

## Bessel K – Non-Integer, $\nu$

Andrew Allsop – AWEC, 4<sup>th</sup> March 2026

### 1.0 Introduction

Modified Bessel functions of the second-kind (Bessel-K) are needed for calculation of coherence functions for wind turbulence with non-integer  $\nu$  (e.g. 1/6, 5/6 and 11/6). But the available package in EXCEL for Bessel-K is only for integers. Although Bessel non-integer functions are available in Python, FORTRAN and C++ libraries, an EXCEL-VBA function was needed by the author.

The method below is based on an integral solution suggested in Wikipedia and includes practical methods to help convergence.

The method has been checked against EXCEL integer values '=BESSELK(x, n<sup>(1)</sup>)' and spot-checked against web-based functions with non-integer values.

### 2.0 Integral Formula for $K_\nu(x)$

The integral formula used is the following:

$$K_\nu(x) = \int_0^\infty e^{-x \cdot \cosh t} \cdot \cosh \nu t \cdot dt$$

Solutions by summation in this case can be relatively slow compared to the summation formulae that are used for calculation of integer  $K_n(x)$ , and the necessary number of steps, spacing, and practical summation limits can only to be determined empirically and vary with the parameters set.

### 3.0 Practical Summations

The VBA solution provided is iterative and converges on the correct value according to limits and steps that are adjusted within the calculation.

The formulation was derived to be robust against numerical overflow errors.

---

<sup>1</sup> Note that non-integer values of 'n' are rounded down in EXCEL, without generating any error or warning!

## 4.0 EXCEL VBA Code

### CODE

Option Explicit

Function AWECBessK(ByVal X As Double, ByVal Nu As Double) As Double

' Copyright Arup/AWEC – May be used only with user verification.

' Bessel K calculation for non-integer Nu.

' Summation solution to the Wikipedia integral expression

' ~4s per million calls, Windows 11, Excel, Mar. 2026, Laptop

Dim Sum1 As Double, Sum2 As Double, Mult As Double, LnMult As Double, \_  
Val As Double, vMax As Double, Y As Double, Ymax As Double, \_  
t1 As Double, T2 As Double, Ln2 as Double

Ln2 = Log(2#)

Mult = 1.5 ' Optimum start value

LnMult = Log(Mult)

vMax = 0#

Sum1 = 0#

loopback:

Sum2 = 0#

Y = Sqr(Mult)

Do

t1 = Log(Y) \* Nu

T2 = Ln2 + X / 2 \* (Y + 1 / Y)

Val = Exp(-t1 - T2) + Exp(t1 - T2)

' Val = Exp(-x.Cosh(t).Cosh(Nu.t), and Y = Exp(t)

' Formulation avoids numerical overflow with intermediate numbers

' Integration is from Y = ~1 to ~Infinity in equal ratio steps.

If vMax < Val Then

vMax = Val

Ymax = Y

Elseif Val = 0# Then 'Added to address an underflow error

AWECBessK = 0#

Exit Function

End If

Sum2 = Sum2 + Val \* LnMult

Y = Y \* Mult

```
Loop Until Y > 1.5 * Ymax And Val < vMax / 10000000
```

```
If Abs(Sum1 / Sum2 - 1) < 0.0000001 Then ' Normal exit criterion is met
```

```
  AWEKBessK = Sum2
```

```
Else
```

```
  Sum1 = Sum2
```

```
  Mult = Sqr(Mult)
```

```
  LnMult = LnMult * 0.5
```

```
  GoTo loopback
```

```
End If
```

```
End Function
```

```
END CODE
```

## 5.0 Usage

The code above may be used only with user verification that it is working correctly in the users own installation.

The accuracy ratio is currently set to at least  $10^{-6}$ , which is more than accurate enough for wind engineering coherence calculations. Some speed gain is achievable by reducing the accuracy, but when numbers of needed Bessel calculations may exceed millions in conjunction with other calculations, e.g. when carrying out 2D wind coherence summations, it is recommended to pre-calculate results in an integer lookup array/table of perhaps a few thousand values depending on the precision needed.

Andrew Allsop 20260304

PS It would be possible to speed the calculation by a factor of about 2 by reusing the results of the previous calculation iteration and just filling the gaps, but it was fast enough!