

The Bessel function

Mark L. Winther

March 16, 2018

Introduction

The purpose of this project is to make a program that is able to solve the Bessel function numerically for the first kind of integer index, via the integral representation. The underlying purpose is for the author of this project to learn more about coding and solving mathematical problems via numerical methods.

Theory

Based on the material from [1].

Bessel functions are canonical solutions to Bessel's differential equation

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - \alpha^2) y = 0, \quad (1)$$

where α is an arbitrary complex number. The most important and applicable solutions are however when α is a real integer. A specific instance of these solutions are defined as Bessel functions of the first kind $J_\alpha(x)$, which are finite at $x = 0$ for integer or positive α but diverge as x approaches zero for negative non-integer α . Through series expansion it can be defined as

$$J_\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left(\frac{x}{2}\right)^{2m+\alpha}, \quad (2)$$

where $\Gamma(k)$ is the gamma function. Limiting it further to only integer values of α , defined as n , the Bessel function can be represented by the integral

$$J_n(x) = \frac{1}{\pi} \int_0^\pi \cos(n\tau - x \sin \tau) d\tau, \quad (3)$$

which is the representation of the Bessel function that will be solved numerically in this project.

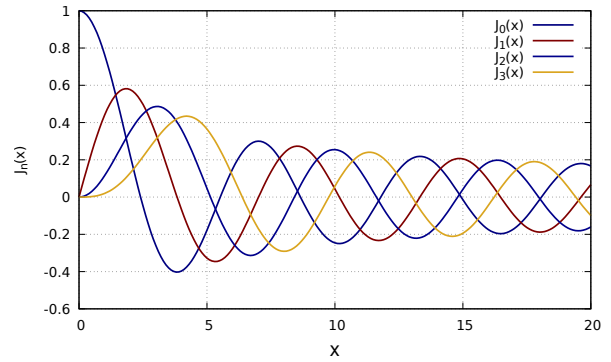


Figure 1: The numerically solved Bessel function, for a range of x and n .

Solution

Firstly, the solution to the problem itself is contained in the program from `bessel.c`. It is a callable function, that can be called by: `./bessel n x`, that prints x , n , $J_n(x)$, J_g , d to the standard output, where $J_n(x)$ is the calculated value, J_g is the corresponding value from using the Bessel function from GSL¹, and d is the difference between these. The rest of the programs are for extracting plotable data, but use the same way of solving the function. The plot of the functions solved numerically can be seen in Figure 1, and a plot of its deviation from the GSL function in Figure 2.

Generally, the program solves the Bessel function by numerical integration, via the use of GSL's numerical integration routine, defined in `gsl.integration.h`. When the parameters n and

¹<https://www.gnu.org/software/gsl/>

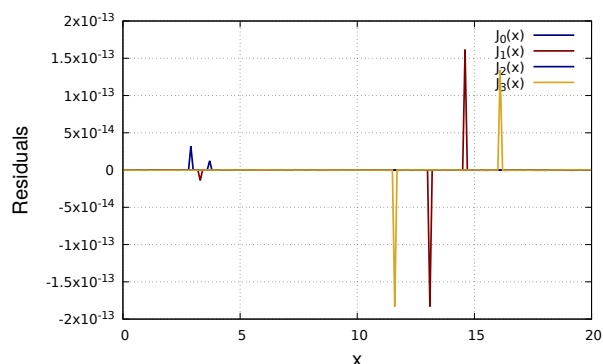


Figure 2: The residuals of the numerically solved Bessel function, for a range of x and n .

x are given to the program, it sets up the appropriate environment for integration, and solves it via the QAGS algorithm to solve the integral, with both relative and absolute uncertainties set to 1×10^{-7} , whereafter the program prints out the result.

The QAGS algorithm solves the the integral by combining adaptive bisection with the Wynn epsilon-algorithm. This allows it to solve integrals with singularities in the integration regoin, by concentrating new subintervals around the singularity.

Conclusion

Although the program is simple, and does not do advanced checks of the solutions, it can be seen in Figure 2 that it is quite close to the actual solutions, and shows no systematical errors. Therefore, it is a good approximation to the exact solution.

References

- [1] Wikipedia. *Bessel function* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Bessel%20function&oldid=829598912>. [Online; accessed 14-March-2018]. 2018.