Subject: Re: IDL's BESELJ returns NAN for small argument and large order
Posted by David Ruffner on Wed, 19 Mar 2014 20:48:42 GMT
View Forum Message <> Reply to Message

On Tuesday, March 18, 2014 7:47:28 PM UTC-4, Phillip Bitzer wrote:
> "That's not a bug, it's a feature" :-)
>
>
>
> The reason you get a NaN can be found in the help:
>
>
>
> ITER
>
> Set this keyword equal to a named variable that will contain the number of iterations performed. If the routine converged, the stored value will be equal to the order N. If X or N are arrays, ITER will contain a scalar representing the maximum number of iterations.
>
> Note: If the routine did not converge for an element of X, the corresponding element of the Result array will be set to the IEEE floating-point value NaN, and ITER will contain the largest order that would have converged for that X value.
>
>
>
> So,
>
> IDL> print,beselj(0.1d,103.d, ITER=n)
>
> IDL> print, n ;get n=102
>
>
>
> So, the algorithm properly converges for order 102, but not 103+. This is why get a number for your second example.

Hi Phillip,
Thanks for your reply. I see what you are saying. It's better that beselj returns a NAN than silently returning something that could be wrong.

I checked the series expansion on Mathworld and found that for integer order and small argument J_n(x) should go to zero. I'm not sure about fractional order which may be why beselj returns the NAN. In my code I only need to use integer order so I just catch the NANs and set beselj to zero whenever I am sure it should be zero.

Thanks,
David