




# SpecialFunctions: A C# package of special functions for scientific computing with MATLAB-compatible API

Jake W. Liu <sup>1</sup>

<sup>1</sup> Department of Electronic Engineering, National Taipei University of Technology, Taiwan

DOI: [10.21105/joss.08516](https://doi.org/10.21105/joss.08516)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Owen Lockwood](#)  

## Reviewers:

- [@JeWaVe](#)
- [@jasonyaaa](#)

Submitted: 12 June 2025

Published: 09 September 2025

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

SpecialFunctions is a C# library designed for scientific computing applications requiring a broad range of special functions, such as those found in mathematical physics and engineering ([Abramowitz & Stegun, 1965](#); [Arfken et al., 2011](#)). While Math.NET Numerics is the dominant open-source library for numerical computing in C#, its support for special functions remains incomplete. Furthermore, alternatives like ALGLIB are restricted by GPL licensing, making them unsuitable for commercial use.

This package addresses these gaps by offering a more complete set of special functions under the permissive MIT license, facilitating both academic and commercial adoption. The implementation leverages Math.NET Numerics where applicable ([Math.NET Numerics, n.d.](#)) and incorporates extensions such as Legendre functions based on open-source contributions (e.g., AminKH's repository) ([Legendre-Polynomials, n.d.](#)). The API design follows MATLAB conventions to support easier transition for users familiar with MATLAB, and includes functions like Bessel functions (including spherical variants), Airy functions, incomplete gamma functions, error functions, Legendre polynomials, and logarithmic derivatives.

The library is developed to support electromagnetic simulations and field transformations—specifically, cylindrical and spherical near-to-far-field (NTFF) calculations—in environments where C# is frequently used for hardware control. Full documentation and MATLAB-style references are provided for user guidance.

## Statement of need

The C# ecosystem lacks a comprehensive, permissively licensed library that offers a wide range of special functions required in scientific and engineering computations. Existing libraries such as Math.NET Numerics provide partial coverage but omit important function families such as associated Legendre polynomials. Other alternatives like ALGLIB include more functionality but are distributed under the GPL license, restricting their use in commercial applications.

This project addresses that gap by providing an open-source, MIT-licensed library that combines ease of use with broad coverage of special functions. It is designed to support both academic research and industrial development, particularly in domains where C# is widely adopted, such as real-time control systems and electromagnetic simulation software. The implementation adheres to MATLAB-style API conventions, facilitating a low-friction transition for users familiar with numerical computing in MATLAB.

By offering consistent naming conventions, support for complex-valued inputs, and compatibility with Math.NET Numerics, this library serves as a practical and extensible foundation for

scientific applications—including, but not limited to, near-to-far-field (NTFF) transformations, wave propagation modeling, and geodetic calculations.

## Overview of features

The `SpecialFunctions` library provides a comprehensive set of special functions commonly used in scientific computing. The implementation is structured to be extensible, allowing for future inclusion of additional functions. Key features are grouped into the following families:

- **Bessel Functions:** A comprehensive suite of Bessel functions, including the first (`besselj`) and second (`bessely`) kinds, modified variants (`besseli`, `besselk`), and Hankel functions (`besselh`). Spherical Bessel functions (`sbesselj`, `sbessely`, `sbesselh`) are also supported.
- **Airy Functions:** Airy functions (`airy`) and their derivatives for real and complex arguments.
- **Gamma and Beta Functions:** The gamma function (`gamma`), its logarithm (`gammaLn`), the incomplete gamma function (`gammaInc`), and the digamma function (`psi`). The beta function (`beta`) and its logarithm (`betaLn`) are also implemented.
- **Error Functions:** The error function (`erf`), its complement (`erfc`), and their respective inverses (`erfinv`, `erfcinv`).
- **Legendre Polynomials:** Legendre polynomials (`legendre0`) and associated Legendre functions (`legendre`) with multiple normalization options.

Each function supports real and complex input variants where applicable. The library is implemented as a static class (SF) under the `SpecialFunctions` namespace.

## Acknowledgements

Legendre function implementations reference AminKH's repository. The library also relies on selected components from Math.NET Numerics.

## References

- Abramowitz, M., & Stegun, I. A. (1965). *Handbook of mathematical functions: With formulas, graphs, and mathematical tables* (Vol. 55). Courier Corporation.
- Arfken, G. B., Weber, H. J., & Harris, F. E. (2011). *Mathematical methods for physicists: A comprehensive guide*. Academic press.
- Legendre-polynomials*. (n.d.). <https://github.com/AminKH/Legendre-Polynomials>
- Math.NET numerics* (Version 5.0.0). (n.d.). <https://github.com/mathnet/mathnet-numerics>