ИНФОРМАТИКА

COMPUTER SCIENCE

IRSTI 20.53.19

10.51889/2959-5894.2024.88.4.014

G. Alkhanova[1*], S. Zhuzbayev[2], A. Baikonys[3], Ye. Pernebayev[4]

[1]Alikhan Bokeikhan University, Semey, Kazakhstan,
[2]L.N.Gumilyov Eurasian National University, Astana, Kazakhstan,
[3]Abai Kazakh National Pedagogical University, Almaty, Kazakhstan,
[4]South Kazakhstan University named after M. Auezov, Shymkent, Kazakhstan
[*]*e-mail: gulnur_alhanova@mail.ru*

MATHEMATICAL MODELING AND DATA PROCESSING IN PYTHON

Abstract

The advent of mathematical modeling has significantly impacted various aspects of people's lives and has contributed to advancements in civilization and the exploration of new frontiers. Scientific modeling is a fundamental component of modern research, offering a method for both qualitative and quantitative representation of processes, phenomena, or objects through numerical models. These models are developed using mathematical tools that effectively capture the essence of the real-world processes, phenomena, or objects being studied. A thorough examination of the history and application scope of mathematical modeling reveals its profound influence in simplifying human life and addressing pressing challenges faced by humanity. This includes an exploration of both the benefits and limitations associated with scientific modeling. Additionally, this study delves into the utilization of computer programs, particularly Python, for simulating physical phenomena. This work encompasses various aspects of scientific modeling, ranging from its historical origins to the classification of numerical modeling techniques and models. Practical experiments on modeling free harmonic motions, such as second-order thermal conductivity, using Python's scientific packages with the capability to manipulate input data are also presented. Through these endeavors, valuable insights into the intricacies of mathematical modeling and its practical applications are gained.

Keywords: python, computer mathematics programs, mathematical apparatus, mathematical model, mathematical modeling, heat equation, sweep method, fractional steps method, data processing.

Г. Алханова[1], С. Жүзбаев[2], А. Байконыс[3], Е. Пернебаев[4]
[1]Alikhan Bokeikhan University, Семей қ., Қазақстан,
[2]Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан,
[3]Абай атындағы Қазақ ұлттық педагогикалық университеті, Алматы қ., Қазақстан,
[4]М.Әуезов атындағы Оңтүстік Қазақстан университеті, Шымкент қ., Қазақстан,

PYTHON ТІЛІНДЕ МАТЕМАТИКАЛЫҚ МОДЕЛЬДЕУ ЖӘНЕ ДЕРЕКТЕРДІ ӨҢДЕУ

Аңдатпа

Математикалық модельдеудің пайда болуы адамдар өмірінің әртүрлі аспектілеріне айтарлықтай әсер етті және өркениеттің дамуына және жаңа шекаралардың дамуына үлес қосты. Ғылыми модельдеу сандық модельдер арқылы процестерді, құбылыстарды немесе объектілерді сапалы және сандық бейнелеу әдісін ұсынатын заманауи зерттеулердің негізгі құрамдас бөлігі болып табылады. Бұл модельдер зерттелетін процестердің, құбылыстардың немесе нақты әлем объектілерінің мәнін тиімді көрсететін математикалық құралдарды қолдану арқылы жасалады. Математикалық модельдеудің тарихы мен қолданылу аясын мұқият зерттеу оның адам өмірін жеңілдетуге және адамзат алдында тұрған өзекті мәселелерді шешуге терең әсерін көрсетеді. Бұл ғылыми модельдеуге байланысты артықшылықтар мен шектеулерді зерттеуді қамтиды. Сонымен қатар, бұл зерттеу физикалық құбылыстарды модельдеу үшін компьютерлік программаларды, атап айтқанда Python -. қолдануға бағытталған. Бұл жұмыс ғылыми модельдеудің әртүрлі аспектілерін қамтиды, оның тарихи бастауларынан бастап сандық модельдеу әдістері мен модельдерін жіктеуге дейін. Кірістерді манипуляциялау мүмкіндігі бар Python ғылыми пакеттерін қолдана отырып, екінші ретті жылу өткізгіштік сияқты еркін гармоникалық қозғалыстарды модельдеуге арналған практикалық эксперименттер де ұсынылған. Осы күш-жігердің арқасында математикалық модельдеудің қыр-сыры және оның практикалық қолданылуы туралы құнды ақпарат алуға болады.

Түйін сөздер: python, компьютерлік математикалық программалар, математикалық аппарат, математикалық модель, математикалық модельдеу, жылу өткізгіштік теңдеуі, сыпыру әдісі, бөлшек қадам әдісі, деректерді өңдеу.

Г. Алханова[1], С. Жүзбаев[2], А. Байконыс[3], Е. Пернебаев[4]
[1]Alikhan Bokeikhan University, г. Семей, Казахстан,
[2]Евразийский национальный университет имени Л.Н. Гумилева, г. Астана, Казахстан,
[3]Казахский национальный педагогический университет имени Абая, г. Алматы, Казахстан,
[4]Южно-Казахстанский университет им. М.Ауэзова, г. Шымкент, Казахстан,

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ОБРАБОТКА ДАННЫХ НА ЯЗЫКЕ PYTHON

Аннотация

Появление математического моделирования существенно повлияло на различные аспекты жизни людей и внесло свой вклад в развитие цивилизации и освоение новых рубежей. Научное моделирование является фундаментальным компонентом современных исследований, предлагая метод как качественного, так и количественного представления процессов, явлений или объектов с помощью численных моделей. Эти модели разрабатываются с использованием математических инструментов, которые эффективно отражают суть изучаемых процессов, явлений или объектов реального мира. Тщательное изучение истории и сферы применения математического моделирования показывает его глубокое влияние на упрощение жизни человека и решение насущных проблем, с которыми сталкивается человечество. Это включает в себя изучение как преимуществ, так и ограничений, связанных с научным моделированием. Кроме того, это исследование посвящено использованию компьютерных программ, в частности Python, для моделирования физических явлений. Эта работа охватывает различные аспекты научного моделирования, начиная с его исторических истоков и заканчивая классификацией методов и моделей численного моделирования. Также представлены практические эксперименты по моделированию свободных гармонических движений, таких как теплопроводность второго порядка, с использованием научных пакетов Python с возможностью манипулирования входными данными. Благодаря этим усилиям можно получить ценную информацию о тонкостях математического моделирования и его практическом применении.

Ключевые слова: python, компьютерные математические программы, математический аппарат, математическая модель, математическое моделирование, уравнение теплопроводности, метод развертки, метод дробных шагов, обработка данных.

Main provisions

In this study, two numerical methods - specifically the tridiagonal matrix method (a variant of the sweep method) and the fractional steps method - were evaluated for their efficiency in solving a heat equation represented by a partial differential equation. The research demonstrated that the fractional steps method required significantly fewer iterations (98 iterations) compared to the tridiagonal matrix method (2571 iterations) to achieve convergence. Therefore, it concluded that the fractional steps method is faster and more efficient, making it a preferable choice for numerical simulations, especially when computational efficiency is a priority. Additionally, the study calls for further refinement and integration of advanced techniques, such as machine learning, with the fractional steps method to improve its performance.

Introduction

The emergence of mathematical modeling during the 20th century was a significant discovery that revolutionized various fields. The understanding of mathematical modeling began to take shape in the late 19th and early 20th centuries, with the works of mathematicians R. Fréchet and D. Hilbert. They introduced new perspectives on proximity in mathematics, such as metric and Hilbert spaces, which laid the foundations for modern mathematical modeling.

These developments led to the formation of new methods in computational mathematics and provided the necessary theoretical groundwork for mathematical modeling. One of the key contributions was the concept of integral identities in mathematical physics, as well as the finite element approach proposed by R. Courant. The finite element method became the basis for variational and projection difference methods used to solve problems in mathematical physics.

Russian scientists A.A. Samarsky and O.M. Belotserkovsky played a significant role in advancing the idea of mathematical modeling. Their contributions helped shape the field and further enhance the effectiveness of mathematical models.

Mathematical modeling has had a profound impact on civilization, contributing to the achievements of various disciplines. It has played a fundamental role in the revolution of physics in the 19th and 20th centuries, allowing for a deeper understanding of natural phenomena and facilitating technological advancements. It is important to note that mathematical modeling has made a huge contribution to the achievements of civilization, as well as the revolution in physics in the 19th and 20th centuries [1].

There are instances where possessing an object is feasible, yet its utilization could incur significant expenses or even result in grave calamities. In such scenarios, the researcher's objective is to formulate a model of the original object, thus foreseeing the characteristics and conduct of the object during its application.

The development of an accurate model necessitates a profound understanding of the object slated for modeling. Occasionally, it is contended that a mathematician devoid of familiarity with the object in question can create a model, as can a specialist well-versed in the object but lacking mathematical comprehension. However, it remains crucial to recognize that proficiency in mathematical modeling demands expertise not solely in mathematical models but also in the object being modeled. Moreover, it is important to remember that to achieve success in mathematical modeling, you need to have knowledge not Mathematical modeling encompasses the process of devising and analyzing mathematical models that encapsulate real-world processes and phenomena using mathematical modeling programs or packages. Furthermore, mathematical modeling constitutes an indispensable component of scientific and technological advancement [2].

Contemporary mathematics boasts an extensive array of powerful research tools. When constructing a model, the pertinent parameters and details of the subject under examination are incorporated, which some believe contain the requisite information about the object, while others view them as facilitating mathematical formalization. Understanding the method of mathematical modeling is imperative to comprehend the modeling process. But it is also necessary to understand what a mathematical modeling method is in order to understand how modeling is carried out [3].

Mathematical model that reflects knowledge in the proposed field of software [4].

Related to the field of engineering knowledge or various sections of artificial intelligence as a scientific discipline [5].

A mathematical model is a reflection of expertise in the respective domain of software. It pertains to the realm of engineering knowledge or various branches of artificial intelligence as a scientific discipline. Emphasis is placed on the recommendation to conduct modeling not solely through laborious programming efforts but rather by transforming modern insights into a format conducive to human comprehension.

It is emphasized that it is advisable to carry out not due to time-consuming programming, but due to the introduction of modern data of revelation into a form convenient for a person [6].

Research methodology

Materials and research methodologies: Various methods can be employed for mathematical modeling of physical phenomena, such as the sweep method, explicit method, fractional step method, Jacobi method, Gauss-Seidel method, among others. In this article, we will delve into two methods for solving a single physical equation and compare their efficacy. Gauss-Seidel method, and so on. In the same article we will touch on two methods for one physical equation and compare them [7].

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{1}$$

where $u(x, y, t)$ – function, a ∂ – partial differential.

Tridiagonal matrix method:

Tridiagonal matrix method: This method is a variant of the sequential elimination method for solving unknowns. The sweep method, a special case of the Gauss method, is utilized to solve systems of linear equations represented by Ax = B, where A is a tridiagonal matrix. A tridiagonal matrix is characterized by zeros in all positions except the main diagonal and its adjacent elements. The sweep method comprises two stages: forward sweep and backward sweep. During the first stage, the running coefficients are determined, while the unknown variables x are computed during the second stage. At each stage, the latest calculated values of the values are used [8].

Let's discretize the aforementioned equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} \tag{2}$$

where i –is a sweep across the plate and $O((\Delta x^2), (\Delta t))$ – approximation error.

Python is one of the most widely used programming languages, and according to the PYPL (popularity of programming languages) index, it is the most popular in the world [9]. At the same time, Python has all the functionality of other languages and even more [10]. Derivation of this formula in program code. Below is the output of the formula in the program code in Figure 1.

```
while (maxi>eps):
    for j in range(n):
        newp[0][j]=1
        newp[n-1][j]=1
    for i in range(n):
        newp[i][0]=0
        newp[i][n-1]=0
    for i in range(1,n-1):
        for j in range(1,n-1):
            newp[i][j]=oldp[i][j]+dt*((oldp[i+1][j]-2*oldp[i][j]+oldp[i-1][j])/(dx*2)+(oldp[i][j+1]-2*oldp[i][j]+oldp[i][j-1])/(dy*2))
    maxi=0
```

Figure 1. Re-check data by the loop

In this context, a while loop is employed to re-evaluate the maximum value data with an error threshold, typically set at 0.00001 for our purposes.

Figure 2 following this, a for loop is utilized to track the final iteration, determining the total number of completed iterations.

```
for i in range(1,n):
    for j in range(1,n):
        if (maxi<abs(newp[i][j]-oldp[i][j])):
            maxi=abs(newp[i][j]-oldp[i][j])
for i in range(n):
    for j in range(n):
        oldp[i][j]=newp[i][j]
iter+=1
```

Figure 2. Count of iterations

Transitioning to the second method, we delve into the fractional steps approach. In this method, also known as splitting schemes, the progression to the subsequent time layer is fragmented into multiple intermediate stages. At each of these individual stages, there's no necessity to ensure both approximation and stability. However, the collective result of these stages yields a full-step approximation, enabling the construction of a convergent and cost-effective scheme. Presently, fractional steps methodology stands as an indispensable component in formulating frameworks to address intricate multidimensional challenges in mathematical physics.

The following steps will be used for our equation:

Step 1.

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^{n}}{\Delta t} = \frac{1}{2}\left(\Lambda_1 u^{n+\frac{1}{2}} + \Lambda_1 u^{n}\right) + \Lambda_2 u^{n} \quad (3)$$

Step 2.

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+1/2}}{\Delta t} = \frac{1}{2}\left(\Lambda_2 u^{n+1} - \Lambda_2 u^{n}\right) \quad (4)$$

where operators Λ_1и Λ_1 are equal to:

$$\Lambda_1 = \frac{\partial^2}{\partial x^2} \quad (5)$$

$$\Lambda_2 = \frac{\partial^2}{\partial y^2} \quad (6)$$

Figure 3 these formulas in Python are specified as follows.

```
while (maxx>eps):
    for j in range(1,n):
        betta1[1]=newp[0][j]
        alpha1[1]=0
    for j in range(1,n):
        for i in range(1,n):
            d1=oldp[j][i]/dt+(oldp[j][i+1]-2*oldp[j][i]+oldp[j][i-1])/(2*(dx*2))+(oldp[j+1][i]-2*oldp[j][i]+oldp[j-1][i])/(dy*2)
        for i in range(1,n):
            alpha1[i+1]=-a1/(b1+c1*alpha1[i])
            betta1[i+1]=(d1-c1*betta1[i])/(b1+c1*alpha1[i])
        for i in range(n-1,0,-1):
            newp[i][j]=newp[i+1][j]*alpha1[i+1]+betta1[i+1]
    for j in range(1,n):
        alpha2[1]=0
        betta2[1]=0
    for j in range(1,n):
        for i in range(1,n):
                d2=newp[i][j]/dt+(oldp[j+1][i]-2*oldp[j][i]+oldp[j-1][i])/(2*(dy**2))
        for i in range(1,n):
                alpha2[i+1]=-a2/(b2+c2*alpha2[i])
                betta2[i+1]=(d2-c2*betta2[i])/(b2+c2*alpha2[i])
        for i in range(n-1,0,-1):
            newp[j][i]=newp[j][i+1]*alpha2[i+1]+betta2[i+1]
        maxx=0
```

Figure 3. Re-check data by steps

This employs a loop to iterate over the variables of the x and y axes in two steps. Figure 4 the number of iterations can be tracked similarly to the first method, using a for loop to monitor the progress.

```
for i in range(0,n+1):
    for j in range(0,n+1):
        if maxx<abs(newp[i][j]-oldp[i][j]):
            maxx=abs(newp[i][j]-oldp[i][j])
for i in range(0,n+1):
    for j in range(0,n+1):
        oldp[i][j]=newp[i][j]
iterr=iterr+1
```

Figure 4. Count of iterations

Results of the study

Figure 5 the outcome of applying the tridiagonal matrix method to the heat equation yields the following graph and associated data:

```
2571
[[0.         1.         1.         ... 1.         1.         0.        ]
 [0.         0.49998006 0.69761004 ... 0.69761004 0.49998006 0.        ]
 [0.         0.30231037 0.49992056 ... 0.49992056 0.30231037 0.        ]
 ...
 [0.         0.30231037 0.49992056 ... 0.49992056 0.30231037 0.        ]
 [0.         0.49998006 0.69761004 ... 0.69761004 0.49998006 0.        ]
 [0.         1.         1.         ... 1.         1.         0.        ]]
```

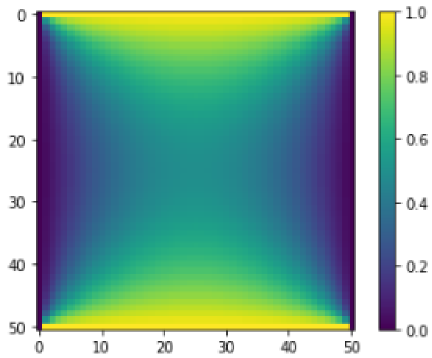

Figure 5. Results from tridiagonal matrix method

Figure 6 we have generated a graph illustrating the thermal conductivity distribution across the plate, along with a tridiagonal data matrix. Solving this problem required 2571 iteration.

```
98
0.0      0.0      1.0
0.0      0.1111111111111111      0.0
0.0      0.2222222222222222      0.0
0.0      0.3333333333333333      0.0
0.0      0.4444444444444444      0.0
0.0      0.5555555555555556      0.0
0.0      0.6666666666666666      0.0
0.0      0.7777777777777777      0.0
0.0      0.8888888888888888      0.0
0.0      1.0      0.0
0.1111111111111111      0.0      1.0
0.1111111111111111      0.1111111111111111      0.03072442962858342
0.1111111111111111      0.2222222222222222      0.03187715019905951
0.1111111111111111      0.3333333333333333      0.03192039801948008
0.1111111111111111      0.4444444444444444      0.03192202050763673
0.1111111111111111      0.5555555555555556      0.031922079099496146
0.1111111111111111      0.6666666666666666      0.03192202050763672
0.1111111111111111      0.7777777777777777      0.03192039801948007
```

Figure 6. Results from fractional step method

Figure 7 applications of the fractional step method.

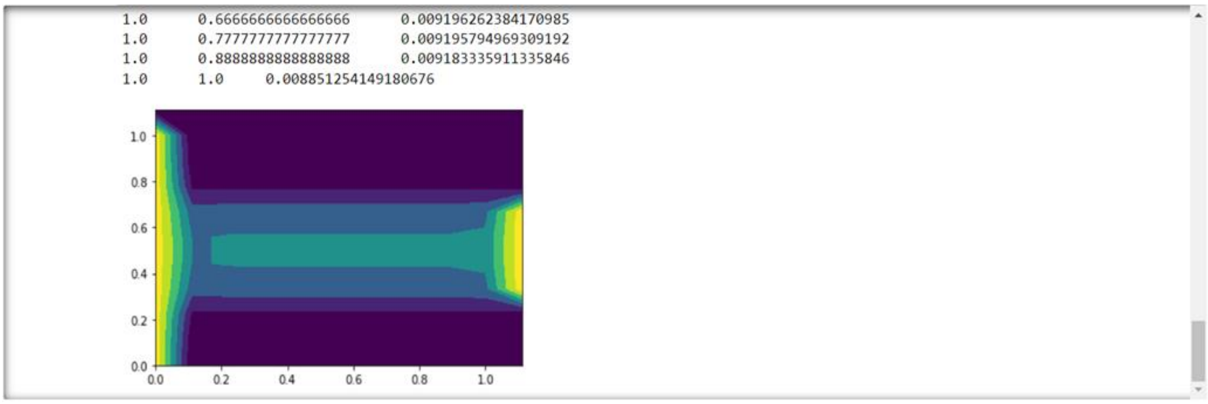

Figure 7. Results from fractional step method

Based on the outcomes derived from this approach, we observe the dispersion of heat across the plate, with data indicating that as time progresses, the heat dispersion also intensifies. Upon segmenting the task into steps, we achieve the desired outcome within 98 iterations.

Discussion

The results of this study have significant implications for computational efficiency in numerical simulations across various domains, including engineering and physics. The ability of the fractional steps method to converge in significantly fewer iterations suggests that it can effectively streamline the processes involved in simulating physical phenomena. This finding aligns with previous research that has emphasized the importance of optimizing numerical methods for enhanced performance, particularly as the complexity of problems increases. The observed superiority of the fractional steps method mirrors trends in the field of computational science that favor methods capable of reducing computational load without sacrificing accuracy. By adopting the fractional steps method, researchers can allocate their computational resources more effectively, addressing larger and more intricate problems in a reasonable timeframe. Future research prospects include exploring the potential of hybrid approaches that combine classical numerical methods with modern techniques, such as machine learning, to further enhance the efficiency and adaptability of numerical simulations. This integration could lead to even faster convergence rates and improved predictive capabilities, thus setting the stage for innovative solutions to complex scientific challenges. Overall, this study contributes valuable insights into the ongoing discourse on numerical methods' optimization and their critical role in advancing scientific research.

Conclusion

In conclusion, based on these findings, it can be inferred that the fractional steps method proves significantly faster and more efficient compared to the sweep method. This is evident from the fact that the number of iterations required for the former is 26 times fewer, despite the potentially larger code involved. The goal of the study was to compare the efficiency of the fractional steps method with the sweep method in numerical simulations. To achieve this, the researcher conducted practical experiments using both methods and analyzed their performance in terms of computational speed and efficiency. The methods involved developing mathematical models, implementing algorithms, and running simulations using appropriate computational tools. The results of the study showed that the fractional steps method outperformed the sweep method in terms of computational efficiency. Specifically, the fractional steps method required significantly fewer iterations to converge compared to the sweep method, despite potentially involving larger code. This indicates that the fractional steps method is faster and more efficient for numerical simulations of the physical phenomena studied. Based on these findings, it can be concluded that the fractional steps method is a preferable choice for numerical simulations when computational efficiency is a priority. Its ability to achieve convergence with fewer iterations can lead to significant time savings in computational tasks.

Additionally, the study highlights the importance of considering different numerical methods and their implications for computational performance in scientific modeling. The findings of this study have implications for various fields where numerical simulations are employed, such as engineering, physics, and computational science. The implementation of the fractional steps method can lead to faster and more efficient simulations, enabling researchers and practitioners to tackle larger and more complex problems within a reasonable computational time frame. Furthermore, future research could explore optimizations and refinements of the fractional steps method to further enhance its performance and applicability in real-world scenarios.

Future work could also investigate the integration of machine learning techniques with the fractional steps method to enhance model predictions and convergence rates. Overall, these advancements could pave the way for more innovative approaches in scientific computing, ensuring that researchers can keep pace with the growing demands for computational power and efficiency.

References

[1] History of mathematical modeling and technology of computational experiment [Electronic resource]. URL: https://www.computer-museum.ru/articles/galglory_ru/1466/ (accessed September 23, 2023).

[2] Mathematical modeling. [Electronic resource]. URL: https://works.doklad.ru/view/NTZgPljhPy8.html (accessed September 20, 2023).

[3] Mathematical modeling. The concept of model and simulation. [Electronic resource]. URL: http://www.pedsovet.info/info/pages/referats/info_00002.htm (accessed September 22, 2023).

[4] Alkhanova, G., Zhuzbayev, S., Syrkin, I., Kurmangaliyeva, N. Model of an automated educational and methodological complex based on a semantic network. Journal of Theoretical and Applied Information Technology, 2021, 99(24), pages 5713–5723. URL: https://www.jatit.org/volumes/Vol99No23/12Vol99No23.pdf

[5] Alkhanova, G., Zhuzbayev, S., Syrkin, I., Kurmangaliyeva, N. Intelligent Mobile Models and Their Application in the Educational Process. International Journal of Interactive Mobile Technologies, 2022, 16(21), pages 201–217. URL: https://online-journals.org/index.php/i-jim/article/view/36069/12251

[6] Alkhanova, G., Stenin, D., Zhuzbaev, S. The semantic network as a promising information platform in the mining industry. E3S Web of Conferences, 2019, 105, 03015. URL: https://www.e3s-conferences.org/articles/e3sconf/pdf/2019/31/e3sconf_iims18_03015.pdf

[7] Alexander Semenov, M.N. Semenova, Yuriy Vladimirovich Bebikhov, Ilya Yakushev. Robotics, Machinery and Engineering Technology for Precision Agriculture. Moscow, 2022, page 438. URL: https://link.springer.com/chapter/10.1007/978-981-16-3844-2_40

[8] Lukin V.V. Mathematical modeling of channeled radiation accelerated emissions in astrophysical systems. Moscow, 2019, page 36.

[9] Paul and Harvey Deitel Python: Artificial Intelligence, Big Data and Cloud Computing [1st ed.] 9785446114320 page 52.

[10] Fletcher Heisler, David Amos, Dan Bader, Joanna Jablonski Copyright. Python Basics: A Practical Introduction to Python 3 Real Python, 2020, page 21.