Combining Programming and Mathematics Through Computer Simulation Problems

Zarema Seidametova[0000-0001-7643-6386]

Crimean Engineering and Pedagogical University named after Fevzi Yakubov,
8 Uchbovyi Ln., Simferopol, Crimea
z.seydametova@gmail.com

Abstract. Nowadays, educators often use different computer algebra systems for teaching advanced math topics for CS students, and as a tool for solving math problems and providing research. Computer algebra systems allow the students to practice skills both programming and mathematics, that help to develop main components of computational thinking (decomposition, pattern recognition, abstraction, and algorithms). We provide the example of the use one of CAS (Mathematica) for the mathematical research on the $D(s)$-function associated with Riemann Zeta function. For solving this problem, we need to find the algorithm in order to get a mathematically correct results generated by Mathematica.

Keywords: Computer algebra system (CAS), Mathematica, Wolfram software, computational thinking, numerical computing, Riemann zeta function, zeta effect.

1 Introduction

Today technically competent young people can easily use digital devices, know how to connect to GPRS, GPS and start streaming video. At the same time, educators say that traditional forms of educational cognitive activity have fallen. In the 20th century the core skills, that every person needed, were the abilities to read, to write and to count – so-called “3R’s” (Reading, wRiting, aRithmetic). In the 21st century another core skill – Computational Thinking (CT) – was added to these 3R’s. CT, which implies a new way to solve emerging problems with the methods of computer science and engineering, information technology, information systems. First the term “Computational thinking” was introduced in [1]. Seymour Papert discussed new pedagogical approaches in mathematical education in the paper [1]. This term denoted a way of thinking for the algorithmic solution of complex mathematical problems. Later Jeannette M. Wing in the paper [2] developed the computational thinking approach beyond mathematics.

In the paper [3] authors outlined that research team at MIT had developed computing environments designed to facilitate computational thinking (Logo, Scratch) and the use of computer as a computational object. Main components of computational thinking are decomposition, pattern recognition, abstraction, and algorithms. Decomposition demonstrates how to divide complex problems into smaller problems. Pattern recognition shows how to find connections between similar problems and how to use previous experience. Abstraction helps to focus only on the important information without irrelevant details. Using algorithms, we can develop a step-by-step solution to the problem, or the road map to solve the problem.

Computational thinking and programming allow students to learn not only math and programming languages but help them to learn in order to become successful.

For some aspects of computer science students need to know mathematics that is a fundamental course in the educating process of CS professionals. Math helps programmers to solve a problem in an efficient way. Discrete math (set theory, logic, combinatorics), number theory (cryptography and security), geometry (geometric objects, transformations, rotations), linear algebra (matrices, series, differential equations), game theory etc. are math fields that are most important and commonly using in computer science. Math is not directly used in computer science. But computer science students have to think logically and analytically for being good programmers. These are the same types of thinking in solving difficult mathematical problems. Without math, students will face a longer learning curve in programming and vice versa.

In the papers [4], [5], [6], [7] authors provided an overview of educational aspects of math teaching and learning with integrated platforms and computer aided learning software.

Studies related to the effect of computer algebra systems (CAS) on learning efficiency of computer science students presented in papers [8], [9], [10]. The papers [11], [12], [13], [14], [15] presented how to use Mathematica and other CAS for solving math analytical and numerical problems. The ways of using Mathematica as a tool for visualization of the results of the different types of mathematical researches are described in [16], [17], [18], [19], [20], [21], [22], [23]. Some issues about organization of the workspace of a computational system are presented in [24]. There is a bibliography of publications about the Mathematica symbolic algebra language in the [25]. Special advanced math researches using Mathematica as a tool for computing are presented in papers [26], [27], [28], [29], [30]. Topics of the instability that is related to the well-known Gibbs phenomenon [31], [32] and is not in the specifics of the CAS, are presented in [26], [27], [28].

The paper is organized as follows. Section 2 details the advantages of using a computer algebraic software for solving math problems and presents a brief review and comparison of computer algebraic systems. Section 3 presents an advanced math problem that was solved using Mathematica as a main tool. In this section we illustrate the methods and results.

2 Programming, math and computer algebra systems

One way to implement the paradigm of the computational thinking is the use CAS for teaching mathematics and programming at CS departments.

Computer algebra system is a software that helps to manipulate mathematical expressions and mathematical objects, to provide symbolic or numeric computations, to plot different graphics and to visualize math objects. CAS may be divided into two classes:

- specialized, that can be used for solving specific problems of mathematics or statistics;
- general-purpose that can be used for a scientific domain that requires manipulation of mathematical expressions or objects.

The main features of general-purpose CAS are

- a user interface, allowing to enter math formulas or data, and graphics capability;
- a programming language and interpreter;
- a memory manager and garbage collector;
- a rewrite system for simplifying mathematics formulas;
- a large library of mathematical algorithms, special functions, efficient data structures;
- an arbitrary-precision (bignum) arithmetic, needed for calculations are performed on the huge size numbers;
- a fast kernel.

You can see a comparison of most popular CAS in the table 1.

Maple [33] was released by Maplesoft in 1982 as a symbolic and numeric computing environment. It is based on a kernel (written in C) and has libraries (written in Maple language) for performing technical and numeric computations. For storing symbolic expressions Maple uses such data structure as directed acyclic graphs. Maple's interfaces are written in Java. Maple software allows to analyze, explore, visualize, and solve mathematical problems. It can be used in mathematics, smart document environment, application areas, application development, high performance computing, connectivity and education.

Mathcad [34] is computer software product of the Parametric Technology Corporation (PTC) first introduced in 1986. It is used for engineering calculations; results are stored as a notebook. Equations and expressions are created in a worksheet and manipulated in the same graphical format.

The Mathcad functionality contains:

- numerous numeric functions covering such areas as statistics, data analysis and image processing;
- systems of equations (including ordinary and partial differential equations);
- roots of polynomials and functions finder;
- symbolical calculation and manipulation of math expressions;
- parametric, 2D and 3D plotting;
- vector and matrix operations (including eigenvalues, eigenvectors);
- statistical functions, regression analysis on experimental datasets.

Table 1. Popular computer algebra systems

Name of CAS / creator	First / Latest releases	Latest version	Price	License	Notes
Maple / Maplesoft	1984 / 2020	2020.0 (March 2020)	$2,390 (Commercial), $2,265 (Government), $995 (Academic), $239 (Personal Edition), $99 (Student), $79 (Student, 12-Month term)	Proprietary	For symbolic and numeric computing. Written in C, Java, Maple
Mathcad / Mathsoft, PTC	1985 / 2019	6.0.0.0 (October 2019)	$1,600 (Commercial), $105 (Student), Free (Express Edition)	Proprietary	Includes some of the capabilities of CAS. For numerical computing of the engineering problems
Mathematica / Wolfram Research	1988 / 2020	12.1.0 (March 2020)	$2,495 (Professional), $1095 (Education), $295 (Personal), $140 (Student), $69.95 (Student annual license), free on Raspberry Pi hardware	Proprietary	For solving problems in many technical, scientific, engineering, mathematical, and computing fields. Written in Wolfram Language, C/C++, Java
SageMath / William Arthur Stein	2005 / 2020	9.1 (May 2020)	Free	GNU GPL	Open-source system with features covering many aspects of mathematics, including algebra, combinatorics, graph theory, numerical analysis, number theory, calculus and statistics. Written in Python, Cython
Symbolic Math Toolbox (MATLAB) / MathWorks	2008 / 2020	R2020a (March 2020)	$3,150 (Commercial), $99 (Student Suite), $700 (Academic), $194 (Home)	Proprietary	For solving and manipulating symbolic math expressions and performing variable-precision arithmetic.
SymPy / Ondřej Čertík	2007 / 2020	1.6 (May 2020)	Free	modified BSD license	Open-source Python library for symbolic computation.
Wolfram|Alpha / Wolfram Research	2009 / 2020	2020	Pro version: $4.99 per month, Pro version for students: $2.99 per month	Proprietary	Online computational platform or toolkit that encompasses computer algebra, symbolic, numerical computation, visualization.

Wolfram Mathematica [35] is an application for mathematical symbolic calculations that consists of two parts – kernel (back end) and interface (front end). In general, Mathematica is a great tool for solving problems, it integrates all functionalities such as symbolic calculations, manipulations with equations, numeric and graphical outputs. Mathematica offers predefined functions for mathematics, physics, economy, biology and other areas. It is used for calculations in the scientific, engineering, mathematical and computer fields. The Mathematica is also called the CAS Mathematica uses the Wolfram Language. Wolfram Language is a multi-paradigm programming language developed by Wolfram Research for symbolic computing, functional and logical programming, which allows the implementation of arbitrary data structures.

SageMath [36] is free and an open source, python-based alternative to Mathematic, Mathcad, Maple. It uses many python packages, for example, Numpy, Matplotlib, Scipy, Pylab. SageMath has features covering many parts of mathematics – algebra, combinatorics, graph theory, numerical analysis, number theory, calculus and statistics.

MATLAB (MATrix LABoratory) [37] is a software package for high performance numerical computation. It provides an interactive environment with hundreds of built in functions for technical computation, graphics and animation and easy extensibility with its own high-level programming language. MATLAB contains a lot of tools for linear algebra computations, data analysis, signal processing, optimization, numerical solution of Ordinary Differential Equations (ODEs), quadrature and many other types of scientific computations. MATLAB also provides matrix manipulations, parametric, 2D and 3D plotting of functions and data, algorithms implementation, creation of GUI, and interfacing with programs written in other programming languages (C, C++, C#, Java, FORTRAN, Python).

SymPy [38] is an open-source library for symbolic computation that completely written in Python. It provides computer algebra (algebra, matrices, etc.) capabilities either as a standalone application, as a library to other applications, or live application on the web. The SymPy library is split into a core with many optional modules (polynomials, calculus, solving equations, discrete math, matrices, geometry, plotting, physics, statistics, combinatorics, printing).

Wolfram|Alpha [39] is a computational knowledge engine (answer engine) developed by WolframAlpha LLC. Wolfram|Alpha is an online service like a fact-based engine. It answers factual queries directly by computing the answer and does not provide a list of documents or web pages like search engine. Wolfram|Alpha uses technologies that can be divided into four key general areas: a data curation pipeline, an algorithmic computation system, a linguistic processing system, an automated presentation system.

CAS can run under different operating systems natively without emulation. Like most modern apps, Mathematica (except mobile OS), SageMath (except mobile Android OS), MATHLAB (except mobile OS), SymPy run on almost all commonly used OS. Maple, Mathcad do not have versions that run under Android, iOS and SaaS. Moreover, Mathcad does not have versions running under masOS, Linux. We provide a list of OS supporting CAS discussed above (Table 2).

To demonstrate the development of CT skills in teaching CS students advanced topics of mathematics, as well as the development of a heuristic, logical and algorithmic thinking, we used Mathematica 12.0 to solve math problems, which may result in meaningful mathematical problems that lie outside the capabilities of the Wolfram Mathematica, or programming problems will appear that also lie outside the scope of the Wolfram Mathematica, which the CS student have to learn how to solve.

Table 2. Operating systems supporting CAS

CAS	OS						
	DOS	**Windows**	**macOS**	**Linux**	**Android**	**iOS**	**SaaS**
Maple	–	+	+	+	–	–	+
Mathcad	+	+	–	–	–	–	–
Mathematica	–	+	+	+	–	–	+
SageMath	–	+	+	+	–	+	+
Symbolic Math Toolbox (MATLAB)	–	+	+	+	+	–	+
SymPy	–	+	+	+	+	+	+

3 Math project "On the function *D*(*s*) associated with Riemann Zeta function"

We used CAS (Mathematica 12.0 [35]) for investigating the *D*(*s*)-function associated with Riemann Zeta function. Results of the project are presented in paper [26].

Let us consider the function *D*(*s*) of the complex argument $s=\sigma+it$, formed with the use of a certain procedure of a transition to the limit:

$$D(s) = (1-s)\lim_{N\to\infty}\left(\frac{1}{N^{1-s}-1}\cdot\sum_{n=1}^{N}\frac{1}{n^s}\right). \tag{1}$$

It is obvious that for σ>1 the limit of the sum in (1) turns into the Riemann zeta function, and, accordingly:

$$D(s) = (s-1)\zeta(s), \quad (\operatorname{Re} s > 1). \tag{2}$$

Where $\zeta(s) = \sum_{n=1}^{\infty}\frac{1}{n^s}$ is Riemann zeta function.

In the paper [26] was showed that for Re s<1:

$$D(s) \equiv 1 \ (\operatorname{Re} s<1). \tag{3}$$

Therefore, the function *D*(*s*), defined by formula (1) in the entire complex plane (with the exception of the straight line σ=1), in the right-hand half-plane (σ>1) in accordance with formula (2) differs from the Riemann zeta function only by the factor (1–*s*), and in the left half-plane, in accordance with formula (3), it is equal to one. Formula (1), in a certain sense, extends the Riemann series $\sum_{n=1}^{\infty} n^{-s}$ to the entire complex plane s (except for the straight line σ=1).

It is convenient to divide the real and imaginary parts of the function D:

$$D(s) = R(\sigma,t) + iI(\sigma,t) \qquad (4)$$

where

$$\left.\begin{aligned} R(\sigma,t) &= t\cdot \mathrm{B}(\sigma,t) + (\sigma-1)\mathrm{A}(\sigma,t), \\ I(\sigma,t) &= t\cdot \mathrm{A}(\sigma,t) - (\sigma-1)\mathrm{B}(\sigma,t), \end{aligned}\right\} \qquad (5)$$

A and B are the real Riemann series:

$$\left.\begin{aligned} \mathrm{A}(\sigma,t) &= \lim_{N\to\infty} \sum_{n=1}^{N} \frac{\cos(t\ln n)}{n^{\sigma}}, \\ \mathrm{B}(\sigma,t) &= \lim_{N\to\infty} \sum_{n=1}^{N} \frac{\sin(t\ln n)}{n^{\sigma}}. \end{aligned}\right\} \qquad (6)$$

It follows from (6) that the series $\mathrm{A}(\sigma,t)$ and $\mathrm{B}(\sigma,t)$ have very simple asymptotics for large values of σ:

$$\left.\begin{aligned} \mathrm{A}(\sigma,t) &\to 1, \\ \mathrm{B}(\sigma,t) &\to 0, \end{aligned}\right\} \quad \sigma\to\infty.$$

Consequently, in accordance with formulas (5), the asymptotics of $D(s)$ for $\sigma\to\infty$. and fixed t has the following form:

$$\left.\begin{aligned} R(\sigma,t) &\cong \sigma-1, \\ I(\sigma,t) &\cong t. \end{aligned}\right\}$$

When calculating the function $D(s)$ in the right half-plane of the complex argument s (for σ>1) we used finite-dimensional approximations of the oscillating real Riemann series (6). Instead of formulas (6) containing the limiting transition $N\to\infty$, we used finite sums:

$$\left.\begin{aligned} \mathrm{A}_N(\sigma,t) &= \sum_{n=1}^{N} \frac{\cos(t\ln n)}{n^{\sigma}}, \\ \mathrm{B}_N(\sigma,t) &= \sum_{n=1}^{N} \frac{\sin(t\ln n)}{n^{\sigma}}. \end{aligned}\right\} \qquad (7)$$

When computing the sums A_N and B_N we usually fix N in the range between $N = 10^5$ and $N = 10^6$. A calculation with a smaller value of N introduced noticeable distortions in the results. Computations with large values N required an unacceptably high time-consuming result. For N in the range $10^5 - 10^6$ one calculation, – for example, plotting the dependence of $R(t)$ for fixed σ and $0 \le t \le 50$ – requires, depending on σ and N from several tens of minutes to several work hours for Mathematica. You can see codes of plotting the R-function at listing 1, and symbolic results at Fig. 1.

Listing 1. Codes for plotting the *R*-function as the function of the argument *t* for σ>1.

```
A[x_, y_] = N[Sum[{Cos[y Log[n]] / n^x}, {n, 1, 600000}]]
B[x_, y_] = N[Sum[{Sin[y Log[n]] / n^x}, {n, 1, 600000}]]
R[x_, y_] = y B[x, y] + (x - 1) A[x, y]
Plot[R[1.05, y], {y, 0.01, 50}, PlotTheme → "Monochrome",
 PlotRange → Automatic, Axes → True, AxesLabel → {t, R},
 LabelStyle → {14, GrayLevel[0]}, Frame → False, PlotLegends → "Expressions"]
```

```
(Debug) Out[•]=
{(-1 + x)
  (1. + 2.^-1.× Cos[0.693147 y] + 3.^-1.× Cos[1.09861 y] + 4.^-1.× Cos[1.38629 y] +
   5.^-1.× Cos[1.60944 y] + 6.^-1.× Cos[1.79176 y] + 7.^-1.× Cos[1.94591 y] +
   8.^-1.× Cos[2.07944 y] + 9.^-1.× Cos[2.19722 y] + 10.^-1.× Cos[2.30259 y] +
   11.^-1.× Cos[2.3979 y] + 12.^-1.× Cos[2.48491 y] + ··· 599977 ··· +
   599990.^-1.× Cos[13.3047 y] + 599991.^-1.× Cos[13.3047 y] +
   599992.^-1.× Cos[13.3047 y] + 599993.^-1.× Cos[13.3047 y] +
   599994.^-1.× Cos[13.3047 y] + 599995.^-1.× Cos[13.3047 y] +
   599996.^-1.× Cos[13.3047 y] + 599997.^-1.× Cos[13.3047 y] +
   599998.^-1.× Cos[13.3047 y] + 599999.^-1.× Cos[13.3047 y] +
   600000.^-1.× Cos[13.3047 y]) + y (··· 1 ···)}
large output   show less   show more   show all   set size limit...
```

Fig. 1. Symbolic results of the *R*-function of the argument *t* for σ=1.05 (N=6·10^5).

Fig. 2 demonstrates, in addition to real "slow" *t*-oscillations of sufficiently large amplitude, also the presence of an interesting effect of short-period "parasitic" oscillations of small amplitude. This effect (we called it the "zeta effect") is generated by a sharp break in the Riemann series (6) for a finite (albeit sufficiently large) value of n, equal to the "cut-off parameter" N $N \cong 10^5 - 10^6$. In Fig. 2, these zeta oscillations significantly deform the dependence $R = R(t)$ up to $t \approx 15$, but are also noticeable at t>15.

A qualitative explanation of the nature of the "zeta effect" is given in the paper [26]. In some extent, this "zeta effect" is one of the version of the Gibbs phenomenon related to the Fourier series theory [32].

How can we suppress these parasitic zeta-oscillations generated by the termination of infinite Riemann series (5)?

To suppress the zeta effect, we used "exponential β-damping", replacing each term in the Riemann sums (7) with its "damped" expression:

$$\left.\begin{aligned} \mathrm{A}_N(\sigma,t) \to \mathrm{A}_{N,d}(\sigma,t,\beta) = \sum_{n=1}^{N} \frac{\cos(t\ln n)}{n^{\sigma}} e^{-\beta \frac{n}{N}}, \\ \mathrm{B}_N(\sigma,t) \to \mathrm{B}_{N,d}(\sigma,t,\beta) = \sum_{n=1}^{N} \frac{\sin(t\ln n)}{n^{\sigma}} e^{-\beta \frac{n}{N}}. \end{aligned}\right\} \tag{8}$$

Where β is the damping parameter: $\beta > 1$. (In the calculations, we used the value β=5).

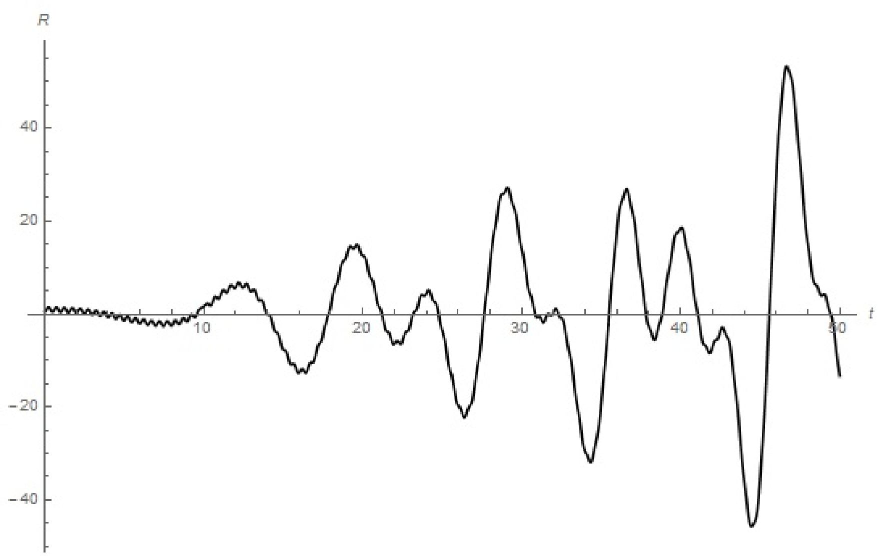

Fig. 2. The R-function as the function of the argument t for σ=1.05 (N=6·10^5).

Exponential β-damping (8) does not significantly affect the contribution of the majority of "low-frequency" harmonics with $n << N$ and substantially reduces the contribution of terms with large numbers n, approaching to $n=N$. This method smooths out the effect of a sharp break in the Riemann series (5) for $n=N$.

Fig. 3 demonstrates the effect of exponential β-damping on the numerical results of calculating the function $D(s)$. This figure shows the same graph shown earlier in Fig. 2: dependence of the function R on the argument t for σ=1.05 and $N = 6 \cdot 10^5$. In Fig. 3 this dependence is calculated by the formulas (14), taking into account β-damping (β=5). The "smoothed" function $R(t)$ in this figure completely repeats the function $R(t)$ of Fig. 2 in all that concerns real large-scale slow oscillations, but is practically free of parasitic oscillations generated by the zeta effect. The trace of these parasitic oscillations remained only for small t (t < 1). The suppression of the zeta-effect in the region of small t requires an increase in the damping parameter β. An increase in β can cause distortion in real large-scale oscillations of the function $R(t)$. Here, the researcher must compromise, determining what is more important in a particular task – total suppression of the zeta-effect at small t or preservation of correct results for large t.

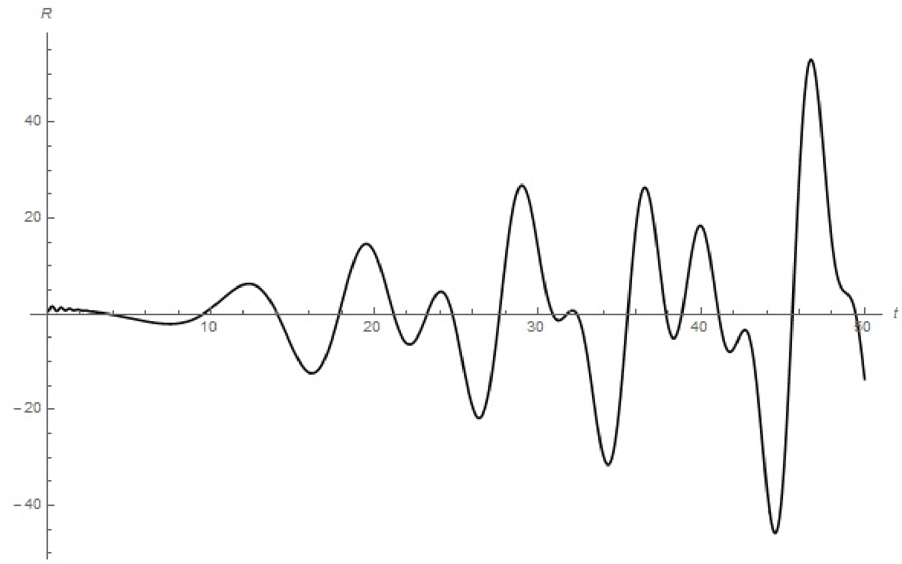

Fig. 3. The dependence of the function *R* of argument *t* for σ=1.05 (N=6·10^5). The dependence is calculated using the exponential β-damping procedure described in the article for β=5.

4 Conclusions

Of the many problems that the author had to solve (on her own or in collaboration with colleagues and students) using various CAS packages, the author chose one problem here for purely illustrative purposes. In this task, the mathematics package used by itself works flawlessly, but you need to reflect on the algorithm in order to get a mathematically correct result using this package that suppresses some kind of pure instability. This instability is inherent in the task itself, and not in the specifics of the Mathematics package. In some sense, this instability is related to the well-known Gibbs phenomenon, which consists in a certain instability of the process of numerical summation of Fourier series in some situations.

In other problems, there may be situations where Wolfram Mathematica cannot perform certain actions at all (for example, plotting a curve containing cusp [27]) or perform calculations with the necessary accuracy in a reasonable time (for example, constructing fractal curves defined by Riemann-Weierstrass series).

References

1. Papert, S.: An exploration in the space of mathematics educations. International Journal of Computers for Mathematical Learning **1**, 95–123 (1996). doi: 10.1007/BF00191473
2. Wing, J.M.: Computational thinking. Communications of the ACM **49**(3), 33–35 (2006). doi:10.1145/1118178.1118215

3. Bull, G., Garofalo, J., Hguyen, N.R.: Thinking about computational thinking: Origins of computational thinking in educational computing. Journal of Digital Learning in Teacher Education **36**(1), 6–18 (2020). doi:10.1080/21532974.2019.1694381
4. Chmielewska, K., Gilanyi, A.: Educational Context of Mathability. Acta Polytechnica Hungarica **15**(5), 223–237 (2018)
5. Meusel, M.: Computational design of an integrated learning and assessment platform for higher education. Dissertation, University of Zurich (2019)
6. Nneji, G.U., Deng, J., Shakher, S.S., Monday, H.N., Agomuo, D., Ukwuoma, C.C.: A Multimedia Computer Aided Learning Software. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 807–813, Vancouver, BC (2018). doi:10.1109/IEMCON.2018.8614770
7. Kfoury, A.: Personal Reflections on the Role of Mathematical Logic in Computer Science. Fundamenta Informaticae **170**(1-3), 207–221 (2019). doi:10.3233/FI-2019-1860
8. Güyer, T.: Computer algebra systems as the mathematics teaching tool. World Applied Sciences Journal **3**(1), 132–139 (2008)
9. Barba-Guaman, L.R., Quezada-Sarmiento, P.A., Calderon-Cordova, C.A., Sarmiento-Ochoa, A.M., Enciso, L., Luna-Briceno, T.S., Conde-Zhingre, L.E.: Using Wolfram software to improve reading comprehension in mathematics for software engineering students. In: 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–4. IEEE (2018). doi:10.23919/CISTI.2018.8399388
10. León, J.G.S.: Mathematica beyond mathematics: The Wolfram language in the real world. Chapman and Hall/CRC, London (2017)
11. Kristalinskii, V.R., Chernyi, S.N.: On solving dynamic programming problems in the Wolfram Mathematica system. International Journal of Open Information Technologies 7(2), 42–48 (2019)
12. Mokhasi, P., Adduci, J., Kapadia, D.: Understanding differential equations using Mathematica and interactive demonstrations. CODEE Journal **9**(1), 8 (2012). doi:10.5642/codee.201209.01.08
13. Abbena, E., Salamon, S., Gray, A.: Modern differential geometry of curves and surfaces with Mathematica, 4th edn. Chapman and Hall/CRC, London (2017)
14. Czajkowski, A A., Sidorkiewicz, K.: Analytical and numerical solving of general triangles with application of numerical programs MS-Excel, MathCAD and Mathematica. Problemy Nauk Stosowanych **3**, 15–32 (2015)
15. Durán, A.J., Pérez, M., Varona, J.L.: The Misfortunes of a Trio of Mathematicians Using Computer Algebra Systems. Can We Trust in Them? Notices of the AMS **61**(10), 1249–1252
16. Wolfram, S.: Differential Equation Solving with DSOLVE. Wolfram Research, Champaign. https://reference.wolfram.com/language/tutorial/DSolveIntroduction.html (2008). Accessed 21 Mar 2020
17. Torrence, B.F., Torrence, E.A.: The Student's Introduction to Mathematica and the Wolfram Language. Cambridge University Press, Cambridge (2019)
18. Mureşan, M.: About *Mathematica*. In: Introduction to Mathematica® with Applications, pp. 1–3. Springer, Cham (2017). doi:10.1007/978-3-319-52003-2_1
19. Mureşan, M.: First Steps to Mathematica. In: Introduction to Mathematica® with Applications, pp. 5–12. Springer, Cham (2017). doi:10.1007/978-3-319-52003-2_2
20. Trott, M.: The Mathematica GuideBook for Programming. Springer, New York (2004). doi:10.1007/978-1-4419-8503-3
21. Borwein, J.M., Skerritt, M.P.: An Introduction to Modern Mathematical Computing: With Mathematica®. Springer Science & Business Media, Cham (2012)

22. Trefethen, L.N.: Computing numerically with functions instead of numbers. Communications of the ACM **58**(10), 91–97 (2015). doi:10.1145/2814847
23. Tomiczkova, S., Lavicka, M.: Using dynamic geometry and computer algebra systems in problem-based courses for future engineers. The International Journal for Technology in Mathematics Education **22**(4), 147–154 (2015). doi:10.1564/tme_v22.4.02
24. Gray, T.W., Wolfram, S.: Method and system for presenting input expressions and evaluations of the input expressions on a workspace of a computational system. US. Patent 8,407,580, 26 Mar 2013
25. Beebe, N.H.F.: A Bibliography of Publications about the *Mathematica* Symbolic Algebra Language. http://www.netlib.org/tex/bib/mathematica.pdf (2020). Accessed 21 Mar 2020
26. Seidametova, Z.S., Temnenko, V.A.: On the Function $D(s)$ Associated with Riemann Zeta Function. International Journal of Mathematics and Statistics Invention **6**(4), 1–12. http://www.ijmsi.org/Papers/Volume.5.Issue.4/A05040112.pdf (2017)
27. Seidametova, Z., Temnenko, V.: Euler's Insignia: Some Admirable Curves Having a Simple Trigonometric Equation in a Natural Form. The College Mathematics Journal **50**(2), 134–139 (2019). doi:10.1080/07468342.2019.1562825
28. Seidametova, Z., Temnenko, V.: Some Geometric Objects Related to a Classical Problem of Galileo. The College Mathematics Journal **51**(1), 57–65 (2020). doi:10.1080/07468342.2020.1681871
29. Sun, Z., Shu, C. W.: Strong stability of explicit Runge-Kutta time discretizations. SIAM Journal on Numerical Analysis **57**(3), 1158–1182 (2019). doi:10.1137/18M1230700
30. Rocco, A., West, B.J.: Fractional calculus and the evolution of fractal phenomena. Physica A: Statistical Mechanics and its Applications **265**(3-4), 535–546 (1999). doi:10.1016/S0378-4371(98)00550-0
31. Borwein, J., Bradley, D.M., Crandall, R.: Computational Strategies for the Riemann Zeta Function. Journal of Computational and Applied Mathematics **121**(1), 247–296 (2000). doi:10.1016/S0377-0427(00)00336-8
32. Weisstein, E.W.: Gibbs Phenomenon. MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/GibbsPhenomenon.html (2020). Accessed 21 Mar 2020
33. Maple - The Essential Tool for Mathematics - Maplesoft. https://www.maplesoft.com/products/Maple (2020). Accessed 21 Mar 2020
34. Mathcad | Mathcad. https://www.ptc.com/en/products/mathcad (2019). Accessed 21 Mar 2020
35. Wolfram Mathematica: Modern Technical Computing. https://www.wolfram.com/mathematica (2020). Accessed 21 Mar 2020
36. SageMath - Open-Source Mathematical Software System. https://www.sagemath.org (2020). Accessed 21 Mar 2020
37. MATLAB - MathWorks - MATLAB & Simulink. https://www.mathworks.com/products/matlab.html (2020). Accessed 21 Mar 2020
38. SymPy. https://www.sympy.org (2020). Accessed 21 Mar 2020
39. Wolfram|Alpha: Computational Intelligence. https://www.wolframalpha.com (2020). Accessed 21 Mar 2020