Application of Mathematical Software in Teaching Numerical Mathematics for Students of Electrical Engineering

Nataša Ćirović
University of Belgrade, School of Electrical Engineering, Belgrade, Serbia
natasa@etf.bg.ac.rs

Abstract: The successful education of numerical mathematics for engineering students requires usage of software tools for various applications. In this paper we present our experiences in using both open and proprietary software tools in teaching numerical mathematics for bachelor students of electrical engineering at the School of Electrical Engineering, University of Belgrade in different applications. We compare the mathematical software used for implementation of numerical solutions in class and present an illustrative example showing the application of mathematical software in teaching process.

Keywords: numerical mathematics; higher education; open software; proprietary software.

I. Introduction

With the increase of computer power in the past few decades, the nature of introductory courses in numerical mathematics for students of engineering is changing from purely theoretical to application driven implementation with ready-to-use tools. Software intended for numerical computation is continuously being developed and improved. The usage of such tools in the classroom allows the shift in focus from tedious repeated computation by-hand to getting insight in the limitations of various numerical methods.

During bachelor studies at the School of Electrical Engineering, University of Belgrade, numerical methods are thought within courses Numerical analysis and discrete mathematics for Computer Engineering and Information Theory and Software Engineering students, and Numerical mathematics for Electrical Engineering second year students. Part of the students' grade is realized by the implementation of programming assignments in mathematical software or programming language of choice. Various tools are used during the teaching process by the lecturers and implementation of programming assignments by the students. In this paper we present and compare the software tools, both propriety and open, used for the needs of numerical mathematics courses for various applications.

II. Software applications

Software tools are used for various applications in teaching numerical mathematics courses for students of electrical engineering. These include online educational platforms as administrative support to the educational process, text processing tools and mathematical software tools. Also, tools for spreadsheet manipulation have shown useful in the educational process.

A. Educational platforms

As support in administrating the educational process online educational platforms are used, which facilitate access to additional learning material such as examples of solved problems in various mathematical software used in the classroom. Also, selection, uploading and grading of programming assignments implemented by students is conducted through these platforms.

The Computer Center of the University of Belgrade hosts and administrates the eLearning platform based on **Moodle** [1]. Moodle is a learning management system and it is a free and open source software package designed to provide educators, administrators, and learners with a single robust, secure and integrated system to create personalized learning environments [2]. The capabilities of Moodle greatly depend on the administrative settings done by the host.

During the spring semester 2020 the COVID-19 pandemic forced the learning process from the classroom to the online environment. School of Electrical Engineering, University of Belgrade provided the **Microsoft Teams** (MST) platform for online teaching. Even though MST was not primarily developed as learning platform, it has all the necessary features such as assignment posting and grading. Live streaming is very easily implemented through MST, together with live recording, which was essential in the situation when no on-site classes were possible. The feature of live streaming is not directly available in Moodle, but it could possibly be realized through additional plug-ins.

B. Text processing tools

Text processing software tools are utilized both by students and teachers in various applications.

LaTeX [3] is a free and open source document preparation system. It uses markup-level directions to define the formatting. Typing and formatting mathematical equations is easily implemented through LaTeX editors. For offline installation we have good experiences with MiKTeX [4] with TeXstudio editor [5] on Microsoft Windows operating systems. Overleaf [6], an online collaborative LaTeX editor, allows fast editing without installation, which can be very useful feature for students. All the above mentioned software tools related to LaTeX are open software.

LibreOffice Writer is part of LibreOffice, one of the leading open source office software suite. LibreOffice was forked from OpenOffice.org in 2010, that contains

DOI: 10.5281/zenodo.4081705

programs for word processing, creating and editing of spreadsheets, presentations, diagrams and drawings, databases, and composing mathematical formulas.

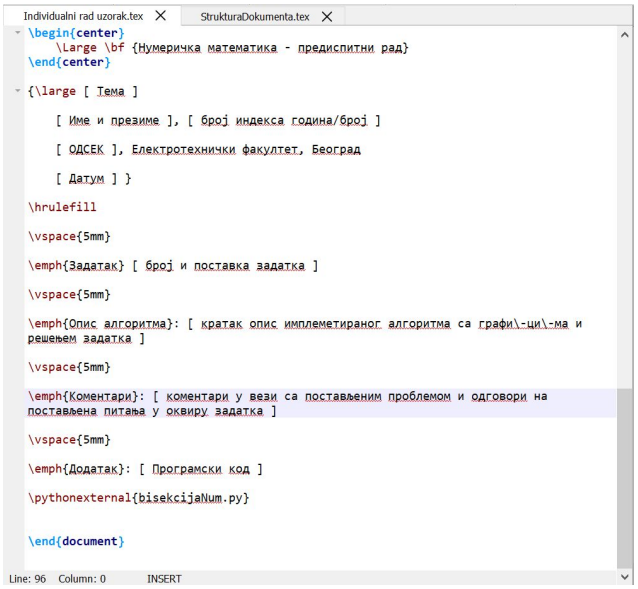


Figure 1: Part of LaTeX template code for programming assignment report with automatic importing of the program code

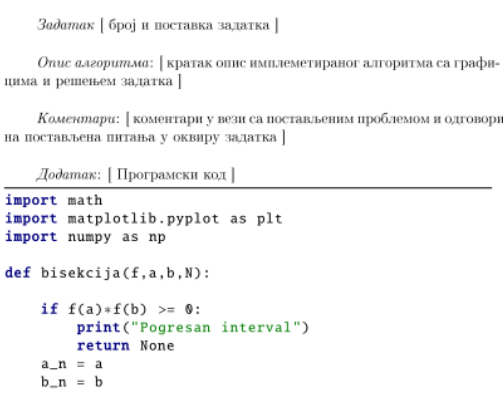


Figure 2: Part of LaTeX template pdf output for programming assignment report with automatic importing of the program code

As its native file format LibreOffice uses the Open Document Format for Office Applications (ODF), and it also supports the file formats of most other major office suites, including Microsoft Office, through a variety of import and export filters [7].

Microsoft Word is part of Microsoft Office, a propriety office package with wide range of applications including word processing, spreadsheet, presentation, but also e-mail client and MS Teams app. Since Office 2013, Microsoft has promoted Office365 which allows the use of the software and other services on a subscription business model. Since 2017 School of Electrical Engineering University of Belgrade implemented Office365 cloud platform for students with a large number of online services [8].

The official literature for the numerical mathematics courses is the electronic textbook by the author of this paper and it is freely available through the School website [9]. The book is prepared in LaTeX with TeXstudio editor and the graphics are prepared in GeoGebra [10], an interactive mathematics software suit for learning and teaching mathematics.

Text processing applications are utilized by students for creation of programming assignment reports. Two templates are available for reporting, in LibreOffice Writer/Microsoft Word and in LaTeX. Figure 1 shows part of the LaTeX code for reporting programming assignment template using the package *listings* for source code printing, and Figure 2 shows the output of this code.

C. Mathematical software tools

Programming assignments are part of the grade and students need to solve two problems that require programming solution in mathematical software tools or programming language of their own choice. During the teaching process the mathematical software is also used to provide deeper insight in various numerical methods.

Spreadsheet software is used for automatic calculations. Even though spreadsheet software allows writing programs (macros), in classroom we implement numerical methods by using the worksheet formulas. This approach has shown especially useful in the teaching process of numerical mathematics, allowing students to go through the steps of the numerical method in detail while avoiding tiresome calculation by hand. It is suitable for solving problems of lower dimensions which are typically considered within the undergraduate numerical mathematics courses [11]. For this purpose LibreOffice Calc or Microsoft Excel can be used with equal success.

Besides the spreadsheet applications, mathematical software tools developed particularly for numerical mathematical computations is also used in courses. A number of software packages have been developed to produce numeric and symbolic mathematical computations. The predominant propriety packages in the academic environment are **Maple** [12], **Mathematica** [13] and **MATLAB** [14], [15]. In parallel, a number of free and open source packages were developed during the past decades, with more or less success.

Maple is a proprietary software created and developed by Maplesoft company, primarily for numeric and symbolic mathematical computations. Within the project supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia in 2019 we were able to obtain the license for Maple, which enabled us to introduce Maple in the teaching process of numerical mathematics. Maple includes the Student package, a collection of routines designed to assist the teaching and learning of standard undergraduate mathematics. It includes a sub-package specifically designed for numerical analysis topics and it contains computation commands, visualization commands and interactive routines, allowing the students to gain insight on how a specific numerical method operates [12].

GNU Octave is one of the most successful alternatives to MATLAB. It is compatible with many MATLAB scripts and has available online version, which

is an advantage for students with modest technical capacities [16].

SageMath is an open source mathematical software built on top of many existing open source packages with a unified Python interface. SageMath can be used on a local computer, in a local network and online. The online version is called CoCalc and it can operate as a collaborative and educational platform as well [17], [18].

Python is a high-level general purpose programming language [19]. It is highly extensible, fast-growing environment, allowing creation of large number of libraries that support numerical and symbolic computation. Some of the most valuable libraries that support numerical methods include:

- *NumPy*, defines a multi-dimensional array object and associated fast math functions that operate on it [20];
- *SymPy*, a Python library for symbolic mathematics, aiming to become a full-featured computer algebra system [21];
- *SciPy*, a Python library build on NumPy, used for solving scientific and mathematical problems. It allows manipulation and visualization of data with a wide range of high-level commands. SciPy includes modules for linear algebra, optimization, integration, special functions, ODE solvers [22];
- *matplotlib*, a comprehensive library for creating static, animated, and interactive visualizations in Python [23];

SciPy with matplotlib library allows creation of interactive routines and animations of various numerical methods. There is a large number of routines available online that are created with the aim to improve the learning process and to provide visualization of various numerical methods. Also, using the above mentioned Python libraries, it is not hard to create such routines, even for inexperienced programmers.

Other general purpose programming languages can also be successful in programming numerical methods, such as **C**, **C++**, **C#**, **Java**.

In our past experience with programming assignments we found that students tend to use the software that they are already familiar with. Thus, students of Computer Engineering and Software Engineering mostly implement their assignments in Python or C++ while students of Electrical Engineering mostly use MATLAB/GNU Octave. Since both MATLAB and GNU Octave use file with m extension and having in mind that GNU Octave is compatible with many MATLAB scripts, it is often hard to tell in which package the code was originally written.

III. Classroom example

We present an example with the aim to illustrate the application of mathematical software tools in teaching numerical mathematics methods for students of electrical engineering. Here we show how we can implement the solution of nonlinear equation with the Newton-Raphson iterative method using Maple Student package, Python and LibreOffice Calc. As an illustrative example we solve the equation $(x-1)e^{3x}=0$. Knowing the correct solution, $x=1$, allows us to compare the exact error in each iteration.

A. Maple Student package

For the implementation of the Newton-Raphson method we use the command *Newton* within the Student [NumericalAnalysis] subpackage. The *output* option displays the sequence of iteration values, *plot* option provides geometric interpretation of the method and *animation* option enables animated display of the iterative process. Figure 3 shows the Maple code and graphical interpretation for our example function.

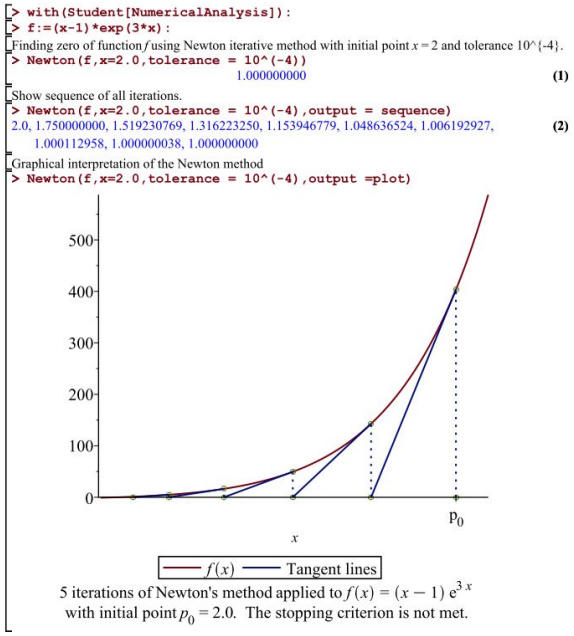


Figure 3: Maple code and output for implementation of Newton-Raphson method with Student package.

B. Python with libraries

SciPy library contains a large number of routines for implementation of numerical methods. For the solution of our example problem $(x-1)e^{3x}=0$, we use the scipy.optimize.newton routine. Additionally, we created the routine that shows the graphical interpretation of Newton-Raphson method using the SciPy and matplotib libraries. Figure 4 shows part of the python code with iteration values and graphical interpretation of Newton-Raphson method implemented on our example problem.

C. LibreOffice Calc

When introducing a numerical method in class we often use spreadsheet software and worksheet formulas. Here we show the solution of our example problem $(x-1)e^{3x}=0$ in LibreOffice Calc. Figure 5 shows the table obtained during implementation of the Newton-Raphson method for solving our example problem. We compare two adjacent iteration values and the actual error.

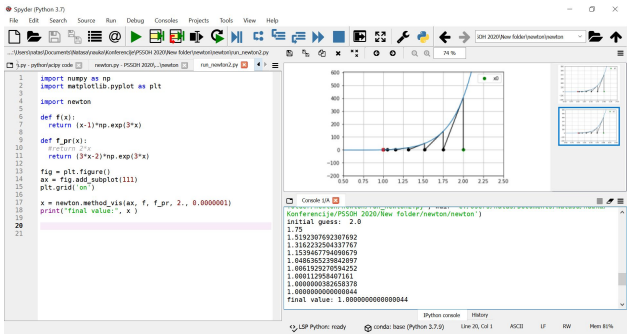
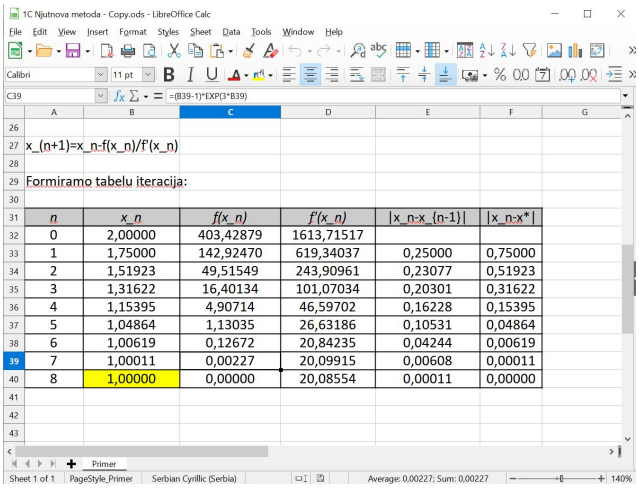
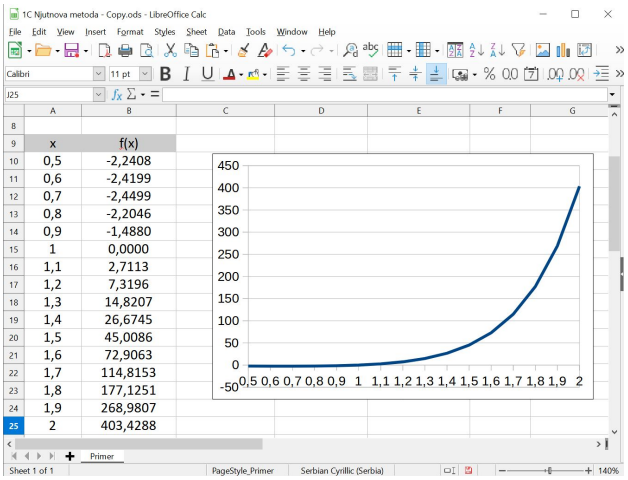


Figure 4: Part of the python code with iteration values and graphical interpretation of Newton-Raphson method for solving $(x-1)e^{3x}=0$.



n	x_n	f(x_n)	f'(x_n)	|x_n-x_{n-1}|	|x_n-x*|
0	2,00000	403,42879	1613,71517		
1	1,75000	142,92470	619,34037	0,25000	0,75000
2	1,51923	49,51549	243,90961	0,23077	0,51923
3	1,31622	16,40134	101,07034	0,20301	0,31622
4	1,15395	4,90714	46,59702	0,16228	0,15395
5	1,04864	1,13035	26,63186	0,10531	0,04864
6	1,00619	0,12672	20,84235	0,04244	0,00619
7	1,00011	0,00227	20,09915	0,00608	0,00011
8	1,00000	0,00000	20,08554	0,00011	0,00000

Figure 5: Table obtained during implementation of the Newton-Raphson method in LibreOffice Calc



x	f(x)
0,5	-2,2408
0,6	-2,4199
0,7	-2,4499
0,8	-2,2046
0,9	-1,4880
1	0,0000
1,1	2,7113
1,2	7,3196
1,3	14,8207
1,4	26,6745
1,5	45,0086
1,6	72,9063
1,7	114,8153
1,8	177,1251
1,9	268,9807
2	403,4288

Figure 6: Graphic of the function $f(x)=(x-1)e^{3x}$ *created in LibreOffice Calc using linear chart.*

Graphing of functions in LibreOffice Calc can be realized by using the chart of linear type on function values, which basically creates linear interpolation of data. This approach does not handle function discontinuities, so one should be especially careful when dealing with function graphing in this way. Figure 6 shows the result of graphing our example function $f(x)=(x-1)e^{3x}$ with linear chart in LibreOffice Calc.

IV. Conclusion

Software tools are used in bachelor numerical mathematics courses for 3 main applications: educational platforms, text processing and mathematical software tools. While educational platforms and text processing tools are used irrelevant of topic, mathematical software is crucial in numerical mathematics courses for understanding and implementing numerical methods. Spreadsheet software tools have shown very useful when a numerical method is introduces in class. Both open source LibreOffice Calc and proprietary Microsoft Excel can be successfully used. Numerical software allows more detailed analysis of numerical methods. For this purpose the Maple Student package, a proprietary software, and Python with SciPy and mathplotlib are used. In comparison to Maple, Python may require creation of new routines, but many are already freely available. Also, such routines can be easily created for any numerical method in Python, while Maple Student package covers only the most common numerical methods. Other relevant software can be used with the same purpose and we plan to include SageMath in teaching process. For implementation of programming assignments students tend to use tools that they are already familiar with and that are available.

Acknowledgements

This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia through project NUMMMO [RVO-41/2019].

References

[1] http://elearning.rcub.bg.ac.rs [Accessed Aug. 25, 2020]

[2] E. Costello, "Opening up to open source: looking at how Moodle wasadopted in higher education", *Open Learning: The Journal of Open, Distance and e-Learning*, vol. 28, no. 3, pp. 187-200, 2013, doi: 10.1080/02680513.2013.856289

[3] https://www.latex-project.org [Accessed Aug. 25, 2020]

[4] https://miktex.org [Accessed Aug. 25, 2020]

[5] https://texstudio.org [Accessed Aug. 25, 2020]

[6] https://www.overleaf.com [Accessed Aug. 25, 2020]

[7] https://www.libreoffice.org [Accessed Aug. 25, 2020]

[8] https://www.microsoft.com/en-us/microsoft-365 [Accessed Aug. 25, 2020]

[9] N. Ćirović, "Numerička matematika", School of Electrical Engineering, University of Belgrade, Belgrade, 2018, https://www.etf.bg.ac.rs/uploads/files/udzbenici/Natasa%20Cirovic%20Numericka%20matematika%20CIP.pdf

[10] https://www.geogebra.org [Accessed Aug. 25, 2020]

[11] H. Djamila, "Excel spreadsheet in teaching numerical methods", *IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 890, 012093, 2017, doi:10.1088/1742-6596/890/1/012093

[12] https://www.maplesoft.com/ [Accessed Sep. 1, 2020]

[13] https://www.wolfram.com/mathematica/ [Accessed Sep. 1, 2020]

[14] https://www.mathworks.com/ [Accessed Sep. 1, 2020]

[15] R.L. Burden and J.D. Faires, "Numerical Analysis", 9th Edition, Brooks/Cole, Cengage Learning, 2011.

[16] https://www.gnu.org/software/octave/ [Accessed Sep. 1, 2020]

[17] https://www.sagemath.org/ [Accessed Sep. 1, 2020]

[18] P. Szabó and J. Galanda, "Sage math for education and research," 15th Int. Conf. on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, 2017, pp. 1-4, doi: 10.1109/ICETA.2017.8102535

[19] https://www.python.org/ [Accessed Sep. 1, 2020]

[20] https://numpy.org/ [Accessed Sep. 1, 2020]

[21] https://www.sympy.org [Accessed Sep. 1, 2020]

[22] https://scipy.org/ [Accessed Sep. 1, 2020]

[23] https://matplotlib.org/ [Accessed Sep. 1, 2020]