply have

$$0 = -F(x) + \langle y, x\rangle - \sup_{\overline{x}\in X}\{\langle y, \overline{x}\rangle - F(\overline{x})\}.$$

Thus we have that

$$F(x) - \langle y, x\rangle = -\sup_{\overline{x}\in X}\{\langle y, \overline{x}\rangle - F(\overline{x})\} = \inf_{\overline{x}\in X}\{F(\overline{x}) - \langle y, \overline{x}\rangle\}$$

which is equivalent to

$$\langle y, \overline{x} - x\rangle \leq F(\overline{x}) - F(x) \text{ for all } \overline{x} \in X.$$

This is the definition of the subgradient. $\square$

This makes it much easier for us to compute the subdifferential. In our context, since $(I_f)^* = I_{f^*}$, we have from that Lemma 12 that y is a subgradient of I_f at x if and only if

$$0 = I_f(x) + I_{f^*}(y) - \langle y, x\rangle = \int f(x(s)) + f^*(y(s)) - x(s)y(s)\, \mathrm{d}s.$$

Now the integrand on the right is nonnegative by Fenchel-Young and so must be zero almost everywhere. However, Lemma 12 gives us that

$$f(x(s)) + f^*(y(s)) - x(s)y(s) = 0 \text{ almost everywhere}$$

if and only if $y(s)$ is a subgradient of f at $x(s)$ for almost all s. Thus we can subdifferentiate with respect to each φ_k in Equation (38) and set equal to zero, obtaining n equations of the form

$$0 = b_k - \int_0^1 (f^*)'\left(\sum_{j=1}^n \varphi_j a_j(s)\right) a_k(s)\, \mathrm{d}s.$$

Thus we have reduced the problem to that of solving Equation (28).

8 Appendix on Computation

8.1 Construction

We present the basic construction of the code in enough detail to reproduce Example 9 and using the corresponding Initialization values. It is straightforward to adapt the basic code to reproduce the other examples. We elect, for many reasons, to use a Gaussian non-adaptive Quadrature rule. We first build an initialization list wherein the user can specify the parameters of the example they wish to construct. We specified the entries as follows.

List Entry	Significance
[1]	Number of moments
[2]	Choice of t
[3]	Number of abscissas and weights
[4]	Digits of precision
[5]	Digits to display
[6]	Stop computing more iterates when the norm of the gradient is less than or equal to 10 to the negative of this value
[7]	Save the value of λ at this iterate in order to print an example.
[8]	Save the value of λ at this iterate in order to print an example.
[9]	Stop computing more iterates if the number of iterates computed reaches this number.
[10]	Newton step size

We show the code with the user entries from Example 9.

```
Initialization:=[8,1/2,20,26,10,10,6,12,50,1/2]:
Digits:=Initialization[4]:
interface(displayprecision=Initialization[5]):
```

Computing the abscissas and their corresponding weights is easy using *Maple*'s built-in Legendre polynomials in the *orthopoly* package.

```
with(orthopoly): abscissas:= fsolve(P(Initialization[3],x)=0,x):
weights:=NULL:
for i from 1 to Initialization[3] do
    expr := (x-abscissas[j])/(abscissas[i]-abscissas[j]);
    expr := (product(expr, j = 1 .. i-1))*(product(expr, j = i+1 ..
        Initialization[3]));
    f[i] := unapply(expr, x);
    c[i] := int(f[i](x), x = -1 .. 1);
    weights := weights, c[i]
od:
```

Now because our integral is over $[0, 1]$ and we must use the interval $[-1, 1]$ for our Gaussian quadrature, we apply the these weights and abscissas to a *translation* of the abscissa values raised to powers and a *translation and scaling* of the function from Equation (33):

$$\frac{1}{2}F\left(\frac{1}{2}(x_l+1)\right)$$
$$\left(\frac{1}{2}(x_l+1)\right)^{\alpha-2}.$$

We define the function we wish to approximate, store a list of the moments, and compute the matrix of abscissa values raised to powers as follows.

```
Objective:=y -> .6+.5*sin(3.1415926*3*y^2):
b:=convert(Vector(Initialization[1], i-> add((1/2)*weights[j]*
     Objective((1/2)*abscissas[j]+1/2)*((1/2)*abscissas[j]+1/2)^(i-1),
     j = 1..Initialization[3])), list):
M := Matrix(2*Initialization[1]-1, Initialization[3], (i,j) ->
     ((1/2)*abscissas[j]+1/2)^(i-1) ):
```

We construct a function *WeightedF* which takes in a list λ and a t value and computes the values $a_l F(x_l)$, $l = 1 \dots m$ returning them as a row vector.

```
F := proc (c, t, s)
    local N, output;
    N := nops(c);
    if t = 0 then
        output := (1/2)*exp(add(c[j]*((1/2)*s+1/2)^(j-1), j = 1 .. N));
    elif t = 1 then
        output := (1/2)*max(add(c[j]*((1/2)*s+1/2)^(j-1), j = 1 .. N), 0);
    else
        output := (1/2)*(1-t)*LambertW(t*exp(add(c[j]*((1/2)*s+1/2)^(j-1),
            j = 1 .. N)/(1-t))/(1-t))/t;
    fi;
    output;
end:

WeightedF := proc (c, t)
    local j, output, weight;
    output := Vector[row](Initialization[3]);
    for j to Initialization[3] do
        weight := weights[j];
        output[j] := weight*F(c, t, abscissas[j]);
    od;
    output;
end:
```

We can now easily obtain the 19 values we need to populate our Hessian matrix and gradient at λ by asking *Maple* for the value

```
M.WeightedF(lambda,Initialization[2])^(%T)
```

where we recall that *Initialization*[2] is our chosen value for t. We create a function which constructs our Hessian matrix procedurally.

```
HessianBuilder:=proc(c,t)
    local A,H;
    A:=M.WeightedF(c,t)^(%T);
    H:=Matrix(Initialization[1],Initialization[1],(i,j)->A[i+j-1]);
end;
```

Recalling that our gradient is just the first row of our Hessian with b_k subtracted from the kth entry for $k = 1 \dots m$, we can build our gradient from the Hessian in the Newton procedure. We construct the Newton procedure with the following code. Instead of a list, it receives a vector as its input and passes a converted list to the previous functions. This nuance allows for the indexing to be much simpler in all of the prior scripting.

```
with(LinearAlgebra):
NextIteration := proc (c, t)
    local H, G, RHS, output, mu, L;
    L := convert(c, list);
    H := HessianBuilder(L, t);
    G := Vector(Initialization[1], j-> H[1, j]-b[j];
    RHS := H.c-G;
    mu := LinearSolve(H, RHS);
    output := c+Initialization[10]*(mu-c);
    output, G;
end:
```

Notice that this function returns a list, the first element of which is the next iterate and the second element of which is the value of the gradient at the current iterate, both in vector form. The reason we have built the function to return this pair is that the gradient values for each iterate are, in fact, the L^2 norms of that iterate's distance from each of the respective moments. This provides a useful gauge for checking our convergence, although it is far from conclusive (recall Example 10). We use it in the construction of the complete Newton procedure.

```
Newtons := proc (c, t, userlimit)
    local pair, iterate, err, counter, grad;
    iterate := c;
    counter := 0;
    err := 1;
    to userlimit while is(10^(-Initialization[6]) <= err) do
        pair := NextIteration(iterate, t);
        iterate := pair[1];
        grad := pair[2];
        err := Norm(pair[2]);
        counter := counter+1
    od;
    iterate, grad, err, counter;
end:
```

We have built our Newton's Method procedure to return a list which consists of our approximation, the gradient value at the previous iterate, the norm of said gradient, and the number of iterations it calculated before terminating. We can now, for instance, recreate the numbers from Example 9 with the following code.

```
start:=Vector(Initialization[1],j->1):
Newtons(start, Initialization[2], Initialization[9])
```

Maple returns our solution, our gradient at the second to last step, its corresponding norm, and the number of steps it ran for. Recalling that where Newton's method returns $\overline{\lambda}$ - in this case as the first entry of our list - our primal solution x_t is given by Equation (34). We may write a procedure which takes our return from Newton's Method and creates a primal solution.

```
Primal := proc (c, t, s)
```

```
    local sumterm, L, output;
    L := convert(c, list);
    sumterm := add(L[j]*s^(j-1), j = 1 .. Initialization[1]);
    if t = 0 then
        output := exp(sumterm);
    elif t = 1 then
        output := max(sumterm, 0);
    else
        output := (1-t)*LambertW(t*exp(sumterm/(1-t))/(1-t))/t
    fi;
    output;
end:
```

This procedure - which we have appropriately named "Primal" - takes in a list and returns a function in one variable, s, when given our solution from the Newton procedure as the first item and our choice of t as the second item.

```
start:=Vector(Initialization[1],j->1):
solution:=Newtons(start, Initialization[2], Initialization[9])[1];
PRIM:=Primal(solution,Initialization[2],s);
```

In the above code, *PRIM* is a function in a single variable s. We can use the procedures *Primal* and *Newtons* to recreate Figure 9. The construction is straightforward.

```
GenerateExample := proc (c, t, userlimit)
    local PRIM1, PRIM2, PRIM3, finish, N3, IMG;
    PRIM1 := Newtons(c, t, Initialization[7]);
    PRIM1 := PRIM1[1];
    PRIM2 := Newtons(c, t, Initialization[8]);
    PRIM2 := PRIM2[1];
    PRIM3 := Newtons(c, t, userlimit);
    N3 := PRIM3[4];
    PRIM3 := PRIM3[1];
    plot([Primal(PRIM1, t, s), Primal(PRIM2, t, s), Primal(PRIM3, t, s),
            Objective(s)],
        s = 0 .. 1, 0 .. 1.5,
        color = [orange, red, blue, green],
        linestyle = [dashdot, longdash, solid, solid],
        legend = [typeset(Initialization[7], " Iterates"),
            typeset(Initialization[8], " Iterates"),
            typeset(N3, " Iterates"), typeset("x(s)")],
        legendstyle = [location = top]);
end:
```

Reconstructing Figure 9 is then as easy as asking for

```
GenerateExample(Vector(Initialization[1]j->1),Initialization[2],Initialization[9])
```

A similar procedure can be written to recreate Example 10. Some re-designing is necessary in order to accommodate the creation of examples with varying moments (such as Example 11), but the fundamental aspects of optimizing the performance remain the same.