Journal of Statistical Software

February 2006, Volume 15, Issue 7. *http://www.jstatsoft.org/*

Introducing elliptic, an R Package for Elliptic and Modular Functions

Robin K. S. Hankin
National Oceanography Centre, Southampton

Abstract

This paper introduces the **elliptic** package of R routines, for numerical calculation of elliptic and related functions. Elliptic functions furnish interesting and instructive examples of many ideas of complex analysis, and the package illustrates these numerically and visually. A statistical application in fluid mechanics is presented.

Keywords: elliptic functions, modular functions, Weierstrass elliptic functions, visualization of complex functions.

1. Introduction

The elliptic functions crop up here and there in diverse areas of applied mathematics such as cosmology (Kraniotis and Whitehouse 2002), chemical engineering (Koopman and Lee 1991), dynamical systems (Kotus and Urbański 2003), and quantum mechanics (Chow 2002); here they are applied to fluid mechanics (Johnson and McDonald 2004, 2005). They also provide interesting and instructive non-elementary examples of many results in complex analysis such as Cauchy's integral theorem and the residue theorem.

In this paper I introduce **elliptic**, a new R package for numerical calculation of Weierstrass and Jacobi elliptic functions, theta functions and modular functions. The emphasis is on efficient numerical calculation, and informative visualization techniques.

The package is available on CRAN (`http://CRAN.R-project.org/`, R Development Core Team 2005).

2. Elliptic functions

This section gives a very brief introduction to elliptic functions. For more detail and rigorous derivations, the reader is referred to the classic literature: the standard reference would

be Whittaker and Watson (1952). Chandrasekharan (1985) approaches the field from a more modern perspective, and Abramowitz and Stegun (1965) provide the definitive reference work for the case of real invariants.

A meromorphic function f is said to be elliptic if $\exists\, \omega_1, \omega_2 \in \mathbb{C}$ with $\omega_2/\omega_1 \in \mathbb{C}\backslash\mathbb{R}$ such that

$$f(z) = f(z + 2m\omega_1 + 2n\omega_2) \tag{1}$$

whenever $f(z)$ is defined and $m, n \in \mathbb{Z}$. Notation in this paper is consistent with that of Abramowitz and Stegun (1965); ω_1 and ω_2 are called the *half periods*. In 1862, Weierstrass introduced his $\wp$ function which is defined as

$$\wp(z) = \frac{1}{z^2} + \sum_{\substack{m,n\in\mathbb{Z}\\ m,n\neq 0}} \left\{ \frac{1}{(z - 2m\omega_1 - 2n\omega_2)^2} - \frac{1}{(2m\omega_1 + 2n\omega_2)^2} \right\}. \tag{2}$$

The $\wp$ function is, in a well defined sense, the simplest nontrivial elliptic function (Whittaker and Watson 1952). Given this, we have a Taylor expansion of the form

$$\wp(z) - z^{-2} = \frac{1}{20} g_2 z^2 + \frac{1}{28} g_3 z^4 + O(z^6) \tag{3}$$

with

$$g_2 = 60 \sum{}' \frac{1}{(2m\omega_1 + 2n\omega_2)^4}, \qquad g_3 = 140 \sum{}' \frac{1}{(2m\omega_1 + 2n\omega_2)^6}, \tag{4}$$

where a prime indicates summation over $\mathbb{Z}^2$ excluding $(m, n) = (0, 0)$. For reasons to be made clear in Section 4.1.1, g_2 and g_3 are known as the *invariants*. Other equivalent forms for $\wp$ include its differential equation

$$\left(\frac{d\wp}{dz}\right)^2 = 4\wp^3 - g_2\wp - g_3 \tag{5}$$

and the relation

$$z = \int_{t=w}^{\infty} \frac{dt}{\sqrt{4t^3 - g_2 t - g_3}} \tag{6}$$

which is equivalent to $w = \wp(z)$.

Related functions include the zeta function $\zeta(z)$, defined by

$$\frac{d\zeta(z)}{dz} = -\wp(z) \tag{7}$$

and the sigma function $\sigma(z)$, defined by

$$\frac{d \log \sigma(z)}{dz} = \zeta(z), \qquad \lim_{z \longrightarrow 0} \left[\frac{\sigma(z)}{z}\right] = 1 \tag{8}$$

(neither $\sigma(z)$ nor $\zeta(z)$ is elliptic). It may be shown that $\zeta(z)$ is analytic except for points on the lattice of periods, at which it has simple poles with residue 1. One classic result is due to Legendre: if ω_1, ω_2 is a pair of basic periods[1], with $\mathrm{Im}(\omega_2/\omega_1) > 0$, then

$$\eta_1\omega_2 - \eta_2\omega_1 = \pi i \tag{9}$$

[1] A pair of basic periods is one that generates the period lattice. Basic periods are not unique as many pairs of periods may generate the same lattice. However, there is one pair of basic periods, the *fundamental* periods that are, in a well-defined sense, optimal (Chandrasekharan 1985).

where $\eta_1 = \zeta(\omega_1)$ and $\eta_2 = \zeta(\omega_2)$.

2.1. Jacobian elliptic functions

Jacobi approached the description of elliptic functions from a different perspective (Weisstein 2005). Given $m = k^2$ and m_1 with $m + m_1 = 1$, Jacobi showed that if

$$u = \int_{t=0}^{\phi} \frac{dt}{\sqrt{(1-t^2)(1-mt^2)}}$$

the functions $\mathrm{sn}(u,k)$, $\mathrm{cn}(u,k)$ and $\mathrm{dn}(u,k)$ defined by

$$\mathrm{sn}u = \sin\phi, \qquad \mathrm{cn}u = \cos\phi, \qquad \mathrm{dn}u = \sqrt{1 - k^2\sin^2\phi} \tag{10}$$

are elliptic with periods

$$K = \int_{\theta=0}^{\pi/2} \frac{d\theta}{\sqrt{1 - m\sin^2\theta}} \tag{11}$$

and

$$iK' = i\int_{\theta=0}^{\pi/2} \frac{d\theta}{\sqrt{1 - m_1\sin^2\theta}}. \tag{12}$$

The Jacobian elliptic functions are encountered in a variety of contexts and bear a simple analytical relation with the Weierstrass $\wp$ function.

3. Numerical evaluation and Jacobi's theta functions

Although Equation 2 is absolutely convergent, it converges too slowly to be of use in practical work, and an alternative definition is needed.

Jacobi presented his four theta functions in 1829 and, although they have interesting properties in their own right, here they are used to provide efficient numerical methods for calculation of the elliptic functions above. They are defined as follows:

$$\theta_1(z,q) = 2q^{1/4}\sum_{n=0}^{\infty}(-1)^n q^{n(n+1)}\sin(2n+1)z \tag{13}$$

$$\theta_2(z,q) = 2q^{1/4}\sum_{n=0}^{\infty} q^{n(n+1)}\cos(2n+1)z \tag{14}$$

$$\theta_3(z,q) = 1 + 2\sum_{n=1}^{\infty} q^{n^2}\cos 2nz \tag{15}$$

$$\theta_4(z,q) = 1 + 2\sum_{n=1}^{\infty}(-1)^n q^{n^2}\cos 2nz \tag{16}$$

It may be seen that, provided $|q| < 1$, the series converges for all $z \in \mathbb{C}$. Indeed, the convergence is very rapid: the number of correct significant figures goes as the square of the number of terms. It should be noted that there are different notations in use, both for the four function names, and for the independent variables.

All the functions described in Section 2 may be expressed in terms of the theta functions. This is the default method in **elliptic**, although alternative algorithms are implemented where possible to provide a numerical and notational check.

For example, Weierstrass's $\wp$ function is given by

$$\wp(z) = \frac{\pi^2}{4\omega_1^2}\left(\frac{\theta_1'(0,q)\theta_2(v,q)}{\theta_2(0,q)\theta_1(v,q)}\right)^2 \tag{17}$$

where $q = e^{i\omega_2/\omega_1}$; the other functions have similar theta function definitions.

4. Package elliptic in use

This section shows **elliptic** being used in a variety of contexts. First, a number of numerical verifications of the code are presented; then, elliptic and related functions are visualized using the function `view()`; and finally, the package is used to calculate flows occurring in an oceanographic context.

The primary function in package **elliptic** is `P()`: this calculates the Weierstrass $\wp$ function, and may take named arguments that specify either the invariants `g` or half periods `Omega`:

```
> z <- 1.9 + (0+1.8i)
> P(z, g = c(1, 0+1i))

[1] -0.550851-2.404205i

> P(z, Omega = c(1, 0+1i))

[1] -12.01771-15.97636i
```

4.1. Numerical verification

Work in the field of elliptic functions is very liable to mistakes[2], and package **elliptic** includes a number of numerical checks to guard against notational inexactitude. These checks generally use the convenient (trivial) function `maxdiff()` that shows the maximum absolute difference between its two arguments:

```
> maxdiff <- function(x, y) {
+     max(abs(x - y))
+ }
```

For example, we may compare the output of `P()`, which uses Equation 17, against the straightforward Laurent expansion, used by `P.laurent()`:

```
> g <- c(3, 2 + (0+4i))
> z <- seq(from = 1, to = 0.4 + (0+1i), len = 34)
```

[2] Abramowitz and Stegun (1965) state that there is a "bewildering" variety of notations in use; the situation has become more confusing in the intervening 40 years.

```
> maxdiff(P(z, g), P.laurent(z, g))
```

```
[1] 1.510067e-15
```

showing reasonable agreement; note that function `P()` uses the conceptually distinct theta function formula of Equation 17. Package **elliptic** includes a large number of such numerical verification tests in the `test` suite provided in the package, but perhaps more germane is the inclusion of named identities appearing in Abramowitz and Stegun (1965). For example, consider function `e18.10.9()`, named for the equation number of the identity appearing on page 650. This function returns the difference between the (algebraically identical) left and right hand side of three grouped identities:

$$\begin{aligned} 12\omega_1^2 e_1 &= \pi^2\left[\theta_3^4(0,q)+\theta_4^4(0,q)\right] \\ 12\omega_1^2 e_2 &= \pi^2\left[\theta_2^4(0,q)-\theta_4^4(0,q)\right] \\ 12\omega_1^2 e_3 &= -\pi^2\left[\theta_3^4(0,q)+\theta_4^4(0,q)\right] \end{aligned} \tag{18}$$

where $q = e^{-\pi K'/K}$ is the nome. From the manpage:

```
> abs(e18.10.9(parameters(g = g)))
```

```
[1] 5.218048e-15 2.009718e-14 2.316086e-14
```

again showing reasonably accurate numerical results, but perhaps more importantly explicitly verifying that the notational scheme used is internally consistent.

Although the examples above use the invariants `g2` and `g3` to define the elliptic function and its periods, sometimes the fundamental periods are known and the invariants are desired. This is done by function `g.fun()`, which takes the fundamental periods as arguments and returns the two invariants g_2 and g_3. Observe that there are many pairs of basic periods that generate the same lattice—see Figure 1—but it usual to specify the unique *fundamental periods* as this pair usually has desirable numerical convergence properties.

Unimodularity

Many functions of the package are *unimodular*. The invariants g_2 and g_3 are defined in terms of a pair of basic periods ω_1 and ω_2. However, any pair of basic periods should have the same invariants, because any pair of basic periods will define the same elliptic function (hence the name). Basic period pairs are related by a unimodular transformation: if ω_1, ω_2 and $\tilde{\omega}_1, \tilde{\omega}_2$ are two pairs of basic periods then there exist integers a, b, c, d with $ad - bc = 1$ and

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \tilde{\omega}_1 \\ \tilde{\omega}_2 \end{pmatrix}$$

Formally, a unimodular function $f(\cdot,\cdot)$ is one with arity 2—it is conventional to write $f(\mathbf{v}) = f(a,b)$—and for which

$$f(\mathbf{v}) = f(\mathbf{M}\mathbf{v}) \tag{19}$$

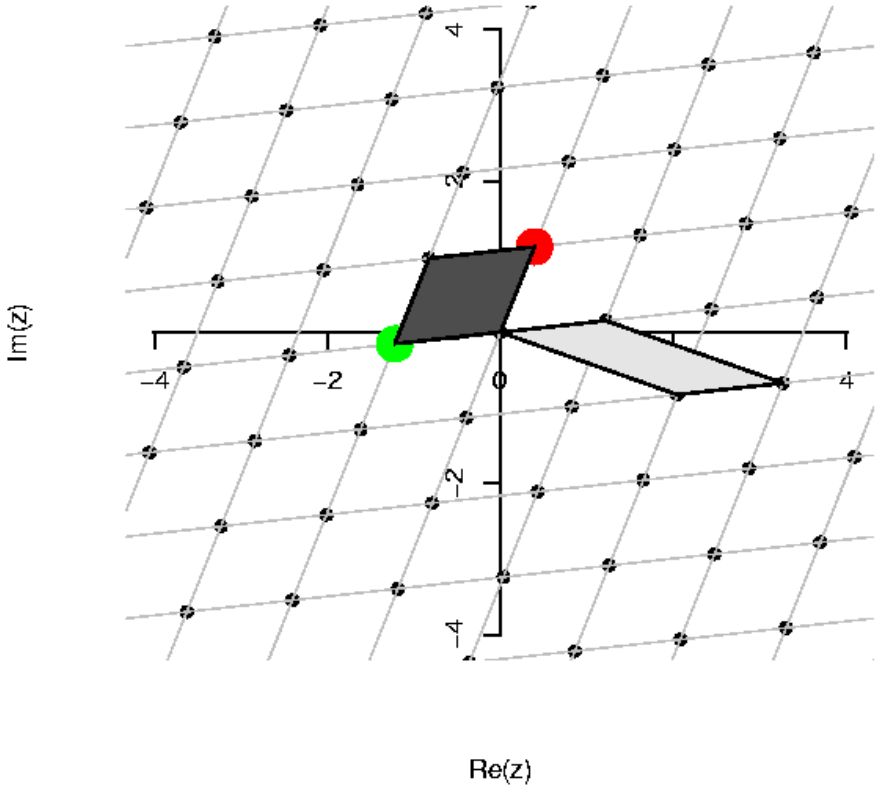

Figure 1: The lattice generated by $\wp(z; 1+i, 2-3i)$; fundamental period parallelogram shown in light gray and a basic period parallelogram shown in darker gray

where $\mathbf{M}$ is unimodular: that is, an integer matrix with a determinant of unity. In this context, unimodular matrices (and the transformations they define) are interesting because any two pairs of basic periods are related by a unimodular transformation.

The package includes functions that generate unimodular matrices. The underlying function is `congruence()`, which generates 2×2 integer matrices with a determinant of 1, given the first row. For example:

```
> M <- congruence(c(4, 9))

     [,1] [,2]
[1,]    4    9
[2,]    3    7
```

(observe that the determinant of $\mathbf{M}$ is unity) and thus we may verify the unimodularity of, for example, `g.fun()` by evaluating the invariants for a pair of fundamental periods, and comparing this with the invariants calculated for a pair of basic periods that are related to the fundamental periods by a unimodular transformation (here $\mathbf{M}$). In R idiom:

```
> o <- c(1, 0+1i)

> maxdiff(g.fun(o), g.fun(M %*% o, maxiter = 800))

[1] 1.036805e-13
```

showing that the invariants for period pair $o = (1, i)^T$ are almost identical to those for period pair $o' = \mathbf{M}o = (4 + 9i, 3 + 7i)^T$. Observe that the latter evaluation requires many more iterations for accurate numerical evaluation: this behaviour is typically encountered when considering periods whose ratio is close to the real axis.

In addition, function `unimodular()` generates unimodular matrices systematically, and function `unimodularity()` checks for a function's being unimodular.

Contour integration and the residue theorem

As noted in Section 2, the zeta function $\zeta(z)$ possesses a simple pole of residue 1 at the origin. The residue theorem would imply that

$$\oint \zeta(z)\, dz = 2\pi i$$

when the contour is taken round a path that encircles the origin but no other poles. This may be verified numerically using **elliptic**'s `myintegrate` suite of functions, which generalize the **stats** package's `integrate()` function to the complex plane. Here, function `integrate.contour()` is used to integrate round the unit circle. This function takes three arguments: first, the function to be integrated; second, a function that describes the contour to be integrated along; and third, a function that describes the derivative of the contour. We may now integrate over a closed loop, using arguments `u` and `udash` which specify a contour following the unit circle:

```
> jj <- integrate.contour(Zeta, u, udash)

> maxdiff(jj, 2 * pi * (0+1i))

[1] 9.003349e-16
```

showing reasonable numerical accuracy. Compare Weierstrass's $\wp$ function, which has a second order pole at the origin:

```
> abs(integrate.contour(WeierstrassP, u, udash))

[1] 2.482534e-16
```

The PARI system

Perhaps the most convincing evidence for numerical accuracy and consistency of notation in the software presented here is provided by comparison of the package's results with that of PARI (Batut, Belabas, Bernardi, Cohen, and Olivier 2000). The PARI system is an open-source project aimed at number theorists, with an emphasis on pure mathematics; it includes some elliptic function capability. Function `P.pari()` of package **elliptic** calls the `pari` system directly to evaluate elliptic functions from within an R session, enabling quick verification:

```
> omega <- c(1, 0+1i)
> z <- seq(from = pi, to = pi * (0+1i), len = 10)

[1] 2.760239e-14
```

again showing reasonable agreement, this time between two independent computational systems.

4.2. Visualization of complex functions

In the following, a Weierstrass elliptic function with invariants of $1 + i$ and $2 - 3i$ will be considered. The half periods ω_1, ω_2 are first evaluated:

```
> jj.omega <- half.periods(g = c(1 + (0+1i), 2 - (0+3i)))
```

and these may be visualized by using `latplot()`, as in Figure 1. Figure 2 shows the real part of such a function, shown over part of the complex plane, and Figure 3 shows the same function using the `view()` function.

The σ function with the same invariants is visualized in Figure 4, showing that its zeros lie on the same lattice as Figure 1.

Figure 5 shows the zeta function, and Figure 6 shows Jacobi's "sn" function.

4.3. Potential flow

One application of complex analysis is to fluid dynamics. In particular, potential flow (steady, two-dimensional, inviscid, incompressible) may be studied with the aid of analytic complex

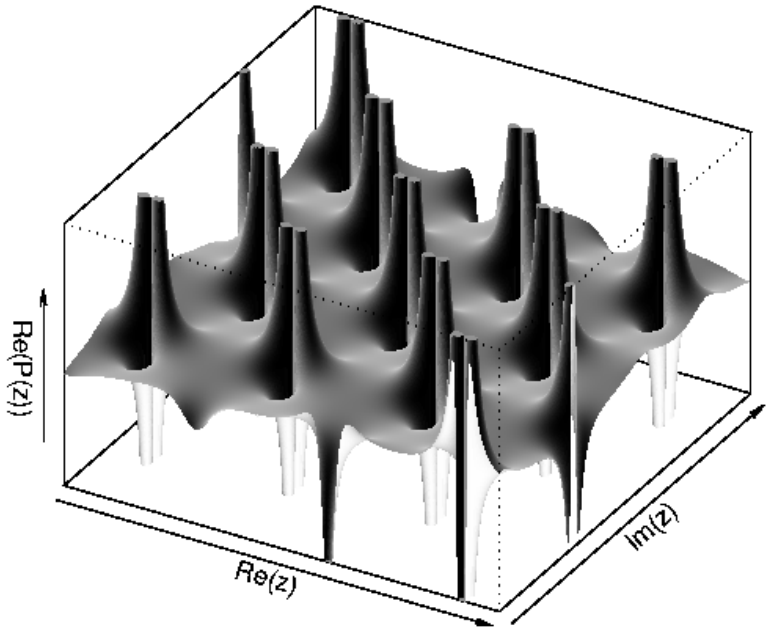

Figure 2: Real part of $\wp(z, 1, 1 + 2i)$. Note the second order poles at each lattice point

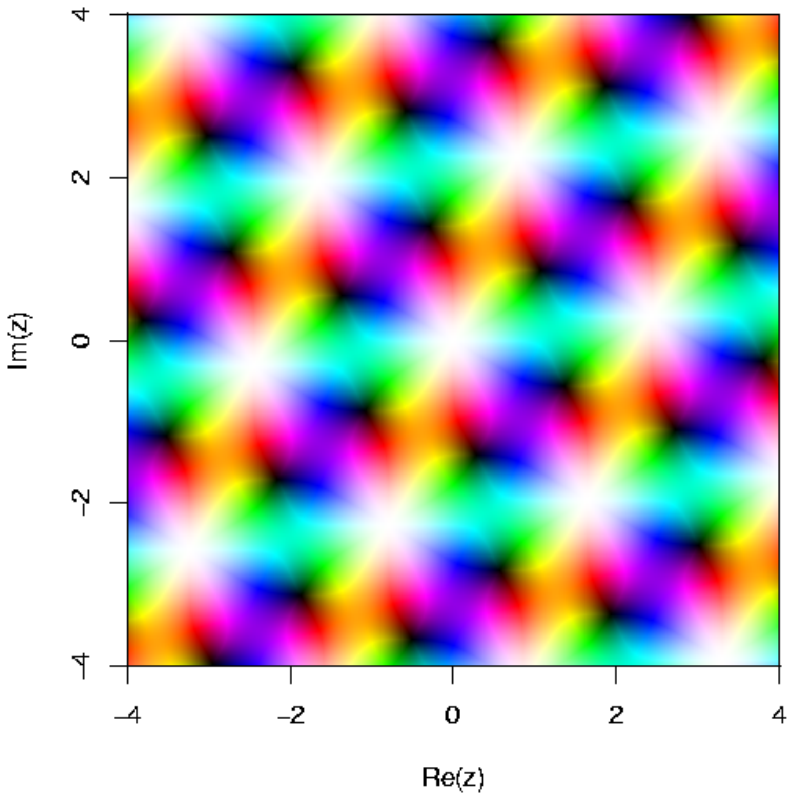

Figure 3: Visualization of $\wp(z, 1, 1+2i)$ using the scheme of Thaller (1998): white corresponds to a pole, black to a zero, and full saturation to $|\wp(z)| = 1$. The poles of $\wp(z)$ occur on a regular lattice, and the zeroes on two shifted lattices. Note how each of the poles is surrounded by two cycles of hue, indicating that they are of second order; and each of the zeroes is surrounded by one cycle of hue, indicating that they are simple roots

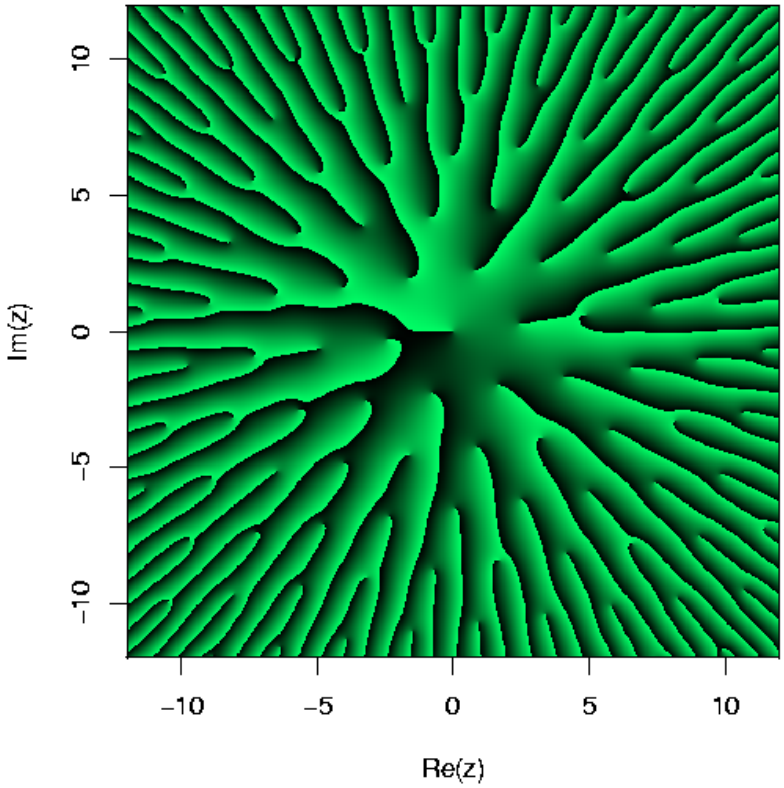

Figure 4: Visualization of $f = \sigma(z, 1, 1 + 2i)$ using `view()`; colour indicates $\mathrm{Arg}(f)$. Thus points at which $f(z)$ is on the negative real axis, that is $\{z : f(z) \in \mathbb{R}^-\}$, are visible as discontinuities of (colourimetric) value. These discontinuities are semi-infinite; note that the zeros of f occur, at the (finite) end of each line, on a regular lattice. As $|z|$ increases, each discontinuity threads its way through an increasing number of other discontinuities and zeros, and the spacing between the discontinuities becomes less and less

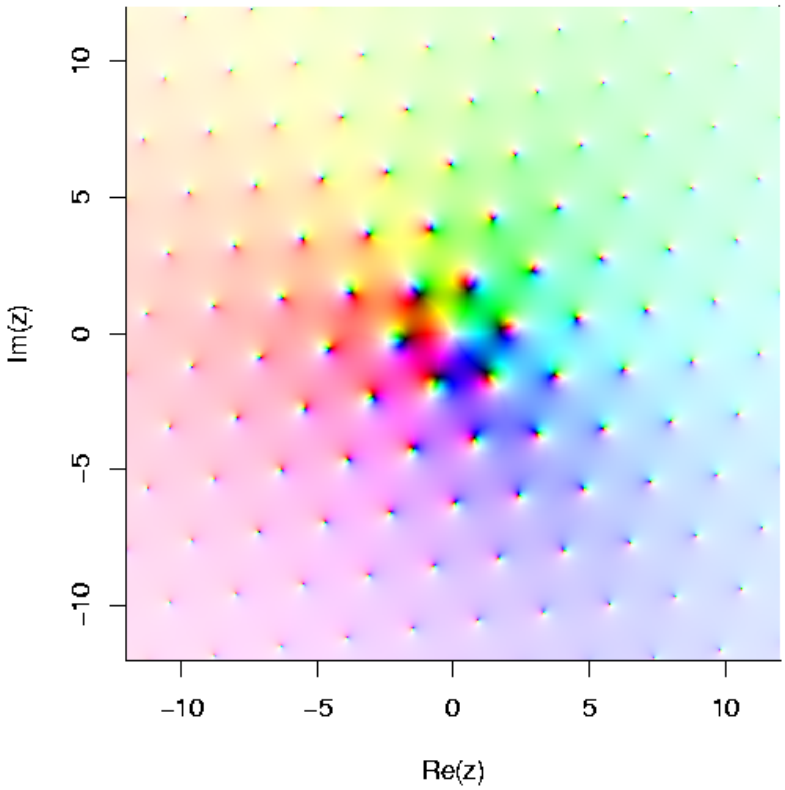

Figure 5: Visualization of $\zeta(z, 1, 1 + 2i)$ using `view()` and the colouring scheme of Thaller. Poles appear as white regions, and zeroes as black regions. Each pole is of single order, each zero is a simple root (one cycle of hue). The poles occur on a lattice; there is no simple pattern to the zeroes. Note the overall tendency towards the edges of the square to whiteness: $|f|$ increases with $|z|$ as per Equation 7

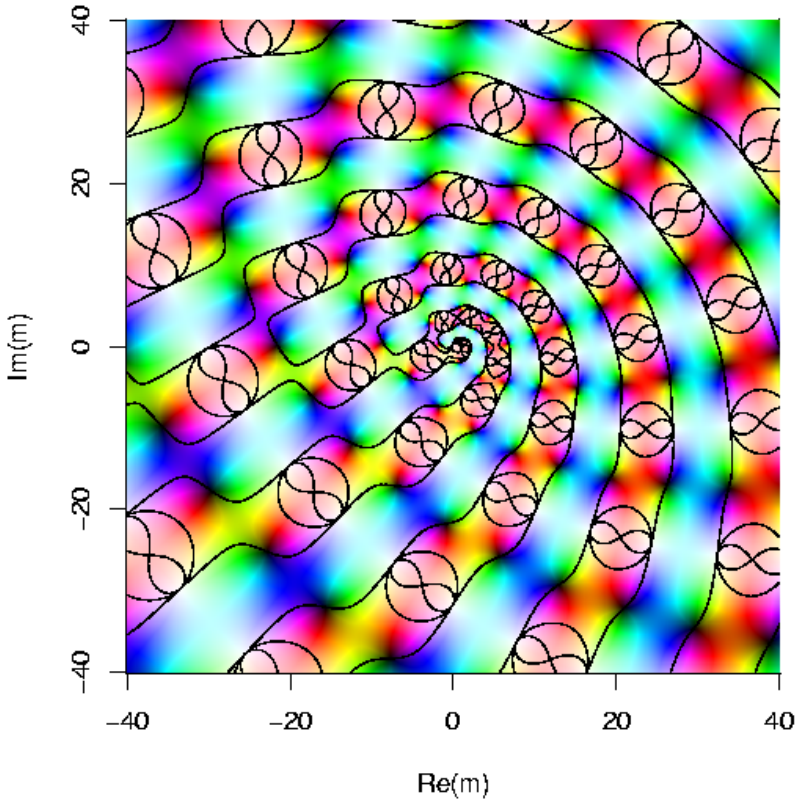

Figure 6: Jacobi's "sn" function using the elliptic package. Here, $f = \mathrm{sn}(5-2i, m)$ is visualized, with background utilizing Thaller's scheme, and contours of equal $\mathrm{Re}(f)$ at three selected values shown as black lines. Note the aperiodic arrangement of poles (white areas) and zeroes (black areas)

functions. Here I show how the elliptic functions discussed in this paper may be used to simulate potential flow in a rectangular domain.

Although the tenets of potential flow appear to be absurdly idealized[3], it is nevertheless a useful technique in many branches of practical fluid mechanics: it is often used to calculate a "theoretical" flowfield with which measured velocities may be compared. A short sketch of potential theory is given here but the reader is referred to Milne-Thomson (1949) for a full exposition. Briefly, we define a *complex potential* $w(z)$ to be a complex function

$$w(z) = \phi + i\psi$$

and observe that both ϕ and ψ obey Laplace's equation if w is differentiable. Given this, we may take the velocity vector $\mathbf{v} = (v_x, v_y)$ of the fluid to be

$$v_x = \frac{\partial \phi}{\partial x}, \qquad v_y = \frac{\partial \phi}{\partial y},$$

and observe that streamlines are given by contours of equal ψ; contours of equal ϕ are called equipotential lines. The two systems of lines cross at right angles (this follows from the Cauchy-Riemann conditions).

Consider, for example, the function $w(z) = z^2$, whose associated flow field is shown in Figure 7. This corresponds to a stagnation point, at which the speed vanishes; the streamlines (solid) intersect the equipotential lines (dashed) at right angles.

Now consider a slightly more complicated case. A point source of strength m at z_0 may be represented by the function

$$m \log(z - z_0)$$

(a sink corresponds to $m < 0$). Any finite number of sources or sinks may be combined, as in $\sum_i m_i \log(z - z_i)$ where the i^{th} source is at z_i and has strength m_i, because the system is linear[4]. Figure 8 shows two sources and two sinks, all of equal strength. Because the flowfield is symmetric with respect to the real axis, there is no flux across it; we may therefore ignore the flow in the lower half plane (ie $\{z : \text{Im(z)} < 0\}$) and consider the flow to be bounded below by the real axis. This is known as *the method of images* (Milne-Thomson 1949).

Now, one may transform a potential flowfield into another form using a conformal mapping from the z- plane to the ζ- plane, traditionally denoted

$$\zeta = f(z).$$

This technique finds application when flow is desired (in the ζ- plane) that obeys some specific boundary condition that is simple to specify in the z- plane.

In this case, we seek a conformal transformation that maps the upper half plane to a rectangle. If we consider the flowfield shown in Figure 8, then the map given by

$$\zeta = \int \frac{dz}{\sqrt{(1 - a^2 z^2)(1 - z^2)}}$$

[3] Feynman, Leighton, and Sands (1966) famously described potential flow as the study of "dry water"

[4] It is often more convenient to work with the algebraically equivalent form $\log\left(\prod (z - z_i)^{m_i}\right)$, as it is easier to deal with the branch cuts

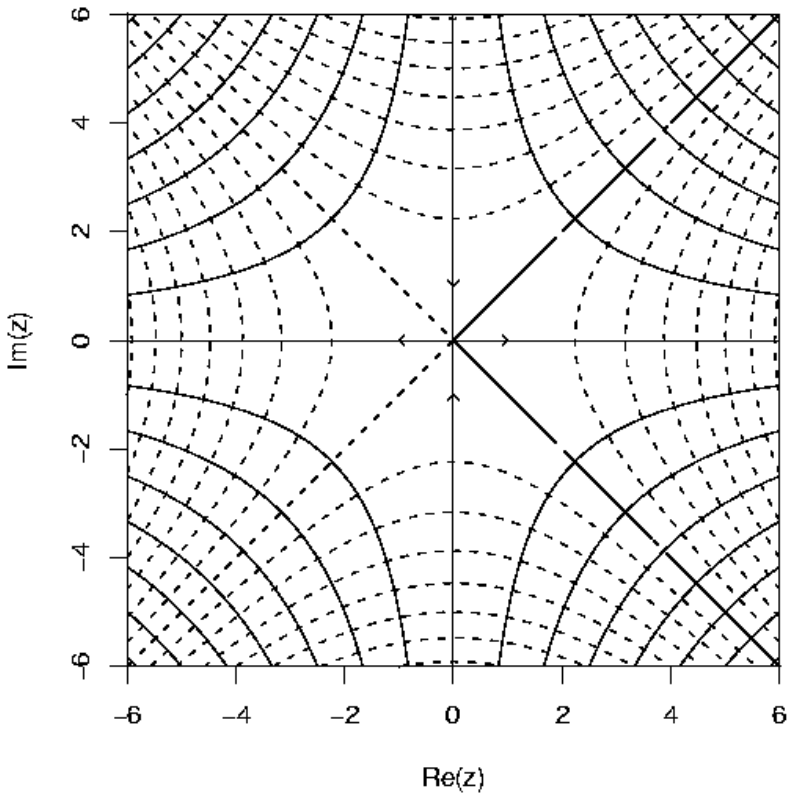

Figure 7: Potential flow on the complex plane: field corresponding to the function $(z) = z^2$. Solid lines represent streamlines and dotted lines represent equipotentials; these intersect at right angles. Note stagnation point at the origin

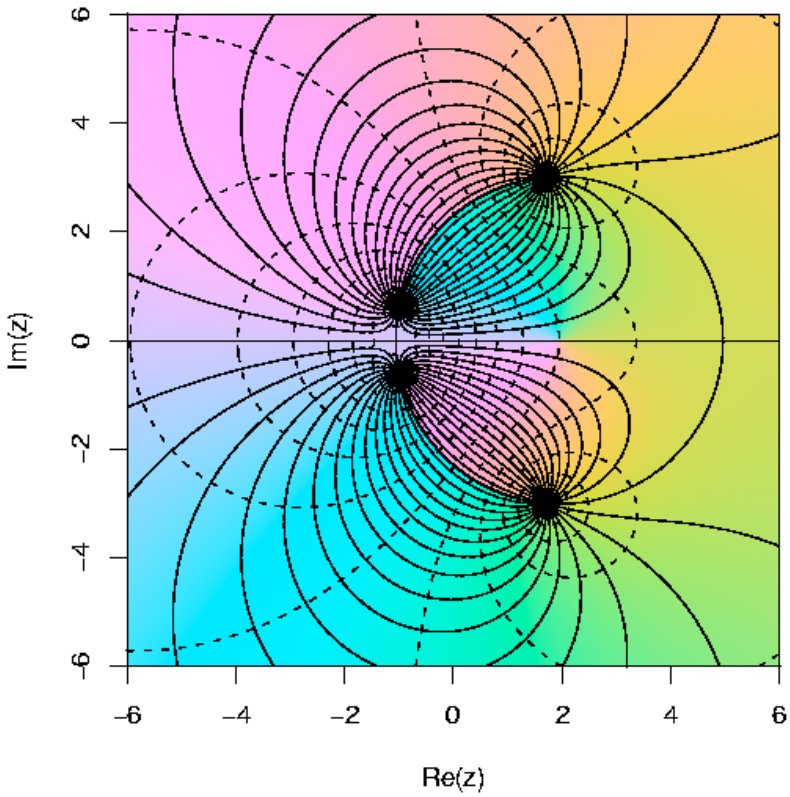

Figure 8: Potential flow in on the complex plane: two sources and two sinks, all of equal strength. Solid lines denote streamlines, dotted lines equipotentials; colour scheme uses the `hcl()` system: hue varies with the argument, and chrominance varies with the modulus, of the potential. There is no flux between the lower and the upper half plane, but there is flux out of, and in to, the diagram. Note the stagnation point at approximately $5 + 0i$

takes the upper half plane of the ζ- plane to a rectangle in the z- plane (Milne-Thomson 1949). Using Equation 10, this is equivalent to $z = \mathrm{sn}(\zeta; m)$, where $\mathrm{sn}(\cdot, \cdot)$ is a Jacobian elliptic function and m a constant of integration.

Figure 9 shows the resulting flow field: observe how the flow speed, which is proportional to the spacing between the streamlines, is very small near the left-hand edge of the rectangle.

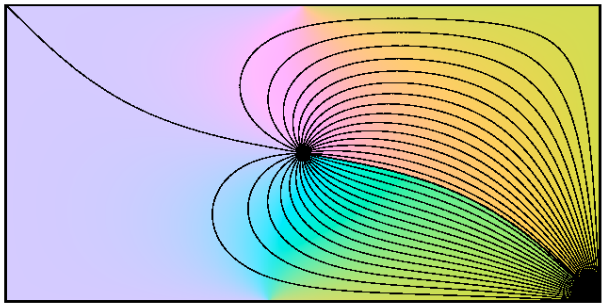

Figure 9: Potential flow in a rectangle of aspect ratio 2: source and sink of equal strength. Colour scheme as in Figure 8. Note the dividing streamline which terminates in a stagnation point on the rectangle boundary

4.4. Bayesian analysis of potential flow

When considering potential flows, it is often necessary to infer the locations of singularities in the flow from sparse and imperfect data (Johnson and McDonald 2004).

Here, I apply the methods of Kennedy and O'Hagan (2001a) and Kennedy and O'Hagan (2001b) (hereafter KOH and KOHa respectively) using the **BACCO** package (Hankin 2005) to assess some characteristics of potential flow in a rectangle.

Kennedy and O'Hagan considered the following inference problem for a set of parameters $\theta \in \mathcal{R}^q$ that are inputs to a computer program. Given an independent variable $x \in \mathcal{R}^n$, and a set of scalar ("field") observations $z = z(x)$, they assume

$$z(x) = \rho \cdot \eta(x, \theta) + \delta(x) + \epsilon \tag{20}$$

where ρ is a constant of proportionality (notionally unity); $\eta(\cdot, \cdot)$ a Gaussian process with unknown coefficients; θ the true, but unknown parameter values; $\delta(\cdot)$ a model inadequacy term (also a Gaussian process with unknown coefficients); and $\epsilon \sim N(0, \lambda^2)$ uncorrelated normal observational errors.

Inferences about $\eta(\cdot, \cdot)$ are made from point observations of the process: Kennedy and O'Hagan call these the "code observations" on the grounds that their chief motivation was the understanding of complex computer codes.

Here, potential flow in a rectangle is considered. The source is at one corner of the rectangle,

which is considered to have lower left point $(-1, 0)$ and upper right point $(1, 1)$. The position of the sink is unknown.

I now show how the position of the sink may be inferred from a sparse and noisy set of observed fluid speeds. Similar inference problems are encountered in practice when considering oceanographic flows such as those occurring near deep sea vents, although the geometry is generally more complex than considered here.

The independent variable $\mathbf{x}$ is the two-dimensional position within the rectangle, and the field observation $z(\mathbf{x})$ is the fluid speed at that point, plus observational error ϵ. The parameter set θ thus has two degrees of freedom corresponding to the $x-$ and $y-$ coordinates of the sink.

Field observations will be obtained numerically, using the **elliptic** package. The simulated flowfield has a sink at a *known* position—in this case the geometric centre of the rectangle—and Bayesian methods will be used to infer its position using only fluid speed data.

In the terminology of KOHa, dataset `y` corresponds to modelled fluid speed, obtained from the appropriate simulations carried out with the sink placed at different locations within the rectangle. Dataset `z` corresponds to field observations, which in this case is fluid speed at several points in the rectangle, obtained from simulations with the sink at the centre of the rectangle.

The code evaluation design matrix `D1` is chosen according to a random Latin hypercube design, and the observation is calculated using the **elliptic** package:

```
> head(D1)
```

```
         x    y x.sink y.sink observation
[1,]  0.77 0.35  -0.17   0.15       -1.87
[2,] -0.50 0.45   0.77   0.22       -4.58
[3,] -0.30 0.55  -0.43   0.05       -4.34
[4,]  0.03 0.95   0.97   0.58        0.51
[5,]  0.57 0.15   0.83   0.75       -2.35
[6,] -0.63 0.68   0.10   0.65       -4.47
```

So the first line shows a simulation with the sink at (-0.17,0.15); the log of the fluid speed at (0.77, 0.35) is -1.87. There are a total of 30 such observations. Figure 10 shows these points superimposed on the "true" flow field.

The field observations are similarly determined:

```
> head(D2)
```

```
         x    y observation
[1,]  0.26 0.63       -2.50
[2,]  0.90 0.53       -2.24
[3,] -0.06 0.31       -0.15
[4,] -0.13 0.40       -3.66
[5,] -0.71 0.44       -4.60
[6,]  0.97 0.15        0.81
```

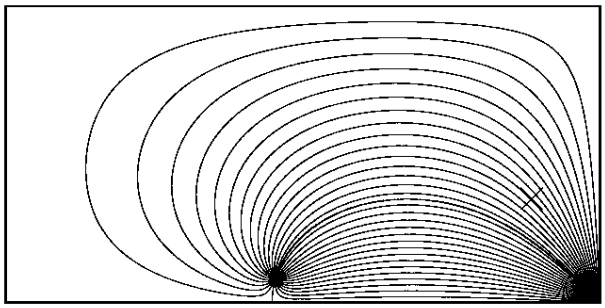

Figure 10: Streamlines of first code observation point; field observation point shown as a cross. The sink is at (-0.17,0.15)

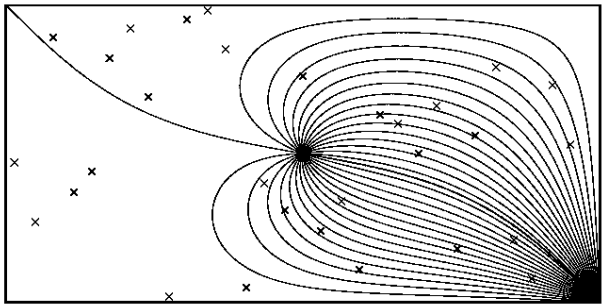

Figure 11: Streamlines of "true" flow; field observation points shown as crosses

showing that the first field observation, at (0.26, 0.63), is -2.5. There are a total of 31 such observations. Figure 11 shows the first code observation in the context of the "true" flow field.

Kennedy and O'Hagan give, *inter alia,* an expression for the likelihood of any value of θ being the true parameter set (in this case, the true position of the sink) in terms of the code evaluations and field observations.

Here, function `support()` calculates the log-likelihood for a pair of coordinates of the sink. This may be evaluated at the centre of the rectangle, and again at the top left corner:

```
> support(c(0, 1/2))

[1] 0

> support(c(-1, 1))

[1] -52.29362
```

showing, as expected, that the support is very much larger at the centre of the rectangle than the edge (here the arbitrary additive constant is such that the support at `c(0,1/2)` is exactly zero). It is now possible to identify the position of the sink that corresponds to maximum support using numerical optimization techniques:

```
> (mle <- optim(c(0, 1/2), support))

[1] -0.1842530  0.6190823
```

Thus the maximum likelihood estimate for the sink is a distance of about 0.2 from the true position. The support at this point is about 3.9 units of likelihood:

```
> support(mle)

[1] 3.908104
```

Discussion of Bayesian statistical analysis

The above example shows the ideas of KOH being applied straightforwardly, but with the novel twist of θ being interpreted as physical characteristics of a fluid flow. In this case θ is the coordinates of the sink.

The MLE is better supported than the true position by about 3.9 units of likelihood: thus, in the language of Edwards (1992), the hypothesis of $\theta_{\text{true}} = (0, 0.5)$ would not be rejected if one accepted Edwards's 2 units of likelihood per degree of freedom.

The discrepancy between $\hat{\theta}$ and θ_{true} (a distance of about 0.2) may be due to the coarseness of the form adopted for the basis functions, and better results might be obtained by using a more sophisticated system of model inadequacy than the simple linear form presented here.

The methods of KOH allow one to make statistically robust statements about the physical characteristics of an interesting flow that are difficult to make in any other way.

5. Conclusions

Elliptic functions are an interesting and instructive branch of complex analysis, and are frequently encountered in applied mathematics: here they were used to calculate a potential flow field in a rectangle.

This paper introduced the R package **elliptic**, which was then used in conjunction with Bayesian statistical methods (the **BACCO** bundle) to make statistically sound inferences about a flow with uncertain parameters: in this case the position of the sink was estimated from a sparse and noisy dataset.

Acknowledgments

I would like to acknowledge the many stimulating and helpful comments made by the R-help list over the years.

References

Abramowitz M, Stegun IA (1965). *Handbook of Mathematical Functions*. New York: Dover.

Batut C, Belabas K, Bernardi D, Cohen H, Olivier M (2000). "User's Guide to PARI/GP." Technical Reference Manual. url: `http://www.parigp-home.de/`.

Chandrasekharan K (1985). *Elliptic Functions*. Springer-Verlag.

Chow KW (2002). "A Class of Doubly Periodic Waves for Nonlinear Evolution Equations." *Wave Motion*, **35**, 71–90.

Edwards AWF (1992). *Likelihood (Expanded Edition)*. Johns Hopkins.

Feynman RP, Leighton RB, Sands M (1966). *The Feynman Lectures on Physics*, volume 1. Addison Wesley.

Hankin RKS (2005). "Introducing **BACCO**, an R package for Bayesian Analysis of Computer Code Output." *Journal of Statistical Software*, **14**(16).

Johnson ER, McDonald NR (2004). "The Motion of a Vortex Near Two Circular Cylinders." *Proceedings of the Royal Society of London A*, **460**, 939–954.

Johnson ER, McDonald NR (2005). "Vortices Near Barriers With Multiple Gaps." *Journal of Fluid Mechanics*, **531**, 335–358.

Kennedy MC, O'Hagan A (2001a). "Bayesian Calibration of Computer Models." *Journal of the Royal Statistical Society B*, **63**(3), 425–464.

Kennedy MC, O'Hagan A (2001b). "Supplementary Details on Bayesian Calibration of Computer Models." *Technical report*, University of Sheffield. URL `http://www.shef.ac.uk/~st1ao/ps/calsup.ps`.

Koopman DC, Lee HH (1991). "Second-Order Reversible Reactions and Diffusion in a Slab-Like Medium: An Application of the Weierstrass Elliptic Pe Function." *Chemical Engineering Science*, **46**(4), 1165–1177.

Kotus J, Urbański M (2003). "Hausdorff Dimension and Hausdorff Measures of Julia sets of Elliptic Functions." *Bulletin of the London Mathematical Society*, **35**, 269–275.

Kraniotis GV, Whitehouse SB (2002). "General Relativity, the Cosmological Constant and Modular Forms." *Classical and Quantum Gravity*, **19**, 5073–5100.

Milne-Thomson LM (1949). *Theoretical Hydrodynamics*. Macmillan, second edition.

R Development Core Team (2005). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Thaller B (1998). "Visualization of Complex Functions." *The Mathematica Journal*, **7**(2), 163–180.

Weisstein EW (2005). "Jacobi Elliptic Functions." From Mathworld–a Wolfram Web Resource, URL http://mathworld.wolfram.com/JacobiEllipticFunctions.html.

Whittaker ET, Watson GN (1952). *A Course of Modern Analysis*. Cambridge University Press, fourth edition.

Affiliation:

Robin K. S. Hankin
National Oceanography Centre, Southampton
European Way
Southampton SO14 3ZH, United Kingdom
E-mail: r.hankin@noc.soton.ac.uk
URL: http://www.noc.soton.ac.uk/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 15, Issue 7 *Submitted:* 2005-07-13
February 2006 *Accepted:* 2006-01-18