SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 21, No. 3, October 2024, 345-357

UDC: 004.41:517.58 DOI: https://doi.org/10.2298/SJEE2403345L

Calculator of Some Special Mathematical Functions

Dedicated to the bright memory of prof. Milić Đekić

Sara Lazić[1], Bratislav Iričanin[1, 2]

Abstract: This paper presents the implementation of a calculator of certain special mathematical functions in the form of an efficient web application with a simple and intuitive GUI (graphical user interface). This application aims to enable accurate numerical approximations of the most frequently used special mathematical functions in engineering and science, eliminating the need to acquire additional notational knowledge or programming language syntax experience. The investigation can be divided into three larger units. The initial section offers brief overview of the special mathematical functions of which numerical approximations are implemented in the subject application. For the time being, this application provides approximations for the following special mathematical functions: Bessel functions of the first kind, gamma and beta functions, and some orthogonal polynomials – Legendre, Laguerre, Hermite (physicist's and probabilist's) polynomials, Chebyshev polynomials of the first and the second kind, as well as Jacobi polynomials. The middle section presents a description of the computer implementation – an overview of the used technology and implemented algorithms, while the final section includes a discussion of the solution, as well as a comparison with existing software that provides the same features as the subject application.

Keywords: Special functions, Orthogonal polynomials, Gamma function, Numerical approximations, Numerical calculator.

1 Introduction to Special Mathematical Functions

Special functions are a unique class of mathematical functions frequently used to solve specific problems in mathematics, physics, and technical sciences. Due to their importance and frequent use, these functions have been assigned specific names and common notations. What is interesting about these functions

[1]School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, Belgrade 11120, Serbia; E-mail: iricanin@etf.rs
[2]Faculty of Mechanical and Civil Engineering in Kraljevo, University of Kragujevac, 36000 Kraljevo, Serbia; E-mail: sara.lazic@outlook.com
An abridged version of this work was successfully verbally presented at the 15th SMAK (Serbian Mathematical Congress) in June 2024, held in Belgrade, Serbia.

is that for most of them, the exact values are known only for some values of the argument, while for all other points, their values are calculated numerically or by using other approximations. Almost all special functions can be written in one of two ways (which can be proven), and based on that, they can be classified into two groups:

– functions in the form of integrals,

– functions in the form of infinite convergent (usually power) series.

Functions in the form of integrals include beta and gamma functions, which, among other applications, are important in analysing other special functions. The beta and gamma functions are also known as Euler's integral of the first and second kind, respectively.

Given their widespread application in engineering, there is a need for software capable of efficiently approximating the values of these functions – for example, an intuitive web application that will provide the approximation in a minimal number of steps, generate a graphical representation of the function, and provide basic function information. This is the core idea behind the web application at hand.

2 Overview of Implemented Functions

This part of the paper gives a brief overview of the special mathematical functions of which approximations are implemented in the subject application.

2.1 The Bessel function of the first kind

Let n be an integer non-negative constant. Then the function

$$J_n(x) = \sum_{m=0}^{\infty} (-1)^m \frac{\left(\frac{x}{2}\right)^{2m+n}}{(m+n)!\cdot m!} \tag{1}$$

is called a Bessel function of the first kind, order α, with argument *x*, $x \in \mathbb{R}$.

The Bessel functions of the first kind are defined as the solutions to the Bessel differential equation, which is a homogeneous linear differential equation of the second order [6]:

$$x^2 y'' + xy' + (x^2 - \alpha^2) y = 0 . \tag{2}$$

Generating function of Bessel function of the first kind is:

$$g(x,t) = \sum_{n=-\infty}^{\infty} J_n(x) t^n = \mathrm{e}^{\frac{x}{2}\left(t - \frac{1}{t}\right)} . \tag{3}$$

Bessel function of the first kind, as well as most special functions, can be given using the contour integral as [10]:

$$J_n(z) = \frac{1}{2\pi i} \oint e^{\frac{z}{2}\left(t-\frac{1}{t}\right)} t^{-n-1}\,\mathrm{d}t. \tag{4}$$

2.2 The Gamma function

The gamma function is one of the most commonly used special functions. It is used in number theory, probability, integral calculus, and especially in solving problems in physics. It frequently appears in the theory of other special functions, giving it particular significance. It stands out from the other functions discussed in this paper due to the simplicity of its recurrent relations, a crucial feature that arguably defines its important role in the theory of special functions.

The gamma function, on the domain of positive real numbers, is commonly defined by the Euler integral:

$$\Gamma(x) = \int_0^{\infty} \mathrm{e}^{-t}\, t^{x-1}\,\mathrm{d}t. \tag{5}$$

The gamma function is related to the factorial function for argument' integer values via the formula [3]:

$$\Gamma(n) = (n-1)!\,. \tag{6}$$

2.3 The Beta function

The beta function is also known as the Euler integral of the first kind. It finds its application in various fields, most often in mathematical analysis, probability theory, statistics, quantum physics and, of course, engineering.

Beta function is two variable one, and defined using a definite integral:

$$B(p+1, q+1) = \int_0^1 t^p (1-t)^q\,\mathrm{d}t, \quad p,q \in \mathbb{R},\ p,q > 0. \tag{7}$$

This function can be expressed using the gamma function as (8) [10]:

$$B(p+1, q+1) = \frac{\Gamma(p)\cdot\Gamma(q)}{\Gamma(p+q)}\,. \tag{8}$$

2.4 Orthogonal polynomials

Several types of classical orthogonal polynomials exist, some of which were already mentioned in the introduction, including Laguerre, Hermite, and Jacobi polynomials, as well as Legendre and Chebyshev polynomials (both of the first and second kinds). All the aforementioned are classified as classical orthogonal polynomials, which were derived from solving linear differential equations of the second order (of the Fuchs' type) [4].

Legendre polynomials are orthogonal polynomials with a weight function $w(x) = 1$ over the interval $I = (-1,1)$, series expansion of which is [8]:

$$P_n(x)=\frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k}^2(x-1)^{n-k}(x+1)^k. \qquad (9)$$

Legendre differential equation is a linear differential equation of second order [2]:

$$\left(1-x^2\right)y''-2xy'+k(k+1)y=0. \qquad (10)$$

Laguerre polynomials are orthogonal polynomials with a weight function $w(x)=\mathrm{e}^{-x}$, defined over the domain of real numbers, expressed by Rodrigues's formula as:

$$L_n(x)=\frac{\mathrm{e}^x}{n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^n\,\mathrm{e}^{-x}\right)=\frac{1}{n!}\left(\frac{\mathrm{d}}{\mathrm{d}x}-1\right)^n x^n, \qquad (11)$$

and they are solutions to the Laguerre differential equation, which is also a linear differential equation of second order [1]:

$$xy''+(1-x)y'+\lambda y=0. \qquad (12)$$

We distinguish between two types of Hermite polynomials – physicist's and probabilist's. Hermite physicist's polynomial is the polynomial:

$$H_n(x)=n!\sum_{k=0}^{\left[\frac{n}{2}\right]}(-1)^k\frac{(2x)^{n-2k}}{k!(n-2k)!}, \qquad (13)$$

orthogonal with respect to the weight function $w(x)=\mathrm{e}^{-x^2}$ over the interval $(-\infty,\infty)$. Hermite probabilist's polynomial are given by:

$$H_{e_n}(x)=2^{-\frac{n}{2}}H_n\left(\frac{x}{\sqrt{2}}\right). \qquad (14)$$

Hermite physicist's and probabilist's polynomials are solutions to the next two second order linear equations, respectively [10]:

$$\begin{aligned} y''-xy'+ny&=0,\\ y''-2xy'+ny&=0. \end{aligned} \qquad (15)$$

Chebyshev polynomials are sequences of orthogonal polynomials defined by sine and cosine functions[3]. We differentiate between Chebyshev polynomials of the first kind and Chebyshev polynomials of the second kind. Chebyshev

[3]In general, the sine and cosine functions are also considered special functions derived from this equation. However, due to their extensive and frequent use in mathematics, they are classified as elementary functions.

polynomials of the first kind, orthogonal with respect to the weight function $w(x) = (1 - x^2)^{-1/2}$ over the interval $I = [-1, 1]$, are given by the expression:

$$T_n(x) = \cos(n \arccos x), \quad |x| \le 1,\ n \in \mathbb{N}_0. \tag{16}$$

Chebyshev polynomials of the second kind, with weight function $w(x) = (1 - x^2)^{1/2}$ over the same interval, are given as:

$$U_n(x) = \frac{\sin\left((n+1)x\right)}{\sin x}. \tag{17}$$

The Chebyshev differential equations, of which solutions are Chebyshev polynomials of the first and second kind, are respectively [10]:

$$\begin{aligned} &(1 - x^2)y'' - xy' + n^2 y = 0, \\ &(1 - x^2)y'' - 3xy' + n(n+2)y = 0. \end{aligned} \tag{18}$$

Finally, Jacobi polynomials, with real parameters $\alpha > -1$, $\beta > -1$, are sequences of polynomials which have the following form:

$$P_n^{(\alpha,\beta)}(x) = \frac{(-1)^n}{2^n n!}(1-x)^{-\alpha}(1+x)^{-\beta}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left[(1-x)^{\alpha+n}(1+x)^{\beta+n}\right]. \tag{19}$$

Their weight functions are

$$w(x) = (1-x)^{\alpha}(1+x)^{\beta}. \tag{20}$$

The domain of orthogonality of this class of polynomials is a finite interval, for which, without loss of generality, one can take a basic interval $(-1, 1)$ [10].

3 Idea and Motivation

Special functions find their application in solving numerous problems in mathematics, physics and technical sciences. Therefore, there is a need for an efficient yet straighforward way to calculate the values of these functions. This need leads to the concept of a modern, intuitive application that delivers strong performance and enables users to approximate the value of a function with minimal steps. This concept has been previously explored. For example, articles like [5] analyse similar software used until the end of the twentieth century, discussing their characteristics, problems, and prospects for further development.

Several popular applications that offer a vast range of calculations also include special functions, but they often present them in a slightly different format than intended for this application – typically involving more steps, a less intuitive interface, etc. Additionally, existing solutions often demand extra effort to use, such as learning notation, syntax, or even knowledge of basic programming concepts.

4 Implementation

The application is implemented using Typescript programming language and the *Angular* framework.

The libraries used are: *Lodash.js*, *Bignumber.js*, *Math.js*, *Plotly.js*.

Key functionalities:

- Selection of one of the supported special math functions,
- Input of input values in the form of more complex mathematical expressions, using the keyboard or graphical interface,
- Approximation of the value of the special function for given values of the argument and order of the function,
- Graphic display of the selected function,
- Providing basic information about the selected special mathematical function.

4.1 Input

The user will see a menu displaying the supported special mathematical functions when they visit the app. Selecting an item from the menu will redirect the user to a page with an input form. The number of input fields in this form depends on the chosen function – for example, the gamma function needs only one input – the variable, but the Jacoby polynomial takes four – parameters, order and variable. In these input fields the user can enter a number or a mathematical expression.

To handle the input of mathematical expressions, a parser has been implemented. The parsed expression is then converted from standard mathematical notation – infix notation, to postfix notation. This conversion is performed because evaluating an expression is significantly easier when it is in postfix notation. During the conversion process, the correctness of the expression is also verified. If the expression is incorrect, the user will be shown an appropriate message. The evaluation is peformed using a well-known stack–based algorithm, along with an algorithm for converting expressions from infix to postfix notation [11, 7].

All input fields are validated using regular expressions. A domain is defined for every input field for every supported function, and the value is subjected to validation. Of course, if any values for a given function are out of range, the user will be notified with an appropriate message, and he will be unable to approximate the value of the special function until all the values are within their ranges. The described input form can be seen in the Fig. 1.

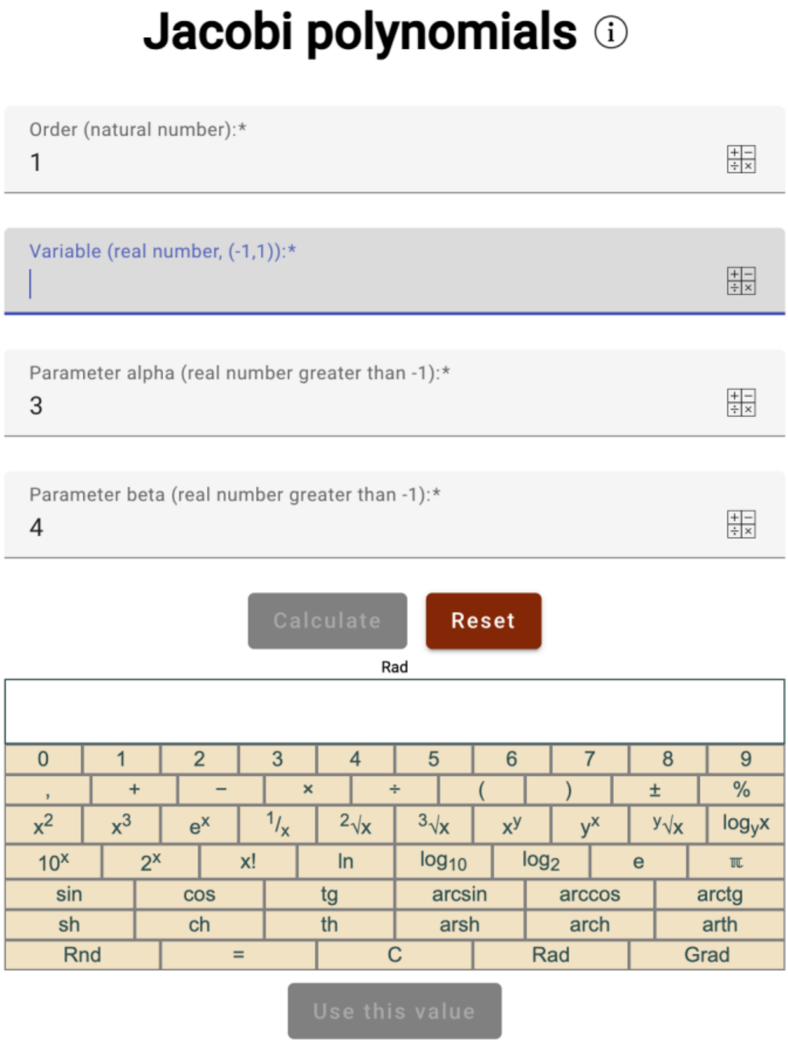

Fig. 1 – *Input form for Jacobi polynomials.*

4.2 Approximation

The values of some functions are approximated by some other implemented function – the beta function and the Bessel function of the first kind use the gamma function, while the Hermite probabilist's polynomial is calculated using Hermite physicist's polynomial. Most of the implemented functions are approximated using their formula in the form of convergent series.

Interestingly, the gamma function required the most effort to implement. The main reason for this is that the gamma function is used in approximating other functions. If the approximation of gamma function lacks sufficient precision, it also affects the precision of these other functions. There are many formulas for approximation of gamma function, some of the most popular being Stirling and Ramanujan, and there are many articles on these methods and proposed improvements to these formulas. Over ten different methods were implemented during our investigation, and results were compared,but only one produced acceptable results – the method developed by Cornelius Lanczos.

The Lanczos method implementation, given by Paul Godfrey, approximates the value of the natural logarithm of the gamma function using the (21), [4]:

$$\ln \boldsymbol{\Gamma}(\boldsymbol{x}+1) = \ln(\boldsymbol{Z}\cdot\boldsymbol{P}) + (\boldsymbol{x}+0.5)\cdot\ln(\boldsymbol{x}+\boldsymbol{g}+0.5) - (\boldsymbol{x}+\boldsymbol{g}+0.5)\,, \qquad (21)$$

where $\boldsymbol{Z}$ is vector

$$\boldsymbol{Z} = \left(1, \frac{1}{\boldsymbol{x}+1}, \frac{1}{\boldsymbol{x}+2}, \ldots, \frac{1}{\boldsymbol{x}+\boldsymbol{n}-1}\right), \qquad (22)$$

$\boldsymbol{n}$ and $\boldsymbol{g}$ are parameters which are arbitrarily chosen, and $\boldsymbol{P}$ is a vector that is defined as (23)

$$\boldsymbol{P} = \boldsymbol{D}\cdot\boldsymbol{B}\cdot\boldsymbol{C}\cdot\boldsymbol{F}\ , \qquad (23)$$

where $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{F}$ are as follows:

$$\boldsymbol{B}_{\boldsymbol{ij}} = \begin{cases} 1, & i = 0; \\ (-1)^{\boldsymbol{i}-\boldsymbol{j}} \binom{i+j-1}{j-i}, & i > 0, j \geq i; \\ 0, & \text{otherwise}; \end{cases} \qquad (24)$$

$$\boldsymbol{C}_{\boldsymbol{ij}} = \begin{cases} \frac{1}{2}, & i = j = 0; \\ 0, & j > i; \\ (-1)^{\boldsymbol{i}-\boldsymbol{j}} \sum_{\boldsymbol{k}=0}^{\boldsymbol{i}} \frac{(2\boldsymbol{x})^{\boldsymbol{n}-2\boldsymbol{k}}}{\boldsymbol{k}!\,(\boldsymbol{n}-2\boldsymbol{k})!}, & \text{otherwise}; \end{cases} \qquad (25)$$

$$\boldsymbol{D}_{\boldsymbol{ij}} = \begin{cases} 0, & i \neq j; \\ 1, & i = j = 0; \\ -1 & i = j = 1; \\ \frac{2(2\boldsymbol{i}-1)\cdot\boldsymbol{D}_{\boldsymbol{i}-1,\boldsymbol{j}-1}}{\boldsymbol{i}-1}, & \text{otherwise}; \end{cases} \qquad (26)$$

$$\boldsymbol{F}_{\boldsymbol{i}} = \frac{(2\boldsymbol{i})!\cdot\boldsymbol{e}^{\boldsymbol{i}+\boldsymbol{g}+0.5}}{\boldsymbol{i}!\cdot 2^{2\boldsymbol{i}-1}\cdot(\boldsymbol{i}+\boldsymbol{g}+0.5)^{\boldsymbol{i}+0.5}}\,. \qquad (27)$$

Since all the matrices depend only on the parameters $\boldsymbol{n}$ and $\boldsymbol{g}$, if we choose these parameters at the beginning and use them as constants, the matrices can be calculated at the very start of the program, during initialisation, and thus, we can reduce the running time of the algorithm itself because there is no need to calculate them every time. Therefore, matrix $\boldsymbol{P}$, which is the product of these matrices, is fixed and can also be calculated in the initialisation.

The choice of parameters $\boldsymbol{n}$ and $\boldsymbol{g}$ is crucial for the accuracy of the algorithm, and they were carefully chosen experimentally. The program is run for different combinations of values for this pair of parameters, and the combination that produces the best results is selected. This pair achieved a result with up to 50 correct decimal places behind the decimal point, which is significantly better than the 15 decimal places obtained with other approximation methods.

4.3 Graphical representation

To display a graph of a function, the program calculates the values of the selected special function for two hundred variables in a certain interval and plots them on a graph using the *Plotly.js* library. The interval that we use for this depends on the function selected by the user. For special functions which are implemented on a finite domain, the whole interval is shown – those are Legendre polynomials, Chebyshev (of the first and second kind) and Jacobi, and the interval is (–1, 1). Other polynomials are shown on the interval that is chosen based on the variable entered by the user – we take values around this variable. For the entered value of the variable, the point is drawn in different color then the rest of the graph, in order to clearly see where that value is located on the function graphic. Fig. 2 shows the example of a graphical representation of Jacobi polynomial – order is 4, parameters are $\boldsymbol{\alpha} = 2, \boldsymbol{\beta} = 3$ and $\boldsymbol{x} = 0$.

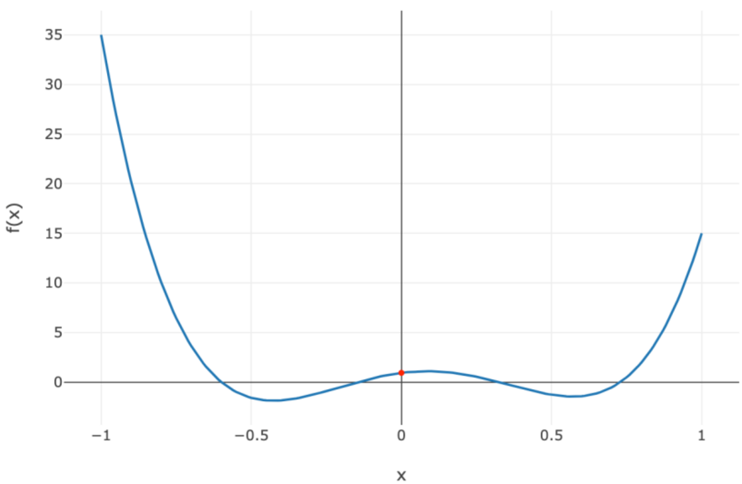

Fig. 2 – *The example of a graphical representation of a special function.*

4.4 Existing software with a similar set of functionalities

Upon examining existing calculators for special functions, we find several web applications that differ from the application discussed in this work across various criteria.

One such tool is *WolframAlpha* (`https://www.wolframalpha.com/`). This software has been around for many years and has expanded well beyond mathematics to cover a wide range of areas. Today, it includes sections on: Science and Technology, Society and Culture, Everyday Life. In this context, comparing this application to WolframAlpha may not be entirely appropriate, as WolframAlpha is a vast web application with numerous functionalities. In contrast, this work focuses solely on the calculation of special functions, which is just a small part of the WolframAlpha software suite. In the context of special functions, such as Bessel functions, users can access the calculator from the home page via a simple keyword search. The application then directs them to a page where they can input the variable, specify the order of the function, and select the type of Bessel function from a drop-down menu, Fig 3.

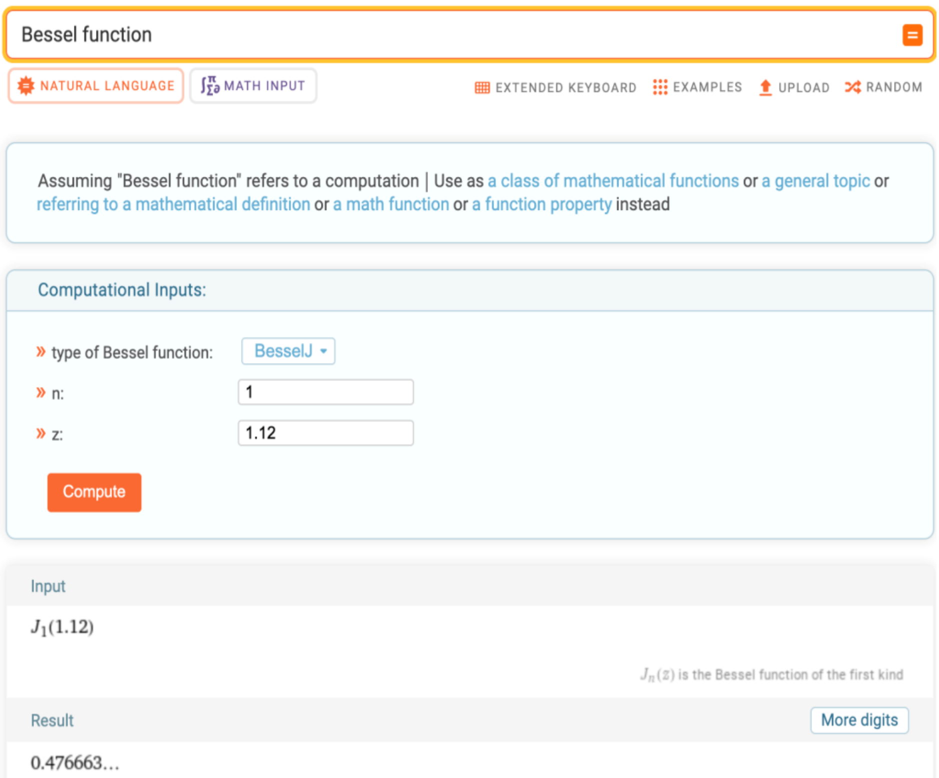

Fig. 3 – *Bessel function calculator at WolframAlpha.*

Apart from the Bessel function of the first kind, this tool also allows for the calculation of values of Bessel functions of the second and third kind, spherical functions, etc. Additionally, different expressions for the function are displayed immediately below the results. One drawback is that users must perform a search to access this calculator (if the user is unaware of this option, they might not search for it and instead enter the function manually). Additionally, the calculator initially displays results with only six decimal places, and users must click a button to view more decimal places. Furthermore, there is no graphical display of the results, so users who want to see a graph must take additional steps to obtain it. In addition to requiring multiple steps to achieve the desired result, which can be frustrating and time–consuming, the process is not overly intuitive. This is understandable because special functions are not the tool's main focus but rather one of its many features.

When discussing *WolframAlpha*, it is inevitable to mention another product of the same company (*Wolfram Research*) – the commercial software package *Mathematica*, which has been in use since 1988, [12]. *Mathematica* is a very popular general–purpose tool that is practically the originator of modern symbolic computation. The significance of this tool is highlighted by the fact that it was the first to offer such a broad range of functionalities: numerical, algebraic, graphic calculations, and more, where previously these capabilities were available only through separate packages designed for specific applications. Over the past three and a half decades, *Wolfram Mathematica* has become a standard tool in engineering research and production. It underpins numerous scientific studies, addresses theoretical problems and questions, plays a crucial role in software development and informatics, and is also utilised for financial modeling, analysis, and planning.

Another commercial software package that has been around longer than *Mathematica* is *Maple*, which was developed at the University of Waterloo in Canada in the early 1980s, [9]. Its goal was to create a tool that could run on low-cost consumer computers, addressing the issue that existing software at the time required highly demanding hardware. At the end of the eighties, the tool was commercialised. It supports numerical calculations, visualisation, and symbolic calculations, and allows for arbitrary precision. Useful ones can be used in standard mathematical notation as well as in programming language.

Another highly popular tool that appears as the top search result is *Symbolab* (`https://www.symbolab.com/solver/functions-calculator`). Similar to the previously described tools, Symbolab is a tool that meets users' various needs in mathematics, physics, chemistry, economics, etc. An key feature of this application lies in its functions. However, the tool does not provide options for specific special functions; instead, users must manually enter the function they wish to calculate. This tool offers basic features for functions and graphs as part of its output, but some essential functionalities are only available in the commercial version. Additionally, the tool does not actually calculate the value in the free version; it only provides the properties of the entered function and its graph. Despite being labelled as a *Functions & Line Calculator*, the value calculation is performed in a different part of the application.

All the tools mentioned are powerful software offering a wide range of functionalities. However, users must go through multiple steps to calculate the desired value, display it graphically, and so on.

5 Conclusion

Special mathematical functions are a special type of mathematical function commonly used in engineering, technical sciences, and physics. This creates a need for a quick and efficient way to approximate their values. The *Calculator of Some Special Mathematical Functions* is an open–source web application that provides an intuitive user interface for quickly and accurately calculating special function values: Bessel functions of the first kind, gamma and beta functions, and some classical orthogonal polynomials – Legendre, Laguerre, Hermite polynomials, Chebyshev polynomials of the first and the second kind, as well as Jacobi polynomials. The main purpose of this tool is to serve as a scientific calculator. However, it can also be applied in educational institutions and scientific research, enabling students and researchers to focus on the application of functions rather than their technical details. The input can be either numeric or a mathematical expression. Consequently, the app includes a parser and algorithms for evaluating expressions in standard mathematical notation. Approximation is achieved through various methods, including the use of other implemented functions, convergent series, and the Lanczos approximation for the gamma function.

Compared to existing software that provides similar functionalities, this application offers a simple interface that enables users to compute precise results in minimal steps and with minimal effort. Furthermore, many existing tools that can perform similar calculations require learning syntax and basic programming concepts. In contrast, this app does not have such requirements, making it accessible and user–friendly, even for those with less experience.

6 Acknowledgments

Bratislav Iričanin received partial support from the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under contract number: 451-03-65/2024-03/200103, and also partial support from the Science Fund of the Republic of Serbia, #GRANT No 7632, Project “Mathematical Methods in Image Processing under Uncertainty – MaMIPU”.

We are grateful to the anonymous reviewers for their insightful and fruitful comments and suggestions, which helped us improve the quality of our article.

7 References

[1] V. Aboites: Laguerre Polynomials and Linear Algebra, Memorias Sociedad Matemática Mexicana, Vol. 52, 2017, pp. 3 – 13.

[2] M. Abramowitz, I. A. Stegun: Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables, Dover Publications, New York, 1965.

[3] G. B. Arfken, H. J. Weber, F. E. Harris: Mathematical Methods for Physicists: A Comprehensive Guide, 7th Edition, Academic Press, Amsterdam, Boston, 2012.

[4] J. K. de Fériet: Fonctions de la Physique Mathematique, 1st Edition, Centre National de la Recherche Scientifique, Paris, 1957.

[5] P. Godfrey: Lanczos Implementation of the Gamma Function, Available at http://www.numericana.com/answer/info/godfrey.htm

[6] B. Iričanin: Actual Computer Algebra Systems and their Future Development, Proceedings of the 6th World MultiConference on Systemics, Cybernetics and Informatics, Orlando, USA, July 2002, pp. 207 – 212.

[7] B. G. Korenev: Bessel Functions and their Applications, 1st Edition, CRC Press, London, 2002.

[8] K. D. Lee, S. Hubbard: Data Structures and Algorithms with Python, 1st Edition, Springer, Berlin, Heidelberg, 2015.

[9] D. S. Mitrinović, R. R. Janić: Uvod u specijalne funkcije, 2nd Edition, Univerzitet: Građevinska knjiga, Belgrade, 1975.

[10] Maplesoft, Maple 2024, Maplesoft, a Division of Waterloo Maple Inc., Waterloo, Ontario.

[11] K. Oldham, J. Myland, J. Spanier: An Atlas of Functions: with Equator, the Atlas Function Calculator, 2nd Edition, Springer, New York, 2009.

[12] M. Tomašević: Algoritmi i strukture podataka, Akademska misao, Belgrade, 2008.
Mathematica, Version 14.1, Wolfram Research, Inc., Champaign, IL, 2024.