Practical Considerations in The Calculation of Kelvin Functions *Ber(x), Bei(x), Ber'(x)* and *Bei'(x)* And Complete Elliptic Integrals *K* and *E*

Robert S. Weaver

October 26, 2009

Part I - Kelvin Functions

To implement the Kelvin functions in Basic, an excellent starting reference is H. B. Dwight's *Tables of Integrals and Other Mathematical Data* (4th Edition, MacMillan, 1961). He gives series formulae for these functions as follows:

$$Ber(x) = 1 - \frac{(\frac{1}{2}x)^4}{(2!)^2} + \frac{(\frac{1}{2}x)^8}{(4!)^2} - \dots \tag{820.3}$$

$$Bei(x) = \frac{(\frac{1}{2}x)^2}{(1!)^2} - \frac{(\frac{1}{2}x)^6}{(3!)^2} + \frac{(\frac{1}{2}x)^{10}}{(5!)^2} - \dots \tag{820.4}$$

$$Ber'(x) = -\frac{(\frac{1}{2}x)^3}{1!2!} + \frac{(\frac{1}{2}x)^7}{3!4!} - \frac{(\frac{1}{2}x)^{11}}{5!6!} + \dots \tag{820.5}$$

$$Bei'(x) = \frac{1}{2}x - \frac{(\frac{1}{2}x)^5}{2!3!} + \frac{(\frac{1}{2}x)^9}{4!5!} - \dots \tag{820.6}$$

(For this discussion the terms in these series will be numbered starting at zero.)

The pattern in the terms is readily apparent. The exponent in the numerator increases by 4 for each successive term. The patterns in the denominators are slightly different for each function, but again, are readily apparent. One can implement a loop in Basic to calculate each term and then sum them, skipping out of the loop when the terms become vanishingly small. However, one must be careful in the way that the terms are calculated. There is a natural tendency to want to evaluate the numerator fully, then the denominator, and then divide the former by the latter. Unfortunately, for values of x>1, the value of the terms of the numerator get very large very quickly and can cause overflows. Likewise, the squared factorials in the denominator will get very large even faster than the numerator. So, even though the value of the term itself may be small, the numerator and denominator calculations may fail due to overflow.

A better method is to calculate them incrementally. For example, term 2 of *Ber*(x), expanded is:

$$\frac{(\frac{1}{2}x)^8}{(4!)^2} = \frac{\frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x}{1 \times 2 \times 3 \times 4 \times 1 \times 2 \times 3 \times 4}$$

We can evaluate it like this:

$$(\frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x \times \frac{1}{2}x)/(1 \times 2 \times 3 \times 4 \times 1 \times 2 \times 3 \times 4)$$

or like this:

$$(\frac{\frac{1}{2}x}{1}) \times (\frac{\frac{1}{2}x}{2}) \times (\frac{\frac{1}{2}x}{3}) \times (\frac{\frac{1}{2}x}{4}) \times (\frac{\frac{1}{2}x}{1}) \times (\frac{\frac{1}{2}x}{2}) \times (\frac{\frac{1}{2}x}{3}) \times (\frac{\frac{1}{2}x}{4})$$

Using the first method, the numerator and denominator get unwieldy very quickly. Using the second method, the values of the individual fractions remain relatively manageable.

This also presents an opportunity to make the calculation computationally more efficient by developing a recurrence relation for the terms. That is, we can define *term n* as a function of *term n*-1.

Expanding the first three terms of *Ber*(x), simplifying the fractions, and ignoring the sign for the time being, we get:

$$term_0 = 1$$

$$term_1 = (\frac{x}{2}) \times (\frac{x}{4}) \times (\frac{x}{2}) \times (\frac{x}{4})$$

$$term_2 = (\frac{x}{2}) \times (\frac{x}{4}) \times (\frac{x}{6}) \times (\frac{x}{8}) \times (\frac{x}{2}) \times (\frac{x}{4}) \times (\frac{x}{6}) \times (\frac{x}{8})$$

It is apparent that term 2 can be calculated by starting with term 1 and multiplying it by:

$$(\frac{x}{6}) \times (\frac{x}{8}) \times (\frac{x}{6}) \times (\frac{x}{8})$$

And in general, each new term n can be calculated by starting with the previous term and multiplying it by a factor which we will call *termi*, which will be:

$$\frac{x}{4n-2} \times \frac{x}{4n} \times \frac{x}{4n-2} \times \frac{x}{4n}$$

In this way, we don't have to calculate each new term in its entirety, only the incremental part consisting of the four sub-factors shown. The savings in computation become significant when the number of terms to be calculated becomes large.

The general algorithm then, for calculating *Ber*(x) will be as follows:

1. Explicitly calculate $term_0$ as the starting *term*.
2. Set an initial value for the *sign* (+1 or -1), which will alternate for each *term* calculated.
3. Set the initial *sum* of the series equal to $term_0 \times sign$.
4. Using a For-Next loop, perform the remaining calculations:
5. Set *sign=-sign*
6. Calculate incremental *termi* and multiply the previous value of *term* by this value to get the new *term*.
7. Set $sum=sum+sign \times term$
8. Exit the loop when the *term* becomes vanishingly small.

This algorithm is implemented in Open Office Basic as follows:

```
Function Ber(ByVal x as double) as double
  ' Calculates Ber(x) function
  ' Uses recurrence relation based on H.B. Dwight's
  ' series expansion formula 820.3
  ' Basic code by Robert Weaver 2009-10-26
  Dim i,sign As integer
  Dim sum,termi,term As Double
  if x=0 then
    Ber()= 1
  else
    term=1
    sign=1
    sum=term*sign
    '300 iterations is enough to calculate any
    'value within the range of double precision
    for i = 1 to 300
      sign=-sign
      termi=((x/(4*i)) * (x/(4*i-2)))^2
      term=term*termi
      sum=sum+sign*term
      'Skip out of loop if current term < 1e-12 of sum
      if abs(term/sum)<1e-12 then exit for
    next
    Ber()= sum
  end if
end Function
```

The only item that needs further comment is the first IF statement which checks for the case of x=0. In this situation the function simply returns the correct value of 1 rather than executing the For-Next loop. This is not strictly necessary for the *Ber*(x) function, but for the remaining three functions it prevents a divide by zero error in the convergence test at the end of the loop.

By pasting this code into the Open Office Macro Editor, the function may then be used in a spreadsheet formula in the same way as a built-in function.

The remaining functions are calculated in exactly the same way. The only lines of code which are different are for the initial value of *term*, the *sign* value, the formula for the recurrence relation, $term_i$, and the value returned for the case of x=0. The code for $Bei(x)$, $Ber'(x)$ and $Bei'(x)$ follows:

```
Function Ber_(ByVal x as double) as double
  ' Calculates Ber'(x) function -- d/dx Ber(x)
  ' Uses recurrence relation based on H.B. Dwight's
  ' series expansion formula 820.5
  ' Basic code by Robert Weaver 2009-10-26
  Dim i,sign As integer
  Dim sum,termi,term As Double
  if x=0 then
    Ber_()= 0
  else
    term=(x/2)^3/2
    sign=-1
    sum=term*sign
    '300 iterations is enough to calculate any
    'value within the range of double precision
    for i = 1 to 300
      sign=-sign
      termi=(x/(4*i)) * (x/(4*i+2))^2 * (x/(4*i+4))
      term=term*termi
      sum=sum+sign*term
      'Skip out of loop if current term < 1e-12 of sum
      if abs(term/sum)<1e-12 then exit for
    next
    Ber_()= sum
  end if
end Function
```

```
Function Bei(ByVal x as double) as double
  Dim i,sign As integer
  Dim sum,termi,term As Double
  ' Calculates Bei(x) function
  ' Uses recurrence relation based on H.B. Dwight's
  ' series expansion formula 820.4
  ' Basic code by Robert Weaver 2009-10-26
  if x=0 then
    Bei()= 0
  else
    term=x*x/4
    sign=1
    sum=term*sign
    '300 iterations is enough to calculate any
    'value within the range of double precision
    for i = 1 to 300
      sign=-sign
      termi=((x/(4*i)) * (x/(4*i+2)))^2
      term=term*termi
      sum=sum+sign*term
      'Skip out of loop if current term < 1e-12 of sum
      if abs(term/sum)<1e-12 then exit for
    next
     Bei()= sum
  end if
end Function

Function Bei_(ByVal x as double) as double
  Dim i,sign As integer
  Dim sum,termi,term As Double
  ' Calculates Bei'(x) function -- d/dx Bei(x)
  ' Uses recurrence relation based on H.B. Dwight's
  ' series expansion formula 820.6
  ' Basic code by Robert Weaver 2009-10-26
  if x=0 then
    Bei_()= 0
  else
    term=x/2
    sign=1
    sum=term*sign
    '300 iterations is enough to calculate any
    'value within the range of double precision
    for i = 1 to 300
      sign=-sign
      termi=(x/(4*i-2)) * (x/(4*i))^2 * (x/(4*i+2))
      term=term*termi
      sum=sum+sign*term
      'Skip out of loop if current term < 1e-12 of sum e
      if abs(term/sum)<1e-12 then exit for
    next
     Bei_()= sum
  end if
end Function
```

Part II - Complete Elliptic Integrals of the First and Second Kind

Again, we refer to H. B. Dwight's *Tables of Integrals and Other Mathematical Data*. He gives series formulae for these functions as will be shown in the following sections.

It should be pointed out that depending on where these functions are encountered, and how they are implemented, the input argument may be in one of three different forms: k, k^2, or θ, where $\theta=\sin^{-1}k$. The input argument for the formulae presented here will be k, which is known as the *modulus* of the integral.

1. Complete Elliptic Integral of the First Kind

Dwight gives a series formula for this function as:

$$K(k) = \frac{\pi}{2}(1+m)\left[1 + \frac{1^2}{2^2}m^2 + \frac{1^2\cdot 3^2}{2^2\cdot 4^2}m^4 + \frac{1^2\cdot 3^2\cdot 5^2}{2^2\cdot 4^2\cdot 6^2}m^6 + \ldots\right] \tag{773.2}$$

where

$$m = (1-k')/(1+k')$$

$$k' = \sqrt{1-k^2}$$

This formula converges quickly for $k<0.91$, but more slowly for higher input values. Dwight gives a second formula which complements the first one, as it converges quickly for large k, and more slowly for small k:

$$K(k) = \log\frac{4}{k'} + \frac{1^2}{2^2}\left[\log\frac{4}{k'} - \frac{2}{1\cdot 2}\right]k'^2 + \frac{1^2\cdot 3^2}{2^2\cdot 4^2}\left[\log\frac{4}{k'} - \frac{2}{1\cdot 2} - \frac{2}{3\cdot 4}\right]k'^4$$
$$+\frac{1^2\cdot 3^2\cdot 5^2}{2^2\cdot 4^2\cdot 6^2}\left[\log\frac{4}{k'} - \frac{2}{1\cdot 2} - \frac{2}{3\cdot 4} - \frac{2}{5\cdot 6}\right]k'^6 + \ldots \tag{773.3}$$

The approach will be to use both formulae, selecting the one appropriate for the input argument.

Again we will number the terms starting at zero. The pattern is again readily apparent. Dealing with (773.2) first, it can be seen that *term n* is equal to *term n-1* multiplied by:

$$\left[\frac{(2n-1)m}{2n}\right]^2$$

All that remains to be done, is explicitly calculate the starting parameters, iterate the terms and then multiply the final sum by π(1+m)/2.

Formula (773.3) consists of a series of terms, each of which is comprised of a series of sub-terms. The coefficients outside of the brackets can be calculated from the preceding coefficients; *coefficient n* is equal to *coefficient n-1* multiplied by:

$$\left[\frac{(2n-1)k'}{2n}\right]^2$$

Inside the brackets, *subterm n* is equal to *subterm n-1* minus:

$$\frac{2}{(2n-1)2n}$$

One final point worth noting is that *n* never appears without a coefficient of 2 in front of it in either formula. Therefore, we can optimize things a bit further by starting the loop counter at two, and incrementing by two, then do away with the coefficient. The function converges in fewer than 13 iterations.

The Open Office Basic code for the complete elliptic integral of the first kind follows:

```
Function EllipticK (ByVal k As Double) As Double
  ' Calculate the complete elliptic integral of the
  ' first kind with modulus k
  ' Uses series expansion
  ' Based on Dwight's formulas 773.2 & 773.3
  ' Basic code by Robert Weaver, 2009-10-26
  dim n As Integer
  dim sum,term,termi,kp,kp2,m,m2,coeff As Double
  kp2=1-k*k
  kp=sqr(kp2)  'complementary modulus
  if k<=.91 then
    ' if k <= .91 use formula 773.2
    m=(1-kp)/(1+kp)
    m2=m*m
    term=1.0 'the zeroth term is 1.0
    sum=term
    for n=2 to 100 step 2
      'calc nth coefficient
      termi=((n-1)/(n))^2*m2
      term=term*termi
      sum=sum+term
      if (term/sum)<1e-12 then exit for
    next
    EllipticK() = pi()*sum/2*(m+1)
  else
    ' if k > .91 use formula 773.3
    term=log(4/kp)
    coeff=1.0
    sum=term
    for n=2 to 100 step 2
      coeff=coeff*((n-1)/n)^2*kp2
      termi=2/((n-1)*n)
      term=term-termi
      sum=sum+coeff*term
      if (coeff*term/sum)<1e-12 then exit for
    next
    EllipticK() = sum
  end if
End Function
```

2. Complete Elliptic Integral of the Second Kind

Dwight gives a series formula for this function as:

$$E(k) = \frac{\pi}{2(1+m)}\left[1 + \frac{m^2}{2^2} + \frac{1^2}{2^2 \cdot 4^2}m^4 + \frac{1^2 \cdot 3^2}{2^2 \cdot 4^2 \cdot 6^2}m^6 + \dots\right] \tag{774.2}$$

where

$$m = (1-k')/(1+k')$$

$$k' = \sqrt{1-k^2}$$

This formula converges quickly for $k<0.92$, but more slowly for higher input values. Dwight gives a second formula which complements the first one, as it converges quickly for large k, and more slowly for small k:

$$E(k) = 1 + \frac{1}{2}\left[\log\frac{4}{k'} - \frac{1}{1\cdot 2}\right]k'^2 + \frac{1^2 \cdot 3}{2^2 \cdot 4}\left[\log\frac{4}{k'} - \frac{2}{1\cdot 2} - \frac{1}{3\cdot 4}\right]k'^4$$
$$+\frac{1^2 \cdot 3^2 \cdot 5}{2^2 \cdot 4^2 \cdot 6}\left[\log\frac{4}{k'} - \frac{2}{1\cdot 2} - \frac{2}{3\cdot 4} - \frac{1}{5\cdot 6}\right]k'^6 + \dots \tag{774.3}$$

The approach, as before, will be to use both formulae, selecting the one appropriate for the input argument.

For formula (774.2), it can be seen that *term n* is equal to *term n-1* multiplied by:

$$\left[\frac{(2n-3)m}{2n}\right]^2$$

As a point of interest, for the case n=1, the part inside the brackets is negative, but squaring the result yields the proper positive value.

Next, we examine formula (774.3). It's interesting to compare it to formula (773.3). They are very similar, but have a few subtle differences which introduce complications to their implementation. In the coefficient outside of the brackets, the rightmost factors in the numerator and denominator are not squared, and the last term inside the brackets has a numerator of one rather than two.

The coefficients outside of the brackets can be calculated from the preceding coefficients; *coefficient n* is equal to *coefficient n-1* multiplied by:

$$\frac{(2n-1)}{2n} \times \frac{(2n-3)}{2n-2}k'^2$$

Note that this fails in the case n=1 due to a zero denominator. There are a couple of ways to resolve the problem. We can explicitly calculate both coefficients 0 and 1, as well as terms 0 and 1, and then begin the loop at n=2. Alternatively, we can save the incremental part of the coefficient from one iteration, to be used in the following iteration, giving it an initial value of one. This second approach will be used as it will result in fewer mathematical operations. In this case we define *ci* as the incremental part of the coefficient:

$$ci_0 = 1$$

$$ci_n = \frac{2n-1}{2n} \quad \text{for } n>0$$

and then

$$coefficient_n = (ci_{n-1} \times ci_n)k'^2$$

Next, we look at what's inside the brackets. Inside, *term n* is equal to *term n-1* minus:

$$\frac{1}{(2n-1)2n} + \frac{1}{(2n-3)(2n-2)}$$

We see that this also fails for the case n=1 due to a zero denominator. It will be resolved in the same way as before. The incremental part of the *term* is defined as *ti*:

$$ti_0 = 0$$

$$ti_n = \frac{1}{(2n-1)2n} \quad \text{for } n>0$$

and then

$$term_n = term_{n-1} - ti_{n-1} - ti_n$$

As before, the variable n never appears without a coefficient of 2 in front of it. Therefore we initialize the loop at two, and increment by two, and remove the coefficient. The function converges in fewer than 13 iterations.

The Open Office Basic code for the complete elliptic integral of the second kind follows:

```
Function EllipticE (ByVal k As Double) As Double
  ' Calculate the complete elliptic integral of the
  ' second kind with modulus k
  ' Uses series expansion
  ' Based on H. B. Dwight's formulae 774.2 & 774.3
  ' Basic code by Robert Weaver, 2009-10-26
  dim n As Integer
  dim sum,term,termi,kp,kp2,m,m2,coeff,cio,cin,tio,tin As Double
  kp2=1-k*k
  kp=sqr(kp2) 'complementary modulus
  m=(1-kp)/(1+kp)
  if k=1 then
    'This prevents a divide by zero problem
    EllipticE() = 1
  elseif k<.93 then
    'formula 774.2
    term=1.0 'the zeroth term is 1.0
    sum=term
    coeff=1.0 'the zeroth coefficient is 1
    for n=2 to 100 step 2
      termi=m*(n-3)/n
      term=term*termi*termi
      sum=sum+term
      if (term/sum)<1e-12 then exit for
    next
    EllipticE() = pi()*sum/(2*m+2)
  else
    'formula 774.3
    tio=0
    cio=1
    coeff=1
    term=log(4/kp)
    sum=1
    for n=2 to 100 step 2
      cin=(n-1)/n
      coeff=coeff*cio*cin*kp2
      cio=cin
      tin=1/((n-1)*n)
      term=term-tio-tin
      tio=tin
      sum=sum+term*coeff
      if (term*coeff/sum)<1e-12 then exit for
    next
    EllipticE() = sum
  end if
End Function
```