bessY

bessY(*n*, *x* [, *algorithm* [, *accuracy*]])

Obsolete — use **Bessely**.

The bessY function returns the Bessel function of the second kind, Y*n*(*x*) of order *n* and argument *x*.

For real *x*, the optional parameter *algorithm* selects between a faster, less accurate calculation method and slower, more accurate methods. In addition, when *algorithm* is zero or absent, the order *n* is truncated to an integer.

When *algorithm* is included and is 1, *accuracy* can be used to specify the desired fractional accuracy. See Details about algorithms.

If *x* is complex, a complex result is returned. In this case, *algorithm* and *accuracy* are ignored. The order *n* can be fractional, and must be real.

Details

See the **bessI** function for details on algorithms, accuracy and speed of execution.

When *algorithm* is 1, pairs of values for bessJ and bessY are calculated simultaneously. The values are stored, and a subsequent call to bessY after a call to bessJ (or vice versa) with the same *n*, *x*, and *accuracy* will be very fast.

beta

beta(*a*, *b*)

The beta function returns for real or complex arguments as

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)},$$

with *Re*(*a*), *Re*(*b*)>0. Γ is the gamma function.

See Also

The **gamma** function.

betai

betai(*a*, *b*, *x* [, *accuracy*])

The betai function returns the regularized incomplete beta function I*x*(*a*,*b*),

$$I_x(a,b) = \frac{B(x;a,b)}{B(a,b)}.$$

Here

$$B(x;a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\,dt.$$

where $a,b > 0$, and $0 \le x \le 1$.

Optionally, *accuracy* can be used to specify the desired fractional accuracy.

Details

The *accuracy* parameter specifies the fractional accuracy that you desire. That is, if you set *accuracy* to 10^{-7}, that means that you wish that the absolute value of $(f_{actual} - f_{returned})/f_{actual}$ be less than 10^{-7}.

Larger values of *accuracy* (poorer accuracy) result in evaluation of fewer terms of a series, which means the function executes somewhat faster.

A single-precision level of accuracy is about 3×10^{-7}, double-precision is about 2×10^{-16}. The betai function will return full double-precision accuracy for small values of *a* and *b*. Achievable accuracy declines as *a* and *b* increase: