UDC 510.2; 621.3

DOI: 10.20535/2523-4455.mea.359022

Software Support for the Higher Mathematics Course at the Technical University

Part 3. GNU Octave

D. Voitsikh[f], Bachelor student
D. Hryn[f], Bachelor student
M. Hryniuk[f], Bachelor student
O. V. Bogdanov, Assoc. Prof., PhD, 0000-0002-0911-5563
Yu. P. Butsenko, Assoc. Prof., PhD, 0000-0003-4806-9587
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" 00syn5v21
Kyiv, Ukraine

O. I. Balina, Assoc. Prof., PhD, 0000-0001-6925-0794
I. S. Bezklubenko, Assoc. Prof., PhD, 0000-0002-9149-4178
Kyiv National University of Construction and Architecture 02qp15436
Kyiv, Ukraine

Abstract—This paper explores a methodological approach to modernizing the higher mathematics curriculum for engineering students, with a specific focus on the needs of “G5 Electronics, electronic communications, instrumentation and radio engineering” specialty. The study addresses the growing challenge posed by the "black box" paradigm, in which students rely on automated online calculators and artificial intelligence tools without comprehending the underlying mathematical logic or algorithmic structures. To counteract this trend, the authors propose integrating GNU Octave, an open-source computational environment, as a primary tool to bridge the gap between abstract theory and professional engineering practice.

The article demonstrates the practical implementation of GNU Octave across three critical mathematical domains: Integral Calculus, emphasizing symbolic and numerical methods for signal processing applications; Multivariable Functions, focusing on 3D visualization techniques (using meshgrid and surf) for modeling spatial physical fields and potential distributions; Differential Equations, utilizing numerical solvers like ode45 to simulate transient processes in electronic circuits.

The results suggest that transitioning from simplified automated tools to script-based computational modeling fosters a conscious mastery of mathematical concepts. This approach ensures that future engineers develop the analytical and programming competencies needed for complex system optimization and professional modeling in modern electronics.

Keywords — Engineering education; Teaching engineering; Electronics engineering; GNU Octave; Numerical analysis; Mathematical modeling; Circuit analysis; Visualization; Educational technology.

I. INTRODUCTION

The rapid evolution of contemporary electronics, telecommunications, and instrumentation [1], [2] necessitates a paradigm shift in engineering education. Today’s electronics specialist must analyze complex signals, model dynamic processes in semiconductor devices, and optimize multi-component system parameters—tasks that demand a rigorous mathematical foundation paired with proficiency in specialized computational software.

The urgency of modernizing the higher mathematics curriculum is driven by several critical factors:

The fundamental role of mathematical modeling in electronics. Differential calculus serves as the cornerstone for analyzing the rate of change in electrical signals and modeling transient responses in RC/RL circuits. Integral calculus is indispensable for determining signal energy characteristics and conducting spectral analysis via Fourier transforms. Furthermore, linear algebra and the theory of differential equations provide the essential framework for complex circuit analysis and the development of control algorithms, such as PI controllers.

The technological gap between theory and practice. Traditional pedagogical approaches often concentrate on "textbook" problems with simplified analytical solutions. However, real-world engineering challenges often require numerical methods, as many physical processes lack elementary closed-form antiderivatives or are analytically complex and defy manual calculation.

DOI: 10.20535/2523-4455.mea.359022

The "Black Box" paradigm and academic integrity. Modern students increasingly rely on online calculators and AI-driven tools that provide immediate solutions without revealing the underlying algorithmic logic. This trend leads to a superficial understanding of the subject matter and an inability to apply mathematical principles to practical professional scenarios.[3], [4]

II. Pedagogical Strategy and Software Selection

Addressing these challenges requires integrating professional computational environments, such as GNU Octave, into the learning process. As a free, open-source alternative to the industry-standard MATLAB, GNU Octave bridges mathematical theory with programming and numerical modeling. This approach empowers students to implement solution algorithms independently, visualize high-dimensional function surfaces, and evaluate the stability of electronic systems through dynamic simulation.

Building upon our previous research [5], [6], which addressed the fundamental aspects of software-assisted mathematics instruction and basic function analysis, this paper explores the practical application of GNU Octave in teaching integral calculus, multivariable functions, and differential equations. The focus is specifically tailored to the technical requirements of “G5 Electronics, electronic communications, instrumentation and radio engineering” curriculum.

In works [5], [6], the issue of using specialized software resources when studying a higher mathematics course by students was considered. Obviously, students should be encouraged to consciously master mathematical concepts and results, which is the true goal of such a course, and not have it replaced by online calculators, artificial intelligence tools, etc. It is worth noting that mathematics teachers at universities constantly face a problem: students, instead of solving problems independently, use online calculators. Over time, it turns out that some of them can no longer do without such tools - the study of mathematical theory is reduced to nothing. The main conflict that arises in this case is the need, on the one hand, to limit as much as possible the use by students of online calculators and similar programs, including artificial intelligence tools, as such, which are fundamentally incompatible with the goals of studying mathematical disciplines, on the other hand, students must learn the methods of applying specialized mathematical packages, at least on the example of standard problems that are offered to them when studying these disciplines. Is there a way to solve this problem? It is impossible to control whether students cheat. However, with a properly structured computer science (programming) course, usually taught to students starting in the first semester, an interesting possibility arises: the use of software packages such as GNU Octave [7]. GNU Octave is a free, open-source, numerical computing environment that is an analog of MATLAB, providing powerful tools for mathematical analysis, matrix operations, solving complex equations, and data visualization. This and similar packages allow you to combine the study of computer science and mathematics. It should be noted that this is useful, first, for students, for the following reasons:

- using the package develops certain skills, which allow students who have worked with it to more easily move on to programming in the C language, which is similar in syntax, or to using paid packages such as MATLAB;
- the package involves performing quite complex calculations that will be useful, for example, when studying general engineering disciplines.

It should not be assumed that mathematical education is more important for students in the field of Natural Sciences, Mathematics and Statistics. This erroneous attitude has a negative impact on student training across many technical specialties. Let's look at the needs of the G Engineering, Manufacturing and Construction industry (more precisely, the G5 Electronics, Electronic Communications, Instrumentation and Radio Engineering specialty):

1. Differential calculus:
   - signal analysis- Calculating derivatives allows you to determine the rate of change of electrical signals, which is critical for pulse processing in digital systems;
   - modelling electronic circuits- the change in current and voltage in RC and RL circuits is described by differential equations;
   - optimization of device parameters- finding extrema of functions (for example, minimum energy consumption or maximum gain).
2. Integral calculus:
   - calculating signal energy- the integral of the square of a signal over time determines its energy power;
   - spectral analysis- the Fourier transform is based on integrals, allowing you to move from the time domain to the frequency domain;
   - charge accumulation modeling- current integration allows us to determine the stored charge in a capacitor.
3. Linear algebra:

DOI: 10.20535/2523-4455.mea.359022

- solving systems of equations- matrices are used when analyzing complex electronic circuits (for example, using the nodal potential method);
- image and signal processing- transformation, filtering, compression, all of this is based on matrix operations;
- modeling multi-channel systems- the vector-matrix approach allows you to describe the interaction between system elements.

4. Probability theory and statistics:

- noise analysis- modeling random processes in a signal, estimating the mean, variance, correlation;
- sensor data processing- data filtering, measurement reliability assessment, construction of regression models;
- device reliability- statistical failure modeling, component life prediction.

5. Differential equations:

- modeling the dynamics of electronic devices- transistors, generators, filters - their behavior is described by systems of differential equations;
- simulation of transient processes- when turning on/off power, changing load, etc.;
- development of control algorithms- for example, PI controllers in microcontrollers are based on mathematical models.

6. Numerical methods:

- modeling complex processes- when there is no analytical solution, numerical methods are used (Euler's method, Runge-Kutta method);
- calculation of parameters of electronic components- approximate calculation of characteristics that do not have an exact expression;
- automated design- numerical optimization of circuits, simulation of device operation under variable conditions.

And this is just a small list of tasks that modern engineers face. The authors once again emphasize that the market offers many modern software tools, and our goal is only to present possible options for improving the quality of student training. The choice of tools remains with the ultimate teacher, but the problem should be highlighted: it is impossible to train a modern specialist solely with traditional tools. It is also not worth following the conservative path of "separate educational components" at the moment, since modern software tools require an understanding of programming basics and real-world examples of applying mathematical knowledge in one's specialty.

As mentioned above, GNU Octave is a free, open-source program that also offers a large number of extensions and a convenient mobile application. At the same time, some of its features cannot be ignored: compared to the aforementioned MATLAB, it is slower, has a less convenient interface, and offers fewer functions.

Considering the "typical configuration" of the educational process in a technical university, provided that students become familiar with the Octave package at the beginning of studying a programming (computer science) course, which is, of course, not the most common option for constructing such a course, there is an opportunity to integrate it into a higher mathematics course, or rather, to use its corresponding options already when studying sections of such a course that are traditionally considered at the end of the first semester or at the beginning of the second. Examples of such sections include the integral calculus of one variable and the differential calculus of functions of many variables.

III. Computational Methods in Integral Calculus

The GNU Octave software package allows you to work with definite integrals, which may be useful to students not only when studying mathematics, but also for solving practical problems in physics.

First of all, it should be noted that integrals can be divided into those that are "taken" and those that are "not taken" (do not have an antiderivative expressed in elementary functions). Usually, students at technical universities study mathematics on the basis of examples of problems [7] built on integrals that are "taken". Solving "manually" comes down to a simple algorithm: find the antiderivative, calculate the value using the Newton-Leibniz formula. So, the only thing that distinguishes them from standard problems is the method used to find the antiderivative.

When working with Octave, there are two ways to calculate a definite integration: symbolic integration and numerical integration.

In the symbolic integration method (using the int command), the integration formulas/methods are selected automatically based on the integrand. There is no need to identify symbolic integration with the Newton-Leibniz formula; in fact, it can be performed numerically, selecting the most appropriate method for a specific example.

To use this method, you need to connect a symbolic calculation package. If there is only one variable in the expression, the system will determine it; if there are several variables, the integration variable must be

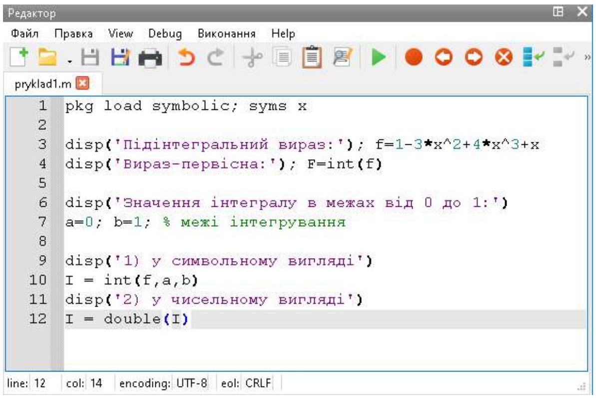
DOI: 10.20535/2523-4455.mea.359022

specified as an argument to the command. If you do not specify limits, Octave returns the original expression without accounting for the integration constant. It is worth considering that any result is given in symbolic form – to get a number, the double command is used (see Fig. 1).

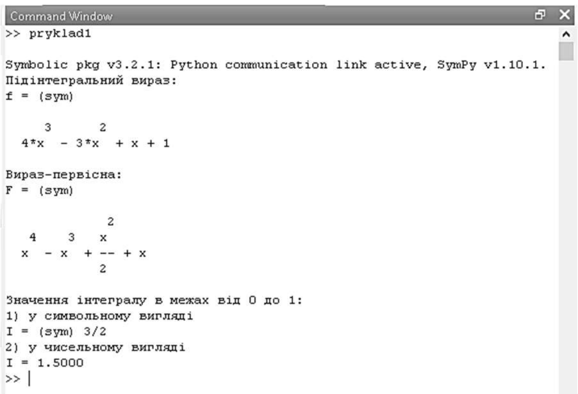
Numerical methods, on the other hand, do not allow finding the initial value; however, these tools allow calculating definite integrals approximately, not limited to those that are "taken". For this, a number of quadrature formulas are available (see, e.g., [8]) using the commands: quad, quadv, quadl, quadgk, and quadcc; the trapezoid method (the trapz command reports the total integral sum, cumtrapz returns an array with intermediate values).

All commands have their own specific syntax, and to use some of them, you need to write the integrand expression as a user function; if necessary, you can get additional data about the calculation progress, such as the number of iterations or the error. Information about this is available to users in the program documentation (Fig. 2).

GNU Octave, like online tools, is effective at finding relatively simple integrals. The difference is that the software package does not “paint” the solution process. The fact is that the integration methods presented in the package are not “tied” to the original (although the int command allows you to find it), but rather, they significantly depend on the “smoothness” of the integrand.

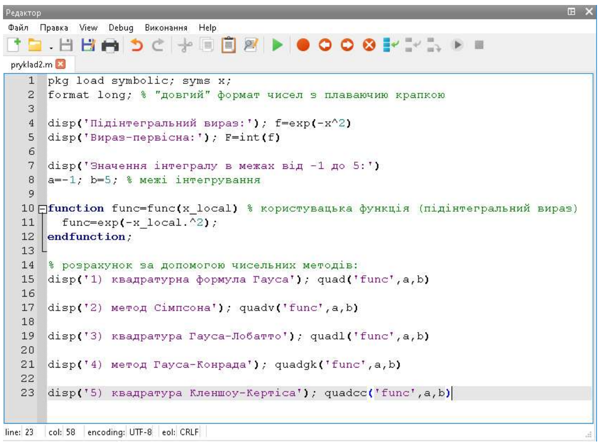


a)

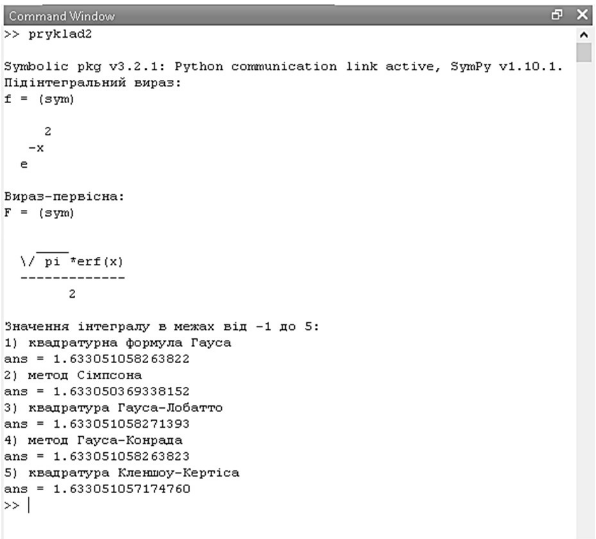


b)

Fig. 1 GNU Octave, symbolic



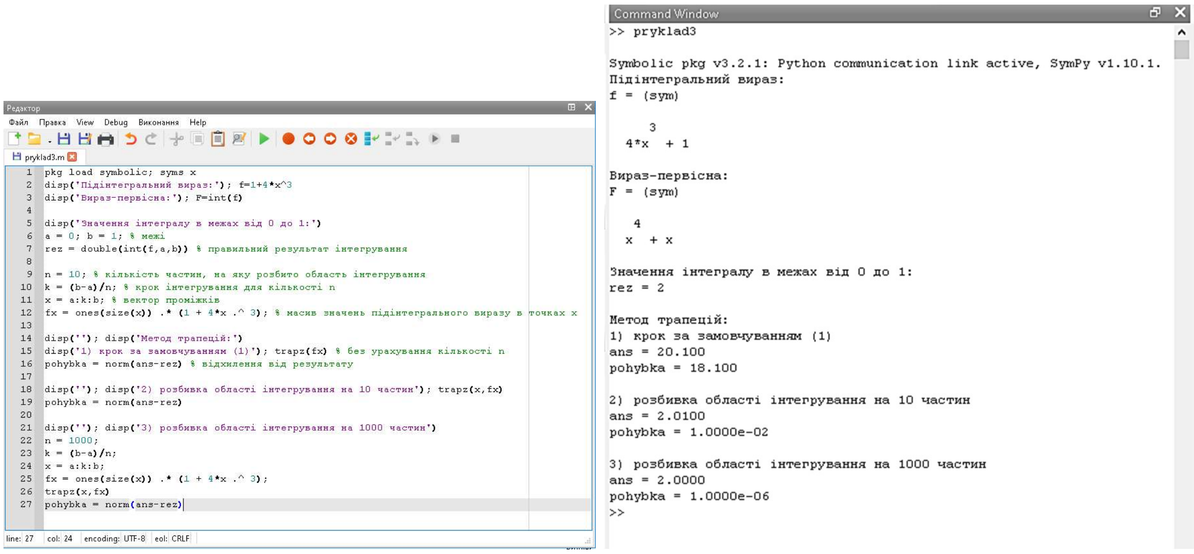
a)



b)

Fig. 2 GNU Octave, example

DOI: 10.20535/2523-4455.mea.359022



a) b)

Fig. 3 Octave, trapezoid method

It is obvious that the accuracy of calculating definite integrals also depends on the number of points considered: the more of them, the more accurate the result. A vivid example of such a dependence is the trapezoid method, where this number is specified by the user (Fig. 3).

It is worth noting that a significant number of integrals encountered in engineering tasks are “not taken”. Many online calculators cannot solve them. This encourages the use of mathematical software packages.

However, all this applies to proper integrals. When there is a need to work with improper integrals [9], the first thing to do is to check the given integral for convergence. Doing this in Octave is quite simple – when applying symbolic integration int to a divergent integral [10], the result will be NaN (not a number) or infinity (Inf).

A convergent definite integral can be calculated in the same way as a proper integral: when using any of the methods considered, some finite numerical value will be obtained, approximately the same for different teams, which will confirm convergence.

However, it should be noted that numerical methods do not always yield correct results for improper integrals (both divergent and convergent). In addition, in terms of improper integrals, the capabilities of the Octave package are somewhat limited: the program functions for Integration does not provide for changes of variables, formulas for integration by parts, calculation of the principal value of the improper integral of the 1st kind from the Cauchy function, etc. But this does not mean at all that formulas of this type cannot be written in code using mathematical operations (limits of expressions, differentiation, integration, etc.), which are represented in Octave by separate commands.

Let us consider two examples of an improper integral of the second kind (with a discontinuity in one of the limits) of the form $\int_a^b \frac{dx}{(x-a)^{\alpha}}$, taking the limits $a=1$ and $b=2$. If $\alpha<1$, then such an integral converges, otherwise it diverges.

The integrand has an infinite discontinuity in the lower limit of integration (Fig. 4).

1) Divergent improper integral, $\alpha=2$ (Fig. 5)

2) Convergent improper integral, $\alpha={}^1\!/_2$ (Fig. 6)

So, it is possible to apply numerical methods to improper integrals, but this does not always guarantee the correct result. For a correct solution, you need to know exactly which command is suitable for calculating a given example; therefore, it is appropriate to rely, first of all, on the results of integration using int.

Thus, GNU Octave provides the ability to work with definite integrals in all their "variety" of types.

IV. Numerical Analysis of Dynamic Systems

Let's consider specific functions that GNU Octave can effectively solve examples from the topic "Differential calculus of functions of many variables", as this allows us to verify the correctness of calculations and deepen our understanding of mathematical concepts through their implementation in program code. For analytical

DOI: 10.20535/2523-4455.mea.359022

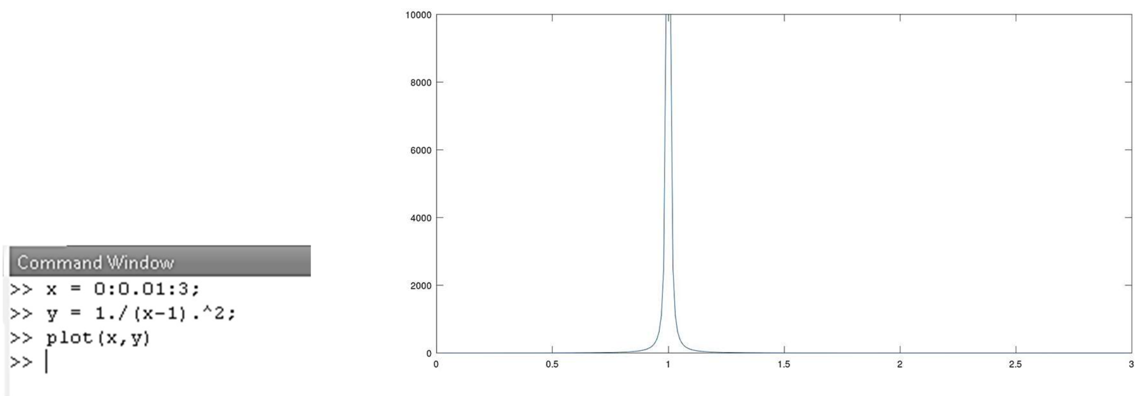
calculation of partial derivatives in GNU Octave, the symbolic package is usually used, which allows you to use the diff(f, x) function for symbolic expressions.

To numerically calculate the gradient of a function given on a grid of values, the gradient function is used. It allows you to evaluate changes in the function's values in both directions. It is worth noting the powerful visualization tools of GNU Octave. When you need to create a graphical image of a surface or analyze the geometric properties of a function, it is advisable to use the meshgrid, surf, or ezsurf functions. Graphical representation significantly simplifies the perception of complex mathematical concepts.

Fundamentally important in the context of the differential calculus of functions of many variables are numerical methods for optimization and extremum finding. GNU Octave provides a robust toolkit for solving such problems, making it indispensable for the educational process. To find the minimum of a function, GNU Octave offers several methods with varying levels of accuracy and complexity.

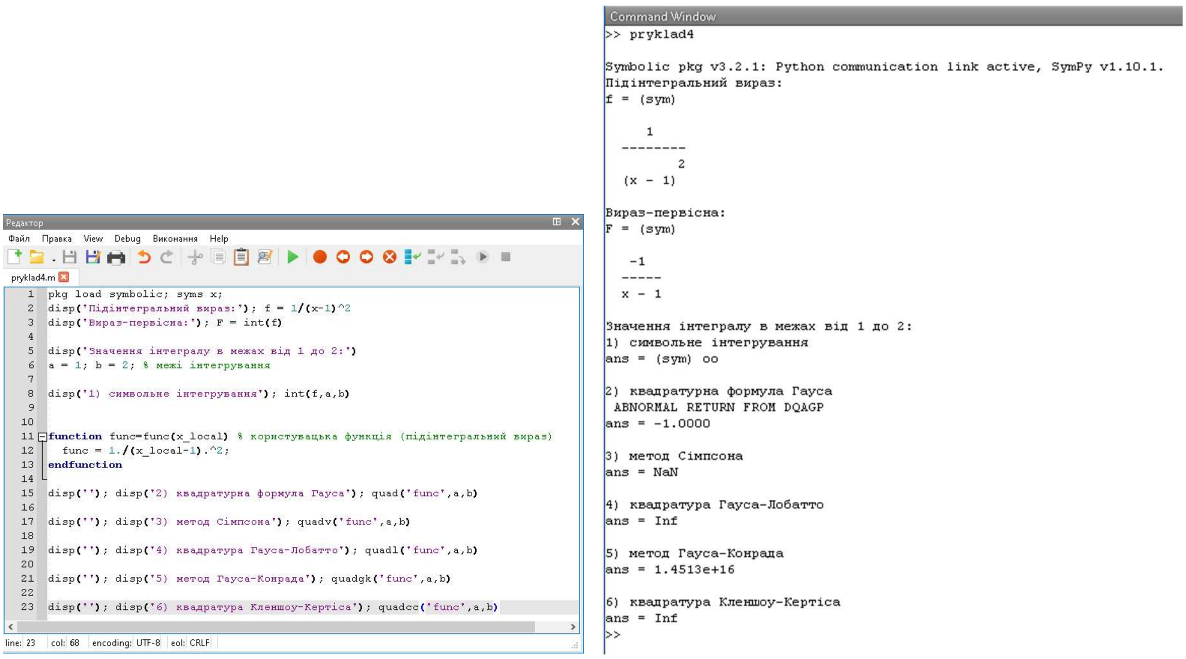
The fminsearch function implements a method without using derivatives (the Nelder-Mead method), which makes it universal for various types of problems.

.



a) b)

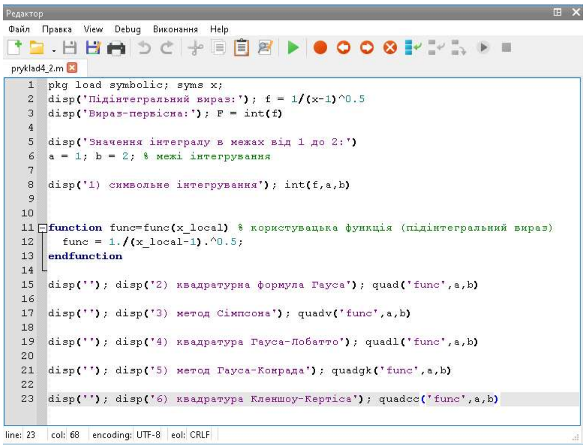
Fig. 4 Octave, integration



a) b)

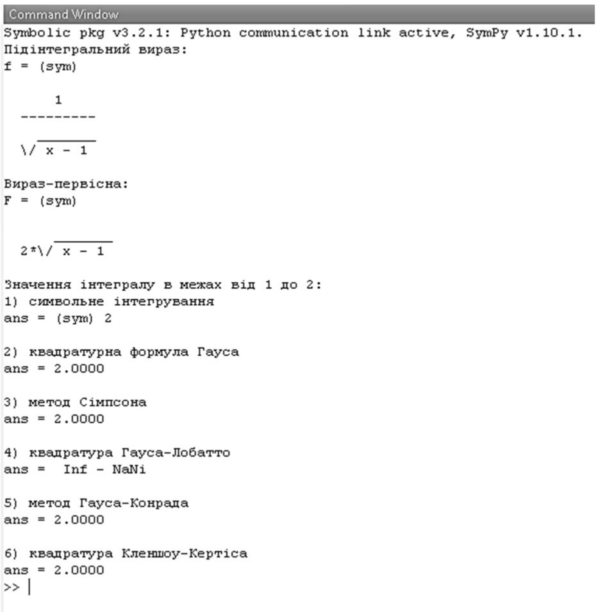
Fig. 5 Octave, integration $\alpha = 2$

DOI: 10.20535/2523-4455.mea.359022



```
pkg load symbolic; syms x;
disp('Підінтегральний вираз:'); f = 1/(x-1)^0.5
disp('Вираз-первісна:'); F = int(f)

disp('Значення інтегралу в межах від 1 до 2:')
a = 1; b = 2; % межі інтегрування

disp('1) символьне інтегрування'); int(f,a,b)


function func=func(x_local) % користувацька функція (підінтегральний вираз)
  func = 1./(x_local-1).^0.5;
endfunction

disp(''); disp('2) квадратурна формула Гауса'); quad('func',a,b)

disp(''); disp('3) метод Сімпсона'); quadv('func',a,b)

disp(''); disp('4) квадратура Гауса-Лобатто'); quadl('func',a,b)

disp(''); disp('5) метод Гауса-Конрада'); quadgk('func',a,b)

disp(''); disp('6) квадратура Кленшоу-Кертіса'); quadcc('func',a,b)
```

a)



```
Symbolic pkg v3.2.1: Python communication link active, SymPy v1.10.1.
Підінтегральний вираз:
f = (sym)

      1
  ---------
    _______
  \/ x - 1

Вираз-первісна:
F = (sym)

      _______
  2*\/ x - 1

Значення інтегралу в межах від 1 до 2:
1) символьне інтегрування
ans = (sym) 2

2) квадратурна формула Гауса
ans = 2.0000

3) метод Сімпсона
ans = 2.0000

4) квадратура Гауса-Лобатто
ans =  Inf - NaNi

5) метод Гауса-Конрада
ans = 2.0000

6) квадратура Кленшоу-Кертіса
ans = 2.0000
>> |
```

b)

Fig. 6 Octave, integration $\alpha = 1/2$

A more accurate result is provided by the fminunc function, which uses derivatives for optimization. This method works best when the function is differentiable, and its derivatives can be easily calculated.

An important feature of the package is that to find the maximum of a function, a transformation of the problem is used – a function sign rotation. That is, instead of finding the maximum of the function f(x), the minimum of the function -f(x) is found using the construction fminsearch(@(x) -f(x)).

Let's consider how Octave handles one of the examples given to students when studying this topic: finding the total differential of the function u at point M.

$$u = \ln\left(x + \frac{y}{2z}\right);\ du\big|_{M_0(1;2;1)} = ?$$

```
Symbolic pkg v3.2.1: Python communication link active, SymPy v1.10.1.
Повний диференціал u:
    dx               dy                 dz*y
 ------- + ------------- - --------------
     y          /     y \        2 /     y \
 x + ---    2*z*|x + ---|   2*z *|x + ---|
    2*z         \    2*z/         \    2*z/
Повний диференціал у точці M0(1,2,1):
 dx   dy   dz
 -- + -- - --
 2    4    2
```

We will also consider an example of studying a function at extrema: $z = x^3 + 8y^2 - 6xy + 5$.

```
Symbolic pkg v3.2.1: Python communication link active, SymPy v1.10.1.
-----------------------------------
Критичні точки (x, y):
  [1.0  0.5]
  [        ]
  [ 0    0 ]
-----------------------------------
Характер критичних точок:
( 1.0000,  0.5000) - локальний мінімум
( 0.0000,  0.0000) - сідлова точка
```

DOI: 10.20535/2523-4455.mea.359022

Octave accurately finds critical points and determines their nature through the Hessian. Using GNU Octave to solve examples and problems in the differential calculus of functions of many variables is appropriate from a pedagogical perspective. This system allows you to automate calculations, visually visualize results, and check solutions, which significantly increases the efficiency of the educational process. When you need to create a graphical representation of a surface or analyze the geometric properties of a function, it is advisable to use the functions meshgrid, surf, or ezsurf. Graphical representation significantly simplifies the perception of complex mathematical concepts.

Considering the use of GNU Octave as the example of studying the section "Differential Equations" of the mathematical analysis course, you can see that GNU Octave solves many different types of differential equations quite well.

In general, the program effectively handles various types of first-order equations, allowing students not only to get the answer, but also to analyze the solution process. GNU Octave also successfully copes with solving higher-order equations, using the important theorem on the equivalence of an nth-order differential equation and a system of n first-order differential equations written in normal form, which makes it a useful tool for studying complex mathematical concepts. At the same time, when it comes to finding general solutions to differential equations, problems arise when solving them symbolically. Having delved into the capabilities of GNU Octave, you can learn about the limits of this program, but they, albeit partially, can be expanded using commands that install additional symbolic packages: pkg load symbolic. However, expanding the capabilities of this free software has its limitations, so there are topics that it cannot cover: differential equations in complete differentials (although the package allows you to classify this type of equations), checking the linear dependence or independence of systems of functions (such tasks require additional macros), selecting partial solutions for a linear inhomogeneous differential equation with constant coefficients and a right-hand side of a special form, the matrix method for solving a system of linear differential equations, and some specific types of equations that require complex symbolic transformations.

Despite this, GNU Octave is not an ordinary calculator; it is much more. Comparing an online differential equation calculator and this software, we can conclude that when using an ordinary calculator, a student does not think about anything, he simply begin to write down the solution, unlike GNU Octave, where to solve even the simplest differential equation, you need to know the special functions of the package and the solution process itself.

Let's consider specific examples:

- $dy/dx = 2^y - 2/(x+y-2)$;
- $y''' - 4y' = 0$, $y(0) = 0$, $y'(0) = 2$, $y''(0) = 4$ (Symbolic solution).

It is necessary to note that none of the online calculators were able to solve Bernoulli's equation normally; we had to use artificial intelligence to compare the answers, but it was noticed that artificial intelligence performed the example not in the general way that is taught in the course of mathematical analysis, but in the most rational way. This leads to the idea that students can additionally use and analyze the methods of solving equations that artificial intelligence offers.

In applying the methods considered in the course of mathematical analysis, students encounter numerous examples of differential equations (DEs) that require not only an analytical but also a numerical approach. GNU Octave, a free mathematical software package, is a powerful tool for this task. It becomes especially effective when studying the numerical solution of the Cauchy problem for ordinary differential equations.

Unlike symbolic solutions, which have limitations, numerical methods (such as Euler, Runge-Kutta, Adams, etc.) are implemented in GNU Octave reliably and efficiently. They allow not only to find approximate values of solutions, but also to analyze the behavior of a function on a given interval by changing the integration step or initial conditions.

A significant advantage of GNU Octave is the built-in ode45 function, which implements the k-th order Runge-Kutta method with automatic step control, making the program very convenient for numerical modeling.

For the numerical solution of the first-order Cauchy problem of the form: $y''' - 4y' = 0$, $y(0) = 0$, $y'(0) = 2$, $y''(0) = 4$ (Numerical solution).

Conclusions

In the modern educational process, it is extremely important not only to give students the correct answer but also to teach them to understand the process that leads to this answer. In this context, symbolic solving, although it provides an accurate analytical solution, often turns into a "black box": the student sees only the formula of the result, without being involved in the process of obtaining it. This leads to superficial mastery of the topic and does not form analytical thinking.

In addition, numerical solutions enable modeling complex real-world processes for which an analytical solution does not exist or is too difficult to obtain. Thus, students do not simply study mathematics as an

DOI: 10.20535/2523-4455.mea.359022

abstraction; they see its practical applications in physics, biology, economics, etc.

Introducing GNU Octave into a mathematical analysis course allows us to resolve the dilemma between the need to limit the use of "ready-made" solutions and the need to teach students to work with modern mathematical tools. This ensures that students consciously master mathematical concepts and results, which is the true goal of mathematical education at a technical university.

Надійшла до редакції 30 серпня 2025 року
Прийнята до друку 18 січня 2026 року
Опублікована 27 квітня 2026 року.

References:

[1] J. Miranda *et al.*, "The core components of education 4.0 in higher education: Three case studies in engineering education," *Computers & Electrical Engineering*, vol. 93, p. 107278, Jul. 2021, **DOI:** 10.1016/j.compeleceng.2021.107278.

[2] S. Frerich *et al.*, Eds., *Engineering Education 4.0*. Cham: Springer International Publishing, 2016, **ISBN:** 978-3-319-46915-7.

[3] S. I. Shankar, "Blended Learning is the future of Education," in *The Asian Conference on Education (ACE2021)*, 2021.

[4] D. Harris, L. Black, P. Hernandez-Martinez, B. Pepin, and J. Williams, "Mathematics and its value for engineering students: what are the implications for teaching?," *Int. J. Math. Educ. Sci. Technol.*, vol. 46, no. 3, pp. 321–336, Apr. 2015, **DOI:** 10.1080/0020739X.2014.979893.

[5] O. V. Bogdanov, Y. P. Butsenko, O. I. Balina, and I. S. Bezklubenko, "Software Support for the Higher Mathematics Course at the Technical University," *Microsystems, Electronics and Acoustics*, vol. 29, no. 2, Aug. 2024, **DOI:** 10.20535/2523-4455.mea.314217.

[6] O. V. Bogdanov, Y. P. Butsenko, O. I. Balina, and I. S. Bezklubenko, "Software Support for the Higher Mathematics Course at the Technical University," *Microsystems, Electronics and Acoustics*, vol. 30, no. 1, Apr. 2025, **DOI:** 10.20535/2523-4455.mea.328241.

[7] E. Yu. Zhelezniakova and L. O. Norik, *Vyshcha matematyka v GNU Octave [Higher Mathematics in GNU Octave]*. Kharkiv, Ukraine: Simon Kuznets Kharkiv National University of Economics, 2024.

[8] I. V. Aleksieieva, V. O. Haidei, A. Yu. Kinybaliuk, N. R. Konovalova, O. P. Trofymchuk, and L. B. Fedorova, *Dyferentsialne ta intehralne chyslennia funktsii odniieii zminnoii: Zbirnyk zavdan do typovoii rozrakhunkovoyi roboty dlia studentiv 1 kursu tekhnichnykh fakultetiv [Differential and integral calculus of functions of one variable: A collection of tasks for typical calculation work for first-year students of technical faculties]*. Kyiv, Ukraine: Politechnika, 2001.

[9] K. M. Holubieva, S. V. Denysov, O. F. Kashpur, D. A. Kliushyn, and A. I. Ryzhenko, *Chyselni metody intehruvannia [Numerical integration methods]*. Kyiv, Ukraine: Lyudmila Publishing House, 2019.

[10] I. V. Aleksieieva, V. O. Haidei, O. O. Dykhovychnyi, and L. B. Fedorova, "Nevlastyvi intehraly [Improper integrals]," in *Matematyka v tekhnichnomu universyteti [Mathematics at the Technical University]*, vol. 3, Kyiv, Ukraine: Igor Sikorsky KPI, 2021, pp. 54–61.

УДК 510.2; 621.3

Програмні засоби супроводження курсу вищої математики у технічному університеті

Частина 3. GNU Octave

DOI: 10.20535/2523-4455.mea.359022

Д. Войціх[f], студент бакалаврату
Д. Грин[f], студент бакалаврату
М. Гринюк[f], студент бакалаврату
О. В. Богданов, доц., канд. техн. наук, 0000-0002-0911-5563
Ю. П. Буценко, доц., канд. техн. наук, 0000-0003-4806-9587
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського» 00syn5v21
Київ, Україна

О. І. Баліна, доц., канд. техн. наук, 0000-0001-6925-0794
І. С. Безклубенко, доц., канд. техн. наук, 0000-0002-9149-4178
Київський національний університет будівництва і архітектури 00syn5v21
Київ, Україна

Анотація—У цій статті досліджується методологічний підхід до модернізації навчальної програми з вищої математики для студентів-інженерів, з особливим акцентом на потреби спеціальності «G5 Електроніка, електронні комунікації, приладобудування та радіотехніка». Дослідження розглядає зростаючу проблему парадигми «чорної скриньки», де студенти покладаються на автоматизовані онлайн-калькулятори та інструменти штучного інтелекту, не розуміючи основної математичної логіки чи алгоритмічних структур. Щоб протидіяти цій тенденції, автори пропонують інтеграцію GNU Octave, обчислювального середовища з відкритим кодом, як основного інструменту для подолання розриву між абстрактною теорією та професійною інженерною практикою.

У статті демонструється практична реалізація GNU Octave у трьох критичних математичних областях: інтегральне числення, з акцентом на символічних та числових методах для застосувань обробки сигналів; багатовимірні функції, зосереджені на методах 3D-візуалізації (з використанням meshgrid та surf) для моделювання просторових фізичних полів та розподілів потенціалу; диференціальні рівняння, з використанням числових розв'язувачів, таких як ode45, для моделювання перехідних процесів в електронних схемах.

Результати свідчать про те, що перехід від спрощених автоматизованих інструментів до обчислювального моделювання на основі сценаріїв сприяє свідомому оволодінню математичними поняттями. Такий підхід гарантує, що майбутні інженери розвинуть необхідні аналітичні та програмні компетенції, необхідні для оптимізації складних систем та професійного моделювання в галузі сучасної електроніки.

Ключові слова — Інженерна освіта; Викладання інженерії; Електронна інженерія; GNU Octave; Чисельний аналіз; Математичне моделювання; Аналіз схем; Візуалізація; Освітні технології.