DISCRETE AND CONTINUOUS MODELS AND APPLIED COMPUTATIONAL SCIENCE

Volume 33 Number 1 (2025)

Founded in 1993

Founder: PEOPLES' FRIENDSHIP UNIVERSITY OF RUSSIA NAMED AFTER PATRICE LUMUMBA

DOI: 10.22363/2658-4670-2025-33-1

Edition registered by the Federal Service for Supervision of Communications, Information Technology and Mass Media
Registration Certificate: ПИ № ФС 77-76317, 19.07.2019

ISSN 2658-7149 (Online); 2658-4670 (Print)
4 issues per year.
Language: English.

Publisher

Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University).

Indexed by

- Scopus (https://www.scopus.com),
- Ulrich's Periodicals Directory (http://www.ulrichsweb.com),
- Directory of Open Access Journals (DOAJ) (https://doaj.org),
- Russian Index of Science Citation (https://elibrary.ru),
- CyberLeninka (https://cyberleninka.ru).

Aim and Scope

Discrete and Continuous Models and Applied Computational Science arose in 2019 as a continuation of RUDN Journal of Mathematics, Information Sciences and Physics. RUDN Journal of Mathematics, Information Sciences and Physics arose in 2006 as a merger and continuation of the series "Physics", "Mathematics", "Applied Mathematics and Computer Science", "Applied Mathematics and Computer Mathematics".

Discussed issues affecting modern problems of physics, mathematics, queuing theory, the Teletraffic theory, computer science, software and databases development.

It's an international journal regarding both the editorial board and contributing authors as well as research and topics of publications. Its authors are leading researchers possessing PhD and PhDr degrees, and PhD and MA students from Russia and abroad. Articles are indexed in the Russian and foreign databases. Each paper is reviewed by at least two reviewers, the composition of which includes PhDs, are well known in their circles. Author's part of the magazine includes both young scientists, graduate students and talented students, who publish their works, and famous giants of world science.

The Journal is published in accordance with the policies of COPE (Committee on Publication Ethics). The editors are open to thematic issue initiatives with guest editors. Further information regarding notes for contributors, subscription, and back volumes is available at http://journals.rudn.ru/miph

E-mail: miphj@rudn.ru, dcm@sci.pfu.edu.ru

Editorial board

Editor-in-Chief

Vice Editors-in-Chief

Members of the editorial board

Computer Design: *Anna V. Korolkova, Dmitry S. Kulyabov*
English Text Editors: *Nikolay E. Nikolaev, Ivan S. Zaryadov, Konstantin P. Lovetskiy*

Address of editorial board:
3 Ordzhonikidze St, 115419 Moscow, Russia
+7 (495) 955-07-16, e-mail: publishing@rudn.ru

Editorial office:
+7 (495) 952-02-50, miphj@rudn.ru, dcm@rudn.su,
site: http://journals.rudn.ru/miph

Paper size 70×108/16. Offset paper. Offset printing. Typeface “Adobe Source”.
Conventional printed sheet 8.92. Printing run 500 copies. Open price. The order 1714.
PEOPLES’ FRIENDSHIP UNIVERSITY OF RUSSIA NAMED AFTER PATRICE LUMUMBA
6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

Printed at RUDN Publishing House:
3 Ordzhonikidze St, Moscow, 115419, Russian Federation,
+7 (495) 955-08-61; e-mail: publishing@rudn.ru

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1)
http://journals.rudn.ru/miph

Contents

Editorial

Computer science

Modeling and Simulation

Letters

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 5–9
http://journals.rudn.ru/miph

Editorial

DOI: 10.22363/2658-4670-2025-33-1-5-9

EDN: DJEBHI

Abstract structure

Dmitry S. Kulyabov[1, 2], Leonid A. Sevastianov[1, 2]

[1] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[2] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

Abstract. We describe the general requirements for the abstract of a scientific article. We recommend to use the structured abstract approach

Key words and phrases: abstract, structured abstract, research article

For citation: Kulyabov, D. S., Sevastianov, L. A. Abstract structure. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 5–9. doi: 10.22363/2658-4670-2025-33-1-5-9. edn: DJEBHI (2025).

1. Annotation assignment

- The abstract summarizes the main results of the research without general phrases.
- The abstract should reflect the content of the article.
- The abstract should be defining rather than descriptive.
- The abstract should provide facts, not tell you what the article is about.
- The abstract should convey information, not promise it.
- The abstract should be self-contained, that is, complete in itself.

2. Annotation requirements

- The abstract is typed in one paragraph.
- The abstract should be typed without a red line.
- The abstract should be self-contained.
- The abstract should not contain footnotes.
- The abstract should not contain references.
- The abstract should be about 5% of the length of the article, but no more than 500 words.
- The minimum length of an abstract is–usually 150 words.
- The optimal length is 200–250 words.
- The abstract is usually followed by keywords.
- References to the list of references in the abstract is not allowed.
- Avoid formulas, especially complex formulas, as this often leads to incorrect representation of the article in various databases and citation systems.
- Do not use long compound sentences, especially those with ambiguous interpretations.
- Superfluous introductory phrases should be avoided.
- Historical references should not be included in the abstract.
- Abstracts in Russian and English should not differ significantly in content.

3. What the abstract should not contain

- Abbreviations and acronyms only if they are not commonly used or are explained.
- Formulas, if they can be avoided.
- References to tables and figures.
- Citations and references to literature.
- Any information or conclusions that are not in the article.
- General statements.
- Complex, cumbersome, verbose sentences.

4. Structured abstract

It is recommended that the annotation be organized according to the following structure [1, 2].

- Background—what is the reason for this paper.
- Purpose—what was the author’s purpose in conducting the study.
- Method—exactly how the research was conducted, what methods were used, what tools were used.
- Results—what the results of the study were.
- Conclusions—the significance or application of the findings.

5. English abstract

5.1. Which tenses to use

- The Present and Past Tenses are commonly used.
- The Present Tense takes precedence when writing an abstract in English.
- The abstract should not look like a simple statement of the results obtained, but should give an idea of the progress of scientific work.
- Statements or common knowledge should be written in the Present Tense.
- For results or research data that preceded the present, use the Past Tense.
- If the sentence is about your current research, you should use the Present Tense.
- In conclusions and explanations, use the Present Tense.
- If the sentence is about experiments or results you observed during your research, use the Past Tense.

5.2. What pronouns to use

In Russian scientific literature, it is not recommended to use the first person pronouns. It is considered that their use in scientific papers looks too egotistical. Therefore, they try to depersonalize the work as much as possible by using impersonal sentences or suffering constructions. In English-speaking scientific circles, until recently, a similar situation was observed. Recently, however, when writing the abstract of an English-language article, it is acceptable (and even recommended) to use the first person pronoun — *I*, *we*.

5.3. Template phrases for writing an abstract in English

5.3.1. Background

- The paper / article discusses / deals with / analyses / considers / explains / describes / establishes / introduces...
- develops / presents / provides / studies / represents / features / contains / concentrates on...
- demonstrates the feasibility of...
- opens up a new field / issue gives / aims to give a comprehensive account of...
- offers a solution to...
- serves as an introduction to...

5.3.2. Purpose

- The main objective / goal / purpose of the paper / article is...
- Our aim with this paper was...
- The aim of the article is...
- Much attention is given to...

5.3.3. Results

- It has been found that ...
- The results show that ...
- The results thus obtained are compatible with ...

5.3.4. Conclusions

- In conclusion...
- The following conclusions are drawn...
- Summing up the results, it can be concluded that...
- In conclusion, it is evident that this study has shown...
- This paper has clearly shown that...
- It has been demonstrated / shown / found that...
- The findings suggest that this approach could also be useful for...
- The findings are of direct practical relevance.

6. Conclusion

The editors expect authors to adhere to the recommended structure of the abstract.

Author Contributions: The contributions of the authors are equal. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analysed during this study. Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gibson, B. F. Editorial: Structured abstracts: A key to enhanced information transfer. *Physical Review C* **84,** 030001–1. doi:10.1103/physrevc.84.030001 (Sept. 2011).
2. Andrade, C. How to write a good abstract for a scientific paper or conference presentation. *Indian Journal of Psychiatry* **53,** 172–175. doi:10.4103/0019-5545.82558 (2011).

Information about the authors

Dmitry S. Kulyabov—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Probability Theory and Cyber Security of RUDN University; Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: kulyabov-ds@rudn.ru, phone: +7 (495) 952-02-50, ORCID: 0000-0002-0877-7063, ResearcherID: I-3183-2013, Scopus Author ID: 35194130800)

Leonid A. Sevastianov—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Computational Mathematics and Artificial Intelligence of RUDN University (e-mail: sevastianov-la@rudn.ru, phone: +7 (495) 955-07-83, ORCID: 0000-0002-1856-4643, ResearcherID: B-8497-2016, Scopus Author ID: 8783969400)

DOI: 10.22363/2658-4670-2025-33-1-5-9 EDN: DJEBHI

Структура аннотации

Д. С. Кулябов[1, 2], Л. А. Севастьянов[1, 2]

[1] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[2] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. Описывается общие требования к аннотации научной статьи. Рекомендуется использовать подход структурированной аннотации.

Ключевые слова: аннотация, структурированная аннотация, исследовательская статья

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 10–26
http://journals.rudn.ru/miph

Research article

UDC 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-10-26
EDN: AMSSFO

Modeling and optimization of an $M/M/1/K$ queue with single working vacation, feedback, and impatience timers under N-policy

Abir Kadi[1], Mohamed Boualem[2], Nassim Touche[2], Aimen Dehimi[1]

[1] University of Bejaia, Laboratory of Applied Mathematics, 06000 Bejaia, Algeria

[2] University of Bejaia, Research Unit LaMOS (Modeling and Optimization of Systems), 06000 Bejaia, Algeria

(received: July 8, 2024; revised: August 10, 2024; accepted: August 25, 2024)

Abstract. This work presents an intensive study of a single server finite-capacity queueing model with impatience timers which depend on the server's states, feedback, and a single working vacation policy operating under an N-policy discipline. We examine the scenario where the server must wait for the number of customers to reach N to start a regular busy period; otherwise, the server will initiate a working vacation or switch to the dormant state if the number of customers increases. By applying the Markov recursive method, the steady-state probabilities were derived. Various performance metrics were visually depicted to assess diverse system parameter configurations. After constructing the expected cost function of the model, Grey Wolf Optimization (GWO) algorithm is utilized to determine the optimum values of the service rates μ^* and μ_v^*. Numerical examples are provided to validate the theoretical findings, offering insights into this intricate system.

Key words and phrases: Queueing system with impatience, N-policy, vacation policy, feedback, GWO algorithm, cost optimization

For citation: Kadi, A., Boualem, M., Touche, N., Dehimi, A. Modeling and optimization of an $M/M/1/K$ queue with single working vacation, feedback, and impatience timers under N-policy. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 10–26. doi: 10.22363/2658-4670-2025-33-1-10-26. edn: AMSSFO (2025).

1. Introduction

Queueing theory is a branch of applied mathematics focused on studying and analyzing processes within a wide range of service, production, management, and communication systems. These systems involve repetitive occurrences of homogeneous events. Examples include consumer services, information reception, processing, and transmission systems, automated production lines, telecommunication networks, among others.

Queueing theory offers invaluable solutions to mitigate long queues in real-life settings, providing mathematical frameworks to address such challenges [1–4].

In contemporary times, research on queueing systems with impatient customers has garnered increased attention. Impatience emerges as a prominent characteristic, as customers often feel anxious and restless while awaiting services. Hence, it is imperative for queueing systems research to incorporate patron impatience to accurately reflect real-world conditions. Typically, customer dissatisfaction is modeled through concepts like balking and reneging. When faced with a long queue, customers may opt to leave (balk) or refrain from entering the system altogether if the queue is excessively lengthy. This behavior is observed in practical systems such as hospital emergency rooms, especially when dealing with critically ill patients [2, 5–7].

The concept of the N-policy was first introduced by Yadin and Naor [8]. Meena et al. [9] developed a non-Markovian system for machine repair problems with finite capacity and the N-policy, a variant of the vacation queueing system under Bernoulli feedback. Bouchentouf et al. [10] studied an $M/M/c$ queue with customers' impatience and Bernoulli feedback, incorporating a variant of multiple vacations. Kathirvel [11] addressed a finite single-server model with an optional second service and obtained the waiting time distribution of customers in the waiting hall using Laplace–Stieltjes transforms. Kadi et al. [12] analyzed an $M/M/2$ machine repairable model for both single and multiple vacations under the triadic policy using a matrix geometric method. Adou et al. [13] applied a comparative study between peremptive and non-preemptive scheduling algorithms for the operation of a wireless network slicing model. Additionally, Sharma et al. [14] examined the N-policy with the vacation interruption concept and impatience behavior. Rajadurai et al. [15] introduced an $M/G/1$ queueing system with multiple working vacations, vacation interruption with feedback, and server breakdowns, applying the results to Simple Mail Transfer Protocol (SMTP) applications. Furthermore, Boualem [16] utilized stochastic orders to analyze an $M/G/1$ queueing model where the server undergoes breakdown and repair processes. Hilquias et al. [17] compared *RED* and $TailDrop$ algorithms with the renovation mechanism for different models of queueing systems. Goswami [18] investigated the interrelationship between F-policy and N-policy considering inter-arrival times of customers and geometrically distributed service times. Vemuri et al. [19] determined the optimum value of the control parameter N for the expected cost function of an $M^X/M/1$ system. Bouchentouf et al. [5] analyzed balking and server state-dependent reneging queues using generating functions and obtained the steady-state solution. Additionally, Bouchentouf et al. [2] analyzed a multi-server model with finite capacity, multiple synchronous working vacations, and balking. As a highly effective approximation function, ANFIS (Adaptive neuro-fuzzy inference system) is considered the best tool of neural and fuzzy systems which ensures smoothness to reduce the optimization search space from fuzzy systems. ANFIS has been Commonly implemented for prediction, control, and optimization Assignments. Divya and Indhira [20] applied ANFIS computing to analyze an unreliable model under hybrid vacation and feedback. Moreover, Indumathi et al. [21] applied the ANFIS to assess the accuracy of cost results for an $M/M/2$ system with two heterogeneous servers and Catastrophe and Restoration phenomena. Recently, Dehimi et al. [22] studied a finite multi-server model operating under a hybrid hiatus policy and applied ANFIS to validate the accuracy of diverse performance metrics achieved.

Scientists believe that the Grey Wolf Optimizer (GWO) has an exceptional hunting mechanism. From this perspective, optimization using the grey wolf method has gained prominence. GWO is employed as a meta-heuristic algorithm known for its strong optimal search capability in studying queueing system costs. The GWO algorithm was initially introduced by [23], who demonstrated its ability to address issues of instability and convergence accuracy, highlighting its superior accuracy and faster convergence speed. As a meta-heuristic algorithm for queueing systems, GWO was introduced in the seminal work of [24], where the authors applied GWO to study repairable systems in cloud computing using an N-policy. Also Dehimi et al. [25] applied GWO to derive the optimal service rates for a finite

$M/M/c$ system with synchronous differentiated working vacation policy, Bernoulli feedback, and impatience.

The originality of the work done is to study a finite-capacity single-server Markovian queue with feedback and impatience operating under the N-policy discipline along with the provision to go on a working vacation (single working vacation policy), simultaneously, which was not studied in past literature. Our study utilized GWO (Grey Wolf Optimizer) to minimize the cost function for this queueing system. To the best of our knowledge, GWO has not been used so far in optimizing this category's queuing systems. We use it to determine service rates couple (μ^*, μ_v^*) to minimize the expected cost function. Our model accurately represents an airport security checkpoint, where passengers arrive randomly following a Poisson process. The main contributions of this work can be summarized as follows:

- Obtaining the steady-state solution for the system using the Markov recursive method, which provides a powerful approach for analyzing steady-state probabilities and stochastic processes.
- Deriving important performance metrics, then performing a numerical analysis to validate the analytical results and investigate the impact of different system parameters on the performance metrics.
- Applying the GWO algorithm to acquire the optimum values of μ and μ_v of the optimum cost function. This offers decision-makers significant management information for designing management policy.

The paper is structured as follows: Section 2 introduces the mathematical description of the proposed model along with a practical application. Section 3 establishes the analysis of the system. Section 4 examines various performance measures based on the steady-state probability distribution of the system. Section 5 provides numerical simulations for the performance metrics. Section 6 applies the GWO algorithm to determine the optimal parameters for the cost function. Finally, a comprehensive conclusion for the study is presented.

2. Description of the queueing model

We consider a finite-capacity $M/M/1/K$ queue with feedback, single working vacation, impatient customers, and N-policy. The assumptions of the proposed model are as follows:

1. Customers arrive according to a Poisson process with rate λ. Upon arrival, a customer decides to join the queue with probability β_i or balk (refuse to join) with probability $1 - \beta_i$, where $0 \leqslant i \leqslant K$. Specifically, $\beta_0 = 1$ and $\beta_K = 0$.
2. Customers, unsatisfied with the service provided, either leave the system with probability θ or return with probability $\theta' = 1 - \theta$. Feedback customers are treated as new arrivals.
3. Service times follow an exponential distribution with rate μ during regular busy periods and μ_v during vacation periods ($\mu_v < \mu$). Service is provided on a First-In-First-Out (FIFO) discipline.
4. Upon entering the queue, a customer activates a timer T_0 (during dormant periods) or T_2 (during working vacation periods). These timers follow exponential distributions with rates ξ_0 and ξ_2 respectively. A customer leaves the queue with probability α and may return to the system with probability $1 - \alpha$. Impatient clients are only active during vacation and dormant states.
5. When the number of customers in the system drops to zero while at least one server is active, the server enter a working vacation period (WV):
   - During a WV, the server serves arriving customers at a rate lower than the regular service rate. At the end of the vacation period, if the system size is N, the server switches to a regular busy period and starts operating under the N-policy.

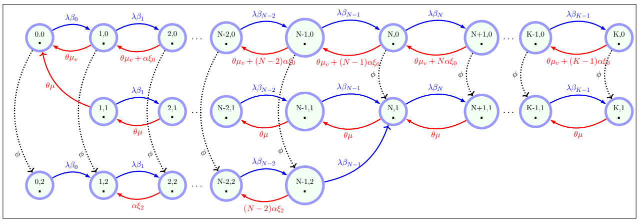


Figure 1. State transition rate diagram

- Under a Single Working Vacation (SWV) policy, server takes only one WV whenever the system becomes empty. If there are at least N customers at the end of the vacation period, the server resumes operation under the N-policy with the usual service rate. Otherwise, servers remain dormant in the system until N customers arrive instead of taking another WV.
- Vacation durations are assumed to follow an exponential distribution with rate ϕ.

We assume that inter-arrival times, service times, and vacation times are mutually independent. Figure 1 illustrates the state-transition rate diagram of the model.

2.1. Implementation of the model in practical scenarios

Consider an airport security checkpoint where passengers arrive randomly following a Poisson process with rate λ. Upon arrival, passengers decide whether to join the security queue based on its current length. Each passenger decides to join the queue based on a probability β_i, potentially opting to delay or avoid joining if the queue reaches its capacity K (balking). Passengers dissatisfied with wait times may leave (quit) with probability θ or return later (feedback) with probability $\theta' = 1 - \theta$. A single security agent performs screenings at rate μ during peak hours and a reduced rate μ_v during off-peak times. Impatient passengers may permanently leave the queue during quieter periods with probability α. If the queue becomes empty with one active agent, they enter a working vacation period, conducting screenings at a reduced rate for an exponentially distributed duration with rate ϕ. The agent resumes full-speed screening if enough passengers accumulate (N or more) by the end of the working vacation, ensuring effective passenger flow management under varying demand and operational conditions.

3. Queueing model analysis

Consider the state of the system at time t characterized by the random variables $N(t)$ representing the system size and $J(t)$ indicating the server's state, defined as:

$$j(t) = \begin{cases} 0, & \text{if the server is in WV period,} \\ 1, & \text{ if the server is active during regular busy period,} \\ 2, & \text{if the server is dormant.} \end{cases}$$

We can define the process $\{N(t), j(t)\}$ with its state space as follows:

$$\Omega = \{(0,0) \cup (n,j) \,:\, 0 \leqslant n \leqslant K, j = 0,1,2\}.$$

Applying Markov process, we derive the following set of equations describing the steady state:

$$(\phi + \lambda\beta_0)P_{0,0} = \theta\mu_v P_{0,1} + \theta\mu P_{1,1}, \quad n = 0, \tag{1}$$
$$(\phi + \theta\mu_v + (n-1)a\xi_0 + \lambda\beta_n)P_{0,n} = (\theta\mu_v + na\xi_0)P_{0,n+1} + \lambda\beta_{n-1}P_{0,n-1}, \quad 1 \leqslant n \leqslant K-1,$$
$$(\phi + \theta\mu_v + (K-1)a\xi_0)P_{0,K} = \lambda\beta_{K-1}P_{0,K-1}, \quad n = K, \tag{2}$$
$$(\theta\mu + \lambda\beta_1)P_{1,1} = \theta\mu P_{1,2}, \quad n = 1,$$
$$(\lambda\beta_n + \theta\mu)P_{1,n} = \lambda\beta_{n-1}P_{1,n-1} + \theta\mu P_{1,n+1}, \quad 2 \leqslant n \leqslant N-1, \tag{3}$$
$$(\lambda\beta_N + \theta\mu)P_{1,N} = \lambda\beta_{N-1}P_{1,N-1} + \lambda\beta_{N-1}P_{2,N-1} + \theta\mu P_{1,N+1} + \phi P_{0,N}, \quad n = N, \tag{4}$$
$$(\lambda\beta_n + \theta\mu)P_{1,n} = \lambda\beta_{n-1}P_{0,n-1} + \theta\mu P_{1,n+1} + \phi P_{0,n}, \quad N+1 \leqslant n \leqslant K-1,$$
$$\theta\mu P_{1,K} = \lambda\beta_{K-1}P_{1,K-1} + \phi P_{0,K}, \quad n = K, \tag{5}$$
$$\lambda\beta_0 P_{2,0} = \phi P_{0,0}, \quad n = 0, \tag{6}$$
$$(\lambda\beta_n + (n-1)a\xi_2)P_{2,n} = \lambda\beta_{n-1}P_{2,n-1} + na\xi_2 P_{2,n+1} + \phi P_{0,n}, \quad 1 \leqslant n \leqslant N-2,$$
$$((N-2)a\xi_2 + \lambda\beta_{N-1})P_{2,N-1} = \lambda\beta_{N-2}P_{2,N-2} + \phi P_{0,N-1}, \, n = N-1. \tag{7}$$

The normalization condition:

$$\sum_{n=0}^{K} P_{0,n} + \sum_{n=1}^{K} P_{1,n} + \sum_{n=0}^{N-1} P_{2,n} = 1. \tag{8}$$

3.1. Steady-state solution

In this sub-section, we employ the Markov recursive method to obtain the steady-state distribution of the server states.

Theorem 1. *The steady-state probabilities of the system size during the working vacation period are provided as follows:*

$$P_{0,n} = \chi_n P_{0,K}, \tag{9}$$

where

$$\chi_n = \begin{cases} 1, & n = K, \\ \dfrac{\phi + \theta\mu_v + (K-1)a\xi_0}{\lambda\beta_{K-1}}, & n = K-1, \\ \left(\dfrac{\lambda\beta_{n+1} + \phi + \theta\mu_v + na\xi_0}{\lambda\beta_n}\right)\chi_{n+1} - \left(\dfrac{\theta\mu_v + (n+1)a\xi_0}{\lambda\beta_n}\right)\chi_{n+2}, & \\ 0 \leqslant n \leqslant K-2. & \end{cases}$$

The steady-state probabilities during the busy period are given by:

$$P_{1,n} = \Upsilon_n P_{0,K} + t_{n-2}P_{2,N-1},$$

where

$$\Upsilon_n = \begin{cases} \dfrac{\phi + \lambda\beta_0}{\theta\mu}\chi_0 - \dfrac{\theta\mu_v}{\theta\mu}\chi_1, & n = 1, \\[2ex] \dfrac{\lambda\beta_1 + \theta\mu}{\theta\mu}\Upsilon_1, & n = 2, \\[2ex] \left(\dfrac{\lambda\beta_{n-1} + \theta\mu}{\theta\mu}\right)\Upsilon_{n-1} - \left(\dfrac{\lambda\beta_{n-2}}{\theta\mu}\right)\Upsilon_{n-2}, & 3 \leqslant n \leqslant N, \\ \left(\dfrac{\lambda\beta_{n-1} + \theta\mu}{\theta\mu}\right)\Upsilon_{n-1} - \left(\dfrac{\lambda\beta_{n-2}}{\theta\mu}\right)\Upsilon_{n-2} - \dfrac{\phi}{\theta\mu}\chi_{n-2}, & N+1 < n \leqslant K-1. \end{cases}$$

and

$$t_n = \begin{cases} 0, & 0 \leqslant n \leqslant N-2, \\[2ex] \dfrac{\lambda\beta_{N-1}}{\theta\mu}, & n = N-1, \\[2ex] \left(\dfrac{\lambda\beta_{N+1} + \theta\mu}{\theta\mu}\right)t_{N-1}, & n = N, \\ \left(\dfrac{\lambda\beta_{n+1} + \theta\mu}{\theta\mu}\right)t_{n-1} - \left(\dfrac{\lambda\beta_n}{\theta\mu}\right)t_{n-2}, & N+1 < n \leqslant K-2. \end{cases}$$

and the stationary probabilities for the dormant period verify the equation:

$$P_{2,n} = \Delta_n P_{2,N-1} + \Theta_n P_{0,K},$$

where

$$\Delta_n = \begin{cases} 0, & N \leqslant n \leqslant K, \\ 1 & n = N-1, \\ \left(\dfrac{(N-2)a\xi_2 + \lambda\beta_{N-1}}{\lambda\beta_{N-1}}\right), & n = N-2, \\ \left(\dfrac{\lambda\beta_{n+1} + na\xi_2}{\lambda\beta_n}\right)\Delta_{n+1} - \left(\dfrac{(n+1)a\xi_2}{\lambda\beta_n}\right), & n = N-3, \\ \left(\dfrac{\lambda\beta_{n+1} + na\xi_2}{\lambda\beta_n}\right)\Delta_{n+1} - \left(\dfrac{(n+1)a\xi_2}{\lambda\beta_n}\right)\Delta_{n+2}, & 0 \leqslant n < N-3. \end{cases}$$

$$\Theta_n = \begin{cases} 0, & N-1 \leqslant n \leqslant K, \\ \left(\dfrac{\phi\psi_{N-1}}{\lambda\beta_{N-2}}\right), & n = N-2, \\ \left(\dfrac{\lambda\beta_{n+1} + na\xi_2}{\lambda\beta_n}\right)\psi_{n+2} - \left(\dfrac{\phi}{\lambda\beta_n}\right)\psi_{n+1}, & n = N-3, \\ \left(\dfrac{\lambda\beta_{n+1} + na\xi_2}{\lambda\beta_n}\right)\Theta_{n+1} - \left(\dfrac{(n+1)a\xi_2}{\lambda\beta_n}\right)\Theta_{n+2} - \dfrac{\phi}{\lambda\beta_n}\psi_{n+1}, & 0 \leqslant n < N-3. \end{cases}$$

$$\Theta_n = \Theta_{n+1} - \frac{\phi}{\lambda\beta_n}\chi_{n+1}, \quad 0 \leqslant n \leqslant N-2.$$

Then, using equation (7)*, we get:*

$$P_{2,N-1} = \left(\frac{\phi\chi_0 - \lambda\beta_0\Theta_0}{\lambda\beta_0\Delta_0}\right)P_{0,K}.$$

Finally, we have:

$$P_{0,K} = \left[\sum_{n=0}^{K} \chi_n + \sum_{n=1}^{K} (\Upsilon_n + \varsigma_0 t_n) + \sum_{n=0}^{N-1} (\varsigma_0 \varDelta_n + \varTheta_n)\right]^{-1}.$$

Where, $\varsigma_0 = \left(\dfrac{\phi\chi_0 - \lambda\beta_0\varTheta_0}{\lambda\beta_0\varDelta_0}\right)$.

4. Performance metrics

In this section, useful performance measures for different server states are presented.

1. The probability of the server being in a working vacation state:

$$P_{Vacation} = \sum_{n=0}^{K} P_{0,n} = \sum_{n=0}^{K} \chi_n P_{0,K}.$$

2. The probability of the server being in a busy state:

$$P_{Busy} = \left[\sum_{n=1}^{K} \Upsilon_n + \varsigma_0 \sum_{n=1}^{K-2} t_n\right] P_{0,K}.$$

3. The probability of the server being in a dormant state:

$$P_{Dormant} = \left[\varsigma_0 \sum_{n=0}^{N-1} \varDelta_n + \sum_{n=0}^{N-2} \varTheta_n\right] P_{0,K}.$$

4. The probability that the server is idle:

$$P_{idle} = \left[\chi_0 + \varsigma_0 \sum_{n=0}^{N-1} \varDelta_n + \sum_{n=0}^{N-2} \varTheta_n\right] P_{0,K}.$$

5. The expected number of customers in the queue:

$$L_s = \sum_{n=0}^{K} nP_{0,n} + \sum_{n=1}^{K} (n-1)P_{1,n} + \sum_{n=0}^{N-1} (n-1)P_{2,n} = \left[\sum_{n=0}^{K} \chi_n + \sum_{n=1}^{K} \Upsilon_n + \varsigma_0 \left(\sum_{n=0}^{K-2} t_n + \sum_{n=0}^{N-1} \varDelta_n\right) + \sum_{n=0}^{N-2} \varTheta_n\right] P_{0,K}.$$

6. The average balking rate:

$$BR = \sum_{n=0}^{K} n\lambda(1-\beta_n)P_{0,n} + \sum_{n=1}^{K} (n-1)\lambda(1-\beta_n)P_{1,n} + \sum_{n=0}^{N-1} n\lambda(1-\beta_n)P_{2,n} = \sum_{n=0}^{K} n\lambda(1-\beta_n)\chi_n P_{0,K} +$$
$$+ \sum_{n=1}^{K} (n-1)\lambda(1-\beta_n)\left(\sum_{n=1}^{K} \Upsilon_n + \varsigma_0 \sum_{n=1}^{K-2} t_n\right) P_{0,K} + n\lambda(1-\beta_n)\left(\varsigma_0 \sum_{n=0}^{N-1} \varDelta_n + \varTheta_n\right) P_{0,K}.$$

7. The average reneging rate:

$$RR = \xi_0 \sum_{n=0}^{K} (n-1)P_{0,n} + \xi_2 \sum_{n=0}^{N-1} (n-1)P_{2,n} = \left[\xi_0 \sum_{n=0}^{K} (n-1)\chi_n + \xi_2 \left(\varsigma_0 \sum_{n=0}^{N-1} (n-1)\varDelta_n + \varTheta_n\right)\right] P_{0,K}.$$

8. The average rate of lost customers:

$$AR = BR + RR.$$

9. The expression for the expected waiting time of customers in the system:

$$W_s = \frac{L_s}{\lambda'}, \text{ where } \lambda' = \lambda - AR.$$

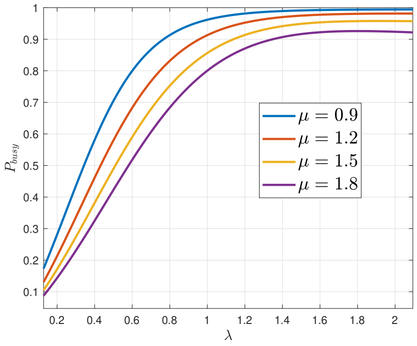


Figure 2. P_{busy} vs. λ for different values of μ

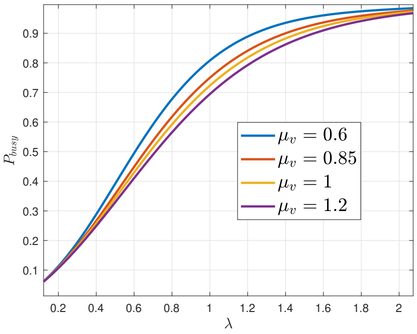


Figure 3. P_{busy} vs. λ for different values of μ_v

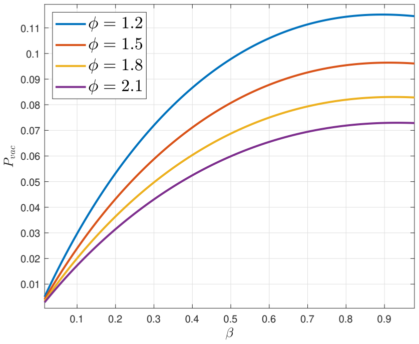


Figure 4. P_{vac} vs. β for multiple values of ϕ

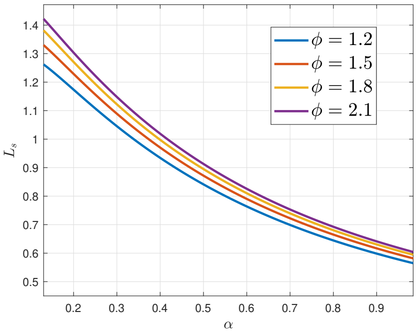


Figure 5. L_s vs. α for multiple values of ϕ

5. Numerical results

In this section, key numerical results are presented graphically to illustrate the impact of various system parameters on different performance metrics. These graphs were generated using the **MATLAB** program. For this purpose, the following set of system parameters was fixed: $\lambda = 0.4$, $\mu = 1.5$, $\mu_v = 0.7$, $\phi = 0.6$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$, $N = 3$, and $K = 5$.

5.1. Analysis of findings

The numerical experiments systematically analyzed the sensitivity of various system parameters on performance measures. Based on these analyses, the following key observations and potential managerial recommendations have been identified:

1. Effect of λ (arrival rate): With an increasing value of λ, several factors are significantly affected. The service rates μ and μ_v increase correspondingly, leading to an increase in the probability of the server being in the busy state P_{busy} (see Figures 2 and 3). Additionally, as the capacity K increases, the expected number of customers in the queue L_s also increases (see Figure 7).

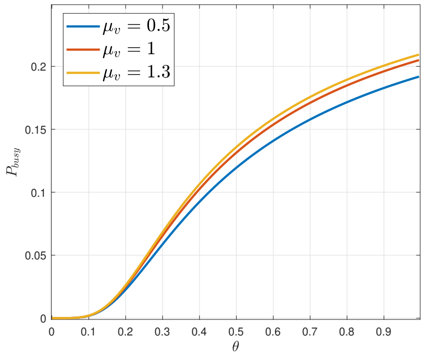


Figure 6. P_{busy} vs. θ for different values of μ_v

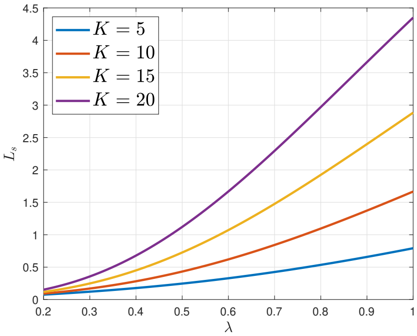


Figure 7. L_s vs. λ for different values of K

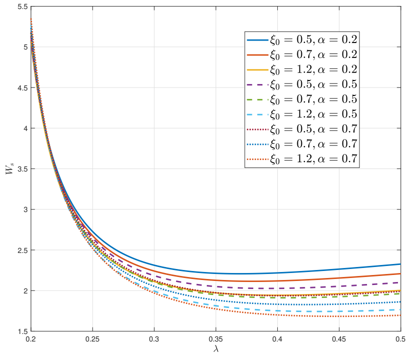


Figure 8. W_s vs. λ for different values of ξ_0 and α

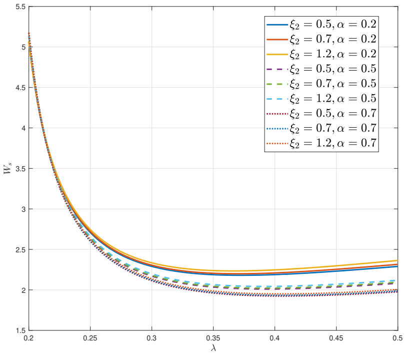


Figure 9. W_s vs. λ for different values of ξ_2 and α

2. Effect of parameter ϕ (vacation rate): As the balking rate β and vacation rate ϕ increase, P_{vac} gradually increases (see Figure 4). Conversely, an increase in the probability α while increasing the vacation rate ϕ leads to a decrease in the expected number of customers in the queue L_s (see Figure 5).
3. Effect of parameter θ (feedback rate): As the feedback rate θ increases, the probability P_{busy} of the server being in the busy state increases for different service rates μ_v (see Figure 6).
4. Effect of parameters ξ_0 and ξ_2 (impatience rate): A lower impatience rate ξ_0 (or ξ_2) and parameter α (probability of leaving the queue) lead to an increase in the expected waiting time of customers in the system W_s (see Figures 8 and 9). Consequently, due to this impatience, there is an increase in the number of customers L_s.

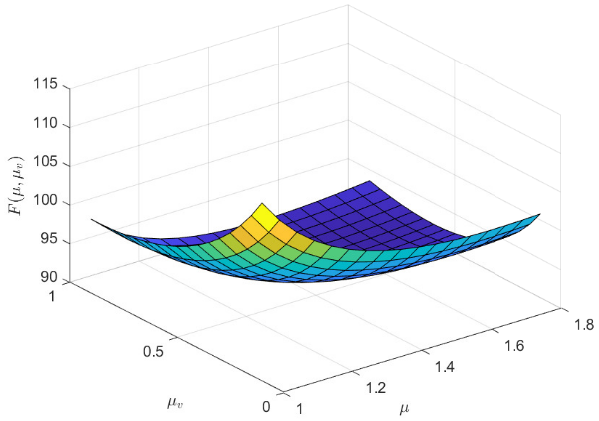


Figure 10. $F(\mu, \mu_v)$ vs. μ and μ_v

6. Numerical cost optimum

6.1. Cost model

The utilization of various cost factors in terms of cost function can play a crucial role in optimizing and enhancing the system availability by considering the benefits associated with different design variable options. Using the Grey Wolf Optimizer (GWO) algorithm . We will assess the expected total cost and determine the optimal values of (μ^*, μ_v^*) the decision variables based on the costs are fixed as: $C_{WV} = 30$, $C_B = 20$, $C_d = 15$, $C_s = 20$, $C_{AR} = 30$, $C_{\mu_v} = 10$, $C_\mu = 20$.

Then we define the total expected cost per unit of time of the system as follows:

$$F = C_{wv}P_{vac} + C_B P_{busy} + C_d(P_{dormant} + P_{idle}) + C_s L_s + C_{AR}AR + \mu C_{mu} + \mu_v Cv,$$

where,

- C_{wv}: denotes the cost per unit time when the server is on a working vacation period,
- C_B: denotes the cost per unit time when the server is on a busy period,
- C_d: denotes the cost per unit time when the server is on a dormant or idle period,
- C_s: denotes the cost per unit time when a customer joins the queue and waits for service,
- C_{AR}: denotes the cost per unit time when a customer leaves the queue,
- C_{μ_v} (resp. C_μ): denotes the cost per service per unit time during normal busy period (resp. working vacation period).

6.2. GWO—Grey Wolf optimizer

The Grey Wolf optimizer (GWO) algorithm is inspired by the leadership organization and hunting strategy of grey wolves in nature. GWO represents a recent innovation in cost optimization methods. This meta-heuristic algorithm effectively explores the search space and converges to the optimal solution by simulating the hunting behavior of grey wolves. This section presents a practical

Table 1
The optimal (μ^*, μ_v^*) and $F^*(\mu^*, \mu_v^*)$ vs. θ, when $\theta = 0.2 : 0.8$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\lambda = 0.4$, $\phi = 0.6$, $\beta = 0.8$, $\alpha = 0.7$, $N = 3$ and $K = 5$

	μ^*	μ_v^*	$F(\mu^*, \mu_v^*)$
$\theta = 0.2$	3.8882	1.0039	143.9779
$\theta = 0.4$	2.4800	0.8102	104.9952
$\theta = 0.6$	2.3922	0.5771	80.0489
$\theta = 0.8$	2.3003	0.4035	83.5105

Table 2
The optimal (μ^*, μ_v^*) and $F^*(\mu^*, \mu_v^*)$ vs. K and N, when $\lambda = 0.4$, $\phi = 0.6$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$

		μ^*	μ_v^*	$F(\mu^*, \mu_v^*)$
$K = 10$	$N = 3$	2.2297	0.8871	126.2557
	$N = 5$	1.8132	0.9201	80.9963
	$N = 7$	1.6801	0.0225	69.3733
$K = 15$	$N = 3$	2.5312	1.3530	151.4877
	$N = 5$	1.8374	0.3141	90.3144
	$N = 7$	1.6846	0.0179	70.9343
$K = 18$	$N = 3$	2.7078	1.5911	164.5993
	$N = 5$	1.8545	0.4193	90.3144
	$N = 7$	1.6944	0.0105	71.4826
$K = 20$	$N = 3$	2.8737	1.7380	172.7564
	$N = 5$	1.8656	0.4832	94.6289
	$N = 7$	1.7018	0.0956	71.9032

application of GWO in the field of queueing systems. The objective of this study is to determine the optimal service rates (μ, μ_v) that minimize the expected cost function. Given the complexity of higher-order non-linear optimization problems, it is recommended to employ nonlinear optimization techniques to find these optimal solutions. For optimizing the cost model, the GWO algorithm is applied by initially setting parameters to obtain the optimal values of (μ^*, μ_v^*).

The optimization problem can be written as:

$$\min_{\mu,\mu_v} F(\mu, \mu_v)\text{s.t}\begin{cases}\mu - \mu_v > 0\\ \mu_v > 0\\ (\mu, \mu_v) \in \mathbb{R}_+^2.\end{cases}$$

Table 3

The optimal (μ^*, μ_v^*) and $F^*(\mu^*, \mu_v^*)$ vs. λ, when $\lambda = 0.6 : 1.8$, $\phi = 0.6$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$, $N = 3$ and $K = 5$

	μ^*	μ_v^*	$F(\mu^*, \mu_v^*)$
$\lambda = 0.6$	2.9196	0.9762	101.4854
$\lambda = 0.8$	2.7014	1.3150	117.0471
$\lambda = 1$	3.1103	1.6098	131.1372
$\lambda = 1.2$	3.4837	1.8806	144.1337
$\lambda = 1.4$	3.8283	2.1393	156.2874
$\lambda = 1.8$	4.1440	2.2980	167.7924

Table 4

The optimal (μ^*, μ_v^*) and $F^*(\mu^*, \mu_v^*)$ vs. ϕ, when $\phi = 0.5 : 5.5$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\lambda = 0.4$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$, $N = 3$ and $K = 5$

	μ^*	μ_v^*	$F(\mu^*, \mu_v^*)$
$\phi = 0.5$	1.7138	0.5838	83.6245
$\phi = 1.5$	1.7043	0.7143	83.5471
$\phi = 2.5$	1.7059	0.7596	83.4869
$\phi = 3.5$	1.7070	0.7801	83.4537
$\phi = 4.5$	1.7082	0.7939	83.4330
$\phi = 5.5$	1.7090	0.8034	83.4188

- Table 1 clearly shows that higher feedback probabilities have a negative effect on the optimal cost and (μ^*, μ_v^*).
- By examining Table 2, it is evident that the optimal cost and (μ^*, μ_v^*) increase with the system capacity K and the number of customers N.
- The impact of arrival rates can be observed through Table 3. The optimal cost and (μ^*, μ_v^*) directly increase as the arrival rate takes larger values.
- Table 4 demonstrates that the vacation rate has a direct impact on the optimal cost and (μ^*, μ_v^*); as the vacation rate increases, so do the optimal cost and (μ^*, μ_v^*).
- Upon analyzing Table 5, it becomes clear that the optimal cost and (μ^*, μ_v^*) exhibit a significant increase as the values of the impatience rates ξ_0 and ξ_2 increase.
- Figure 11, shows that a higher arrival rate λ leads to an increase in the expected cost $F(K, N)$ and (K, N) values.

Table 5

The optimal (μ^*, μ_v^*) and $F^*(\mu^*, \mu_v^*)$ vs. ξ_0 and ξ_0, when $\xi_0 = [0.5; 1; 1.5]$, $\xi_2 = [1; 1.2; 1.4; 1.6; 1.8]$, $\lambda = 0.4$, $\phi = 0.6$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$, $N = 3$ and $K = 5$

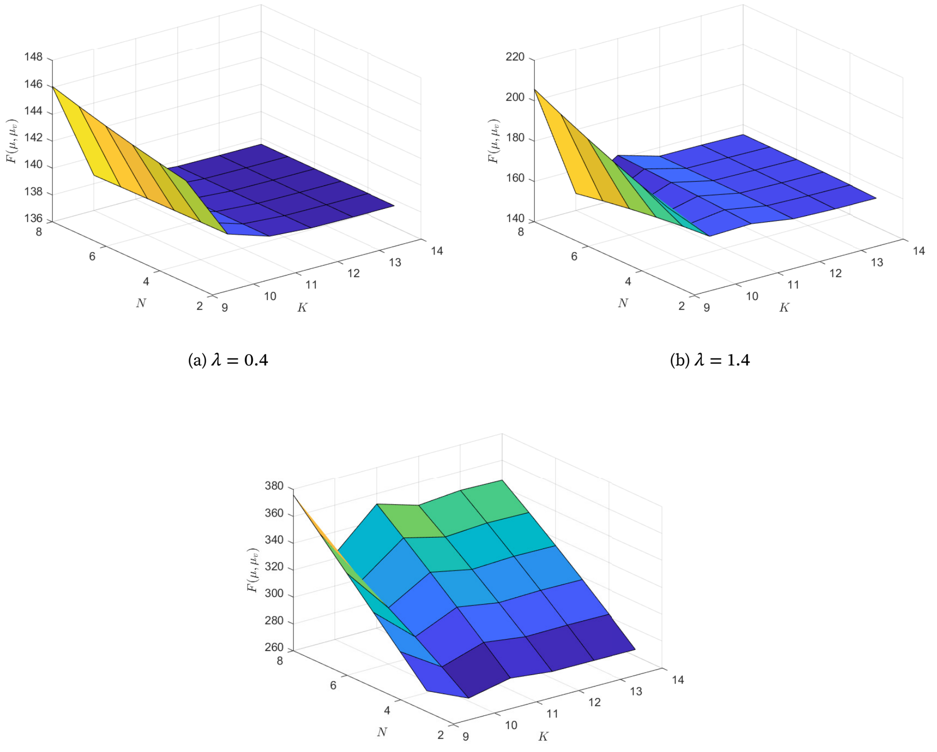
		μ^*	μ_v^*	$F(\mu^*, \mu_v^*)$
$\xi_0 = 0.5$	$\xi_2 = 1$	1.5167	0.4454	77.5487
	$\xi_2 = 1.2$	1.5718	0.4910	79.2290
	$\xi_2 = 1.4$	1.6209	0.5348	80.7922
	$\xi_2 = 1.6$	1.6673	0.5747	82.2544
	$\xi_2 = 1.8$	1.7117	0.6102	83.6292
$\xi_0 = 1$	$\xi_2 = 1$	1.4912	0.2287	77.4745
	$\xi_2 = 1.2$	1.5448	0.2778	79.1299
	$\xi_2 = 1.4$	1.5955	0.3210	80.6766
	$\xi_2 = 1.6$	1.6408	0.3601	82.1278
	$\xi_2 = 1.8$	1.6865	0.3968	83.4951
$\xi_0 = 1.5$	$\xi_2 = 1$	1.4666	0.0140	77.3672
	$\xi_2 = 1.2$	1.5203	0.0641	79.0078
	$\xi_2 = 1.4$	1.5713	0.1081	80.5449
	$\xi_2 = 1.6$	1.6182	0.1483	81.9898
	$\xi_2 = 1.8$	1.6608	0.1854	83.3530

7. Conclusion

This study is based on a finite capacity queueing model tailored for application in the scenario of passengers at the airport. Our model represents single working vacations, N-policy, feedback during both dormant and working vacation periods, and impatience timers which depend on the server's states. By employing the Markov recursive method, we derived closed-form expressions for steady-state probabilities and performance metrics. Furthermore, various metrics were graphically shown and discussed. Additionally, we applied the GWO algorithm intending to obtain the optimum service rates for the expected cost function and also conducted a comprehensive numerical analysis to assess the influence of various parameters on the results obtained.

In the future, the model could be enhanced by incorporating characteristic customer behaviors such as priority mechanisms, as well as multiple optional services. This extension would broaden the model applicability but also increase the complexity of calculations.

Author Contributions: Conceptualization, A. Kadi, M. Boualem and N. Touche; methodology, A. Kadi, M. Boualem and N. Touche; software, A. Kadi, N. Touche and A. Dehimi; formal analysis, A. Kadi, M. Boualem and A. Dehimi; writing-original draft preparation, A. Kadi, M. Boualem and A. Dehimi; writing-review and editing, A. Kadi, M. Boualem, N. Touche and A. Dehimi; visualization, A. Kadi, M. Boualem, N. Touche and A. Dehimi; supervision, M. Boualem and N. Touche. All authors have read and agreed to the published version of the manuscript.



(a) $\lambda = 0.4$

(b) $\lambda = 1.4$

(c) $\lambda = 2.4$

Figure 11. The expected cost $F^*(K, N)$ vs. λ, when $\phi = 0.6$, $\xi_0 = 0.5$, $\xi_2 = 1.8$, $\theta = 0.7$, $\beta = 0.8$, $\alpha = 0.7$

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Acknowledgments: The authors would like to thank the editor for their support and guidance throughout the review process.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bouchentouf, A. A., Cherfaoui, M. & Boualem, M. Analysis and Performance Evaluation of Markovian Feedback Multi-Server Queueing Model with Vacation and Impatience. *American Journal of Mathematical and Management Sciences* **40,** 261–282. doi:10.1080/01966324.2020.1842271 (Nov. 2020).
2. Bouchentouf, A. A., Boualem, M., Yahiaoui, L. & Ahmad, H. A multi-station unreliable machine model with working vacation policy and customers' impatience. *Quality Technology & Quantitative Management* **19,** 766–796. doi:10.1080/16843703.2022.2054088 (Apr. 2022).

3. Gorbunova, A., Zaryadov, I., Samouylov, K. & Sopin, E. A Survey on Queuing Systems with Parallel Serving of Customers. *Discrete and Continuous Models and Applied Computational Science* **25,** 350–362. doi:10.22363/2312-9735-2017-25-4-350-362 (Jan. 2017).
4. Bose, S. K. *An introduction to queueing systems* (Springer Science & Business Media, Dec. 2013).
5. Bouchentouf, A. A., Boualem, M., Cherfaoui, M. & Medjahri, L. Variant vacation queueing system with Bernoulli feedback, balking and server's states-dependent reneging. *Yugoslav Journal of Operations Research* **31,** 557–575. doi:10.2298/yjor200418003b (Jan. 2021).
6. Bouchentouf, A. A., Cherfaoui, M. & Boualem, M. Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers. *OPSEARCH* **56,** 300–323. doi:10.1007/s12597-019-00357-4 (Feb. 2019).
7. Chettouf, A., Bouchentouf, A. A. & Boualem, M. *A Markovian Queueing Model for Telecommunications Support Center with Breakdowns and Vacation Periods* in *Operations Research Forum* **5** (2024), 22. doi:10.1007/s43069-024-00295-y.
8. Yadin, M. & Naor, P. Queueing Systems with a Removable Service Station. *Journal of the Operational Research Society* **14,** 393–405. doi:10.1057/jors.1963.63 (Dec. 1963).
9. Meena, R. K., Jain, M., Assad, A., Sethi, R. & Garg, D. Performance and cost comparative analysis for $M/G/1$ repairable machining system with N-policy vacation. *Mathematics and Computers in Simulation* **200,** 315–328. doi:10.1016/j.matcom.2022.04.012 (Oct. 2022).
10. Bouchentouf, A. A., Medjahri, L., Boualem, M. & Kumar, A. Mathematical analysis of a Markovian multi-server feedback queue with a variant of multiple vacations, balking and reneging. *Discrete and Continuous Models and Applied Computational Science* **30,** 21–38. doi:10.22363/2658-4670-2022-30-1-21-38 (Apr. 2022).
11. Kathirvel, J. A single server perishable inventory system with N additional options for service. *Journal of Mathematical Modeling* **2,** 187–216 (May 2015).
12. Kadi, A., Touche, N., Bouchentouf, A. A. & Boualem, M. Finite-capacity $M/M/2$ machine repair model with impatient customers, triadic discipline, and two working vacation policies. *Journal of Mathematical Modeling* **13,** 183–200. doi:10.22124/jmm.2024.27095.2384 (2024 Article in press).
13. Adou, K. Y. B., Markova, E. V. & Zhbankova, E. A. Performance analysis of queueing system model under priority scheduling algorithms within 5G networks slicing framework. *Discrete and Continuous Models and Applied Computational Science* **30,** 5–20. doi:10.22363/2658-4670-2022-30-1-5-20 (Apr. 2022).
14. Sharma, R. Optimal N-policy for unreliable server queue with impatient customer and vacation interruption. *Journal of Reliability and Statistical Studies* **10,** 83–96 (May 2017).
15. Rajadurai, P., Saravanarajan, M. & Chandrasekaran, V. A study on $M/G/1$ feedback retrial queue with subject to server breakdown and repair under multiple working vacation policy. *Alexandria Engineering Journal* **57,** 947–962. doi:10.1016/j.aej.2017.01.002 (June 2018).
16. Boualem, M. Stochastic analysis of a single server unreliable queue with balking and general retrial time. *Discrete and Continuous Models and Applied Computational Science* **28,** 319–326. doi:10.22363/2658-4670-2020-28-4-319-326 (Dec. 2020).
17. Hilquias, V. C. C., Zaryadov, I. S. & Milovanova, T. A. Queueing systems with different types of renovation mechanism and thresholds as the mathematical models of active queue management mechanism. *Discrete and Continuous Models and Applied Computational Science* **28,** 305–318. doi:10.22363/2658-4670-2020-28-4-305-318 (Dec. 2020).
18. Goswami, V. Relationship between randomized F-policy and randomized N-policy in discrete-time queues. *OPSEARCH* **53,** 131–150. doi:10.1007/s12597-015-0220-y (Sept. 2015).
19. Vemuri, V. K., Siva, V., Kotagiri, C. & Bethapudi, R. T. Optimal strategy analysis of an N-policy two-phase $M^X/M/1$ queueing system with server startup and breakdowns. *Opsearch* **48,** 109–122. doi:10.1007/s12597-011-0046-1 (June 2011).

20. Divya, K. & Indhira, K. Performance Analysis And ANFIS Computing Of An Unreliable Markovian Feedback Queueing Model Under A Hybrid Vacation Policy. *Mathematics and Computers in Simulation* **218,** 403–419. doi:10.1016/j.matcom.2023.12.004 (Apr. 2024).
21. Indumathi, P. & Karthikeyan, K. ANFIS-Enhanced $M/M/2$ Queueing Model Investigation in Heterogeneous Server Systems with Catastrophe and Restoration. *Contemporary Mathematics,* 2482–2502. doi:10.37256/cm.5220243977 (June 2024).
22. Dehimi, A., Boualem, M., Kahla, S. & Berdjoudj, L. ANFIS computing and cost optimization of an $M/M/c/M$ queue with feedback and balking customers under a hybrid hiatus policy. *Croatian Operational Research Review* **15,** 159–170. doi:10.17535/crorr.2024.0013 (2024).
23. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software* **69,** 46–61. doi:10.1016/j.advengsoft.2013.12.007 (Mar. 2014).
24. Chahal, P. K., Kumar, K. & Soodan, B. S. Grey wolf algorithm for cost optimization of cloud computing repairable system with N-policy, discouragement and two-level Bernoulli feedback. *Mathematics and Computers in Simulation* **225,** 545–569. doi:10.1016/j.matcom.2024.06.005 (Nov. 2024).
25. Dehimi, A., Boualem, M., Bouchentouf, A. A., Ziani, S. & Berdjoudj, L. Analytical and Computational Aspects of a Multi-Server Queue With Impatience Under Differentiated Working Vacations Policy. *Reliability: Theory & Applications* **19,** 393–407. doi:10.24412/1932-2321-2024-379-393-407 (2024).

Information about the authors

Abir Kadi—PhD student, at Department of Mathematics, Laboratory of Applied Mathematics, 06000 Bejaia, Algeria (e-mail: abir.kadi@univ-bejaia.dz, ORCID: 0009-0000-6668-0270)

Mohamed Boualem—Full Professor of Applied Mathematics at the Department of Automation, Telecommunications, and Electronics at the University of Bejaia, Algeria. Permanent Researcher at the Research Unit LaMOS (Modeling and Optimization of Systems). (e-mail: mohammed.boualem@univ-bejaia.dz, phone: +213770556978, ORCID: 0000-0001-9414-714X)

Nassim Touche—Full Professor of Mathematics at Department of Operations research, Faculty of Exact Sciences, Research Unit LaMOS (Modeling and Optimization of Systems) (e-mail: nassim.touche@univ-bejaia.dz, ORCID: 0000-0002-9185-3433)

Aimen Dehimi—PhD student, at Department of Mathematics, Laboratory of Applied Mathematics, 06000 Bejaia, Algeria (e-mail: aimen.dehimi@univ-bejaia.dz, ORCID: 0009-0009-1221-9898)

УДК 519.872, 519.217

PACS 07.05.Tp, 02.60.Pn, 02.70.Bf

DOI: 10.22363/2658-4670-2025-33-1-10-26 EDN: AMSSFO

Моделирование и оптимизация очереди $M/M/1/K$ с одиночным рабочим отпуском, обратной связью и таймерами нетерпимости в рамках N-политики

Абир Кади[1], Мохамед Буалем[2], Нассим Туш[2], Аймен Дехими[1]

[1] Университет Беджаи, Лаборатория прикладной математики, 06000 Беджаи, Алжир

[2] Университет Беджаи, Исследовательская группа LaMOS (Моделирование и оптимизация систем), 06000 Беджаи, Алжир

Аннотация. Этот труд представляет собой интенсивное исследование модели очереди с одним сервером и конечной ёмкостью, с таймерами нетерпимости, зависящими от состояний сервера, с обратной связью и политикой одиночного рабочего отпуска, функционирующей в рамках дисциплины N-политики. Мы рассматриваем сценарий, при котором сервер должен дождаться, пока количество клиентов не достигнет N, чтобы начать обычный рабочий период; в противном случае сервер начнёт рабочий отпуск или перейдёт в неактивное состояние, если количество клиентов увеличится. С помощью метода Марковской рекурсии были получены вероятности в установившемся состоянии. Различные показатели производительности были визуально изображены для оценки различных конфигураций параметров системы. После построения ожидаемой функции стоимости модели используется алгоритм Оптимизация серых волков (GWO) для определения оптимальных значений коэффициентов обслуживания μ и μ_v. Приведены числовые примеры для проверки теоретических выводов, что позволяет глубже понять эту сложную систему.

Ключевые слова: система очередей с нетерпимостью, N-политика, политика отпуска, обратная связь, алгоритм GWO, оптимизация стоимости

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 27–45
http://journals.rudn.ru/miph

Research article

UDC 004.85
PACS 07.05.Tp,
DOI: 10.22363/2658-4670-2025-33-1-27-45

EDN: AFZDUC

Statistical and density-based clustering techniques in the context of anomaly detection in network systems: A comparative analysis

Aleksandr S. Baklashov[1, 2], Dmitry S. Kulyabov[1, 3]

[1] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[2] V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya St, Moscow 117997, Russian Federation

[3] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: November 25, 2024; revised: December 10, 2024; accepted: December 12, 2024)

Abstract. In the modern world, the volume of data stored electronically and transmitted over networks continues to grow rapidly. This trend increases the demand for the development of effective methods to protect information transmitted over networks as network traffic. Anomaly detection plays a crucial role in ensuring net security and safeguarding data against cyberattacks.

This study aims to review statistical and density-based clustering methods used for anomaly detection in network systems and to perform a comparative analysis based on a specific task. To achieve this goal, the authors analyzed existing approaches to anomaly detection using clustering methods. Various algorithms and clustering techniques applied within network environments were examined in this study.

The comparative analysis highlights the high effectiveness of clustering methods in detecting anomalies in network traffic. These findings support the recommendation to integrate such methods into intrusion detection systems to enhance information security levels.

The study identified common features, differences, strengths, and limitations of the different methods. The results offer practical insights for improving intrusion detection systems and strengthening data protection in network infrastructures.

Key words and phrases: intrusion detection systems, network systems, clustering methods

For citation: Baklashov, A. S., Kulyabov, D. S. Statistical and density-based clustering techniques in the context of anomaly detection in network systems: A comparative analysis. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 27–45. doi: 10.22363/2658-4670-2025-33-1-27-45. edn: AFZDUC (2025).

1. Introduction

With the growing frequency and complexity of attacks targeting information systems [1], such as DDoS and data breaches, having a system to protect information from these types of attacks becomes a vital aspect of network design. Anomaly detection serves as a key element in ensuring the security of information systems, as anomalies in network traffic often indicate unauthorized access attempts or other forms of intrusion. That is why developing effective methods to detect deviations in network traffic behavior remains a crucial challenge.

In [2], the authors provided a comprehensive overview of methods, systems, and tools for anomaly detection in network traffic. That study placed particular attention on classifying the approaches available at the time, including clustering-based methods. However, due to its publication date, the review does not fully reflect recent advances in data processing and modern algorithms. The study in [2] also overlooked key aspects of density-based clustering methods such as DBSCAN, HDBSCAN, etc.

Therefore, actualization and in-depth study of clustering methods in the context of modern network traffic paradigms presents a highly relevant research direction. Recent approaches offer new opportunities to improve the accuracy and efficiency of anomaly detection.

Researchers now explore a wide range of anomaly detection techniques. For example, some researchers detect it using deep unfolding methods to reconstruct normal and anomalous data flows based on sparse and full-dimensional components [3]. The others use approaches such as Isolation Forest and autoencoders to detect anomalies [4].

Many researchers focus on neural network-based techniques. In [5], the authors investigate deep learning to address the issue of false positives in anomaly detection. At the same time, the others combine traditional approaches with machine learning techniques [6]. These methods have demonstrated strong performance in recognizing different data patterns, making them particularly effective for solving cybersecurity challenges.

This study proposes a clustering-based approach for network intrusion detection. The proposed method aims to serve as the first line of defense against network attacks within intrusion detection systems (IDS), which monitor events occurring within information systems or their individual components.

The objective of this study is to analyze existing clustering methods for anomaly detection and to perform a comparative assessment. To achieve this, the study analyzes and evaluates several clustering algorithms and summarizes their properties in a comparison table. An experimental section follows, presenting results for each method applied to a specific dataset.

The article includes several sections, each addressing a specific aspect of the research:

The section “Types of intrusion detection systems and anomaly detection methodology” defines IDS, outlines main IDS types, and introduces clustering methods.

The section “Methods and instruments” presents a detailed review and analysis of clustering techniques. Subsections cover partitioning methods (e.g., k-means, k-medoids), hierarchical clustering, and density-based clustering (DBSCAN, HDBSCAN, OPTICS). A summary table at the end of this section facilitates the comparison of these methods.

The section “Practical application of clustering methods in network anomaly detection systems” presents the comparative analysis results of six clustering algorithms, tested on a real dataset. This section includes results and interpretation of the metrics obtained for each clustering method experimentally.

The section “Results” presents the metrics obtained by application of the clustering methods to the specific dataset. Data is presented in the summary table, heatmap and text format.

The section “Discussion” summarizes the experimental findings and justifies the selection of the most suitable clustering method for network anomaly detection.

The section “Conclusion” outlines the main outcomes and discusses directions for future research.

2. Types of intrusion detection systems and anomaly detection methodology

An intrusion detection system (IDS) is software or hardware that analyzes network traffic or computer activity to identify potential unauthorized access attempts, attacks, or intrusions into computer systems or networks [7]. IDS detects a wide range of threats, including intrusions, viruses, worms, denial-of-service (DoS) attacks, and other anomalous behaviors, and alerts administrators about it, forcing them to enable timely defensive actions [8].

Researchers classify intrusion detection systems into two main types based on detection methods and the way of deployment [9, 10]:

1. Network-based intrusion detection systems (NIDS) analyze network traffic for anomalies by intercepting data at the network adapter level or via network devices such as switches and routers. NIDS can detect attacks before they reach the target system.
2. Host-based intrusion detection systems (HIDS) run on individual computers and monitor activity at the operating system level, including file system changes, registry modifications, event logs, and other system parameters. HIDS typically detect attacks targeting a specific host and may offer additional insights about system compromise.

This study focuses on clustering methods as a key tool for identifying anomalies in network-based intrusion detection systems (NIDS). Dividing network traffic into clusters that represent normal and abnormal behavior plays a critical role in designing effective NIDS and ranks among the most successful techniques for detecting network anomalies.

This clustering approach enhances both the accuracy and efficiency of IDS work.

The section titled “Methods and instruments” presents a comparison of six clustering algorithms: k-means, k-medoids, hierarchical clustering, DBSCAN, HDBSCAN, and OPTICS.

This analysis aims to further selection of the most appropriate method for anomaly detection in NIDS based on their performance, accuracy, strengths, and limitations.

3. Methods and instruments

This chapter presents a comparative analysis of clustering methods applicable to the stated problem (see Fig. 1).

The analysis focuses on three main types of clustering methods [11]:

1. Partitioning clustering;
2. Hierarchical clustering;
3. Density-based clustering.

To cluster network traffic into two categories this study evaluates the following methods:

- Two partitioning clustering methods: k-means and k-medoids;
- A hierarchical clustering method;
- Three density-based clustering methods: DBSCAN, HDBSCAN, and OPTICS.

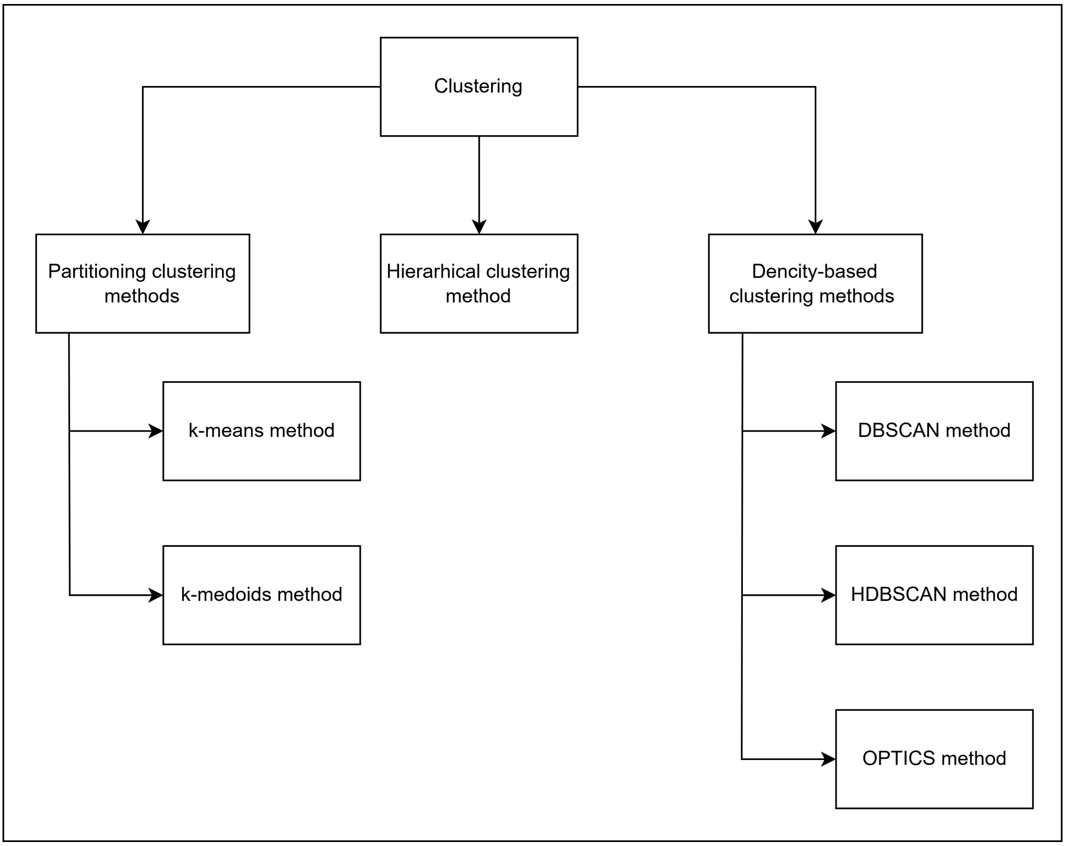

Clustering
Partitioning clustering methods
Hierarhical clustering method
Dencity-based clustering methods
k-means method
k-medoids method
DBSCAN method
HDBSCAN method
OPTICS method

Figure 1. Clustering methods

3.1. Partitioning clustering methods

3.1.1. The k-means clustering method

The k-means method divides data into a predefined number of clusters k. The algorithm begins with selecting random centroids—mean points that represent each cluster. It then assigns each data point to the nearest centroid and after that recalculates centroids as the arithmetic mean of all points within the cluster. These steps repeat iteratively until convergence is achieved, after which the algorithm evaluates the clustering quality (see Fig. 2).

This method offers several advantages relevant to the current task, including simplicity, scalability, interpretability, and versatility.

However, it also introduces some limitations. The algorithm shows high sensitivity to initial centroid placement and outliers, which may distort the final results. Additionally, it does not guarantee an optimal solution [12].

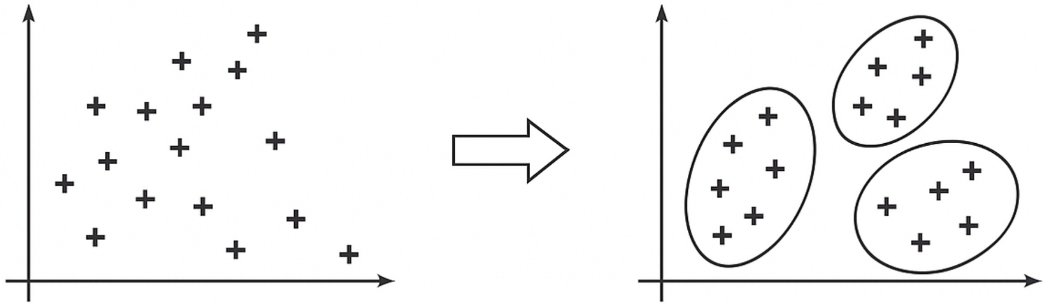


Figure 2. K-means method

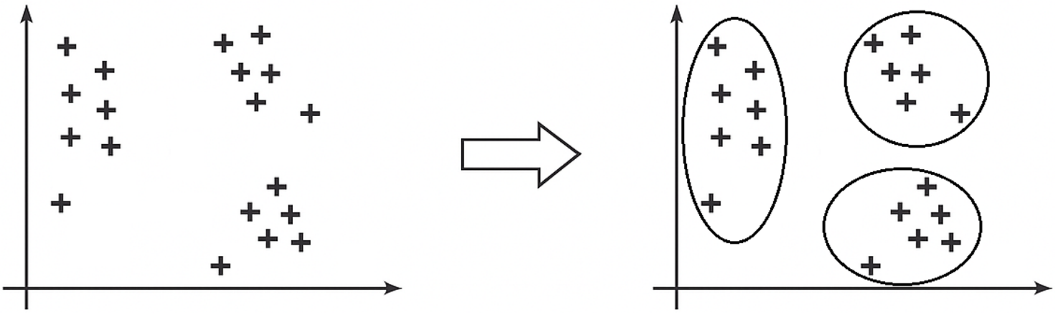


Figure 3. K-medoids method

3.1.2. The k-medoids clustering method

The k-medoids algorithm extends the k-means method by requiring that cluster centers (medoids) belong to the input data points. The algorithm begins by selecting k random points as initial medoids, where k denotes the predefined number of clusters. Then, it assigns each data point to the cluster by choosing the smallest distance from the medoid to the point, using a selected distance metric.

After assigning all points, the algorithm calculates the cost of the current clustering. It then attempts to replace one of the existing medoids with a non-medoid point and recalculates the cost. If the new cost exceeds or remains equal to the previous value, the algorithm reverts the change and stops. Otherwise, it accepts the new medoid and repeats the process from this step (see Fig. 3).

This method has several advantages. It executes a certain number of iterations. Compared to k-means, it provides more determined cluster centers since they correspond to actual data points. Also, this algorithm supports various distance metrics for cluster assignment.

However, k-medoids also introduces some drawbacks. It remains sensitive to the initial choice of medoids, and the randomness in selecting replacement candidates may lead to inconsistent results across different runs [13].

3.1.3. The hierarchical clustering method

The hierarchical clustering algorithm begins by treating each data point as an individual cluster. It then iteratively merges the closest clusters until all points belong to a single cluster. At the end of the

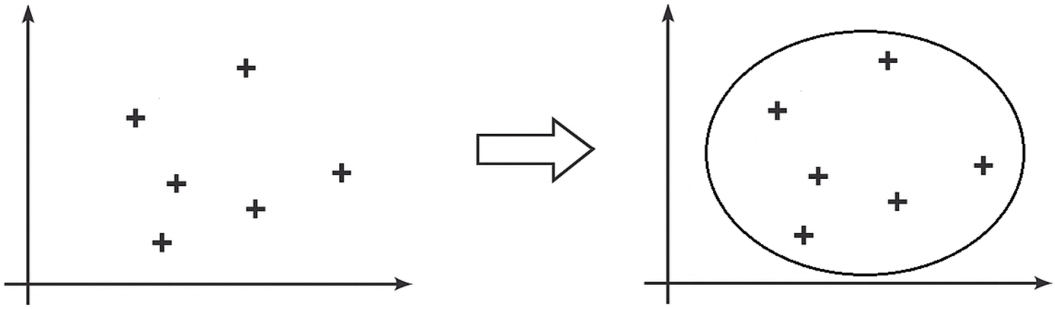


Figure 4. Hierarchical method

process, the algorithm constructs a dendrogram that illustrates the hierarchy of cluster merging and allows the selection of an optimal number of clusters (see Fig. 4).

This method provides several advantages, including a high level of interpretability and versatility.

However, it also has some drawbacks. The algorithm suffers from high computational complexity and is sensitive to the choice of distance metric [12].

3.2. Density-based clustering methods

3.2.1. The DBSCAN clustering method

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm identifies clusters by locating areas of high point density. The process begins by selecting a random point and defining its neighborhood. If the number of neighboring points exceeds a predefined threshold MinPts (minimum number of neighbors), the point becomes a core point.

The algorithm then forms clusters by uniting core points and cluster-related boundary points. Points that do not belong to any cluster and are not part of dense areas are treated as outliers or noise (see Fig. 5).

This method offers several advantages relevant to anomaly detection tasks, such as strong performance on large datasets.

However, DBSCAN has some limitations. It shows sensitivity to the choice of parameters and struggles to cluster data with varying densities or scales. In addition, the computational cost increases with large values of the input parameters [12].

3.2.2. The HDBSCAN clustering method

The HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) algorithm extends DBSCAN by incorporating a hierarchical clustering approach. After running the DBSCAN core procedure, HDBSCAN iteratively merges clusters based on the distances between them, forming a hierarchy of density-connected areas (see Fig. 6).

Overall, HDBSCAN offers a powerful clustering technique with several advantages over the classical DBSCAN. These include automatic determination of the number of clusters and greater robustness to parameter selection.

At the same time, applying HDBSCAN may require increased computational resources and can encounter limitations when dealing with datasets that exhibit highly irregular or complex structures [14].

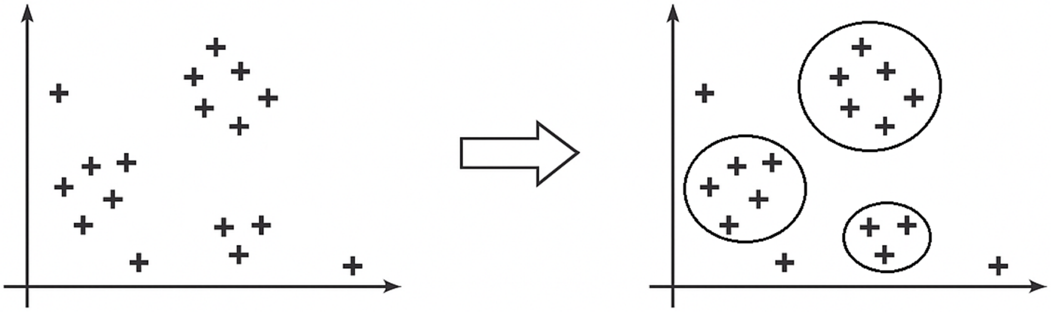


Figure 5. DBSCAN

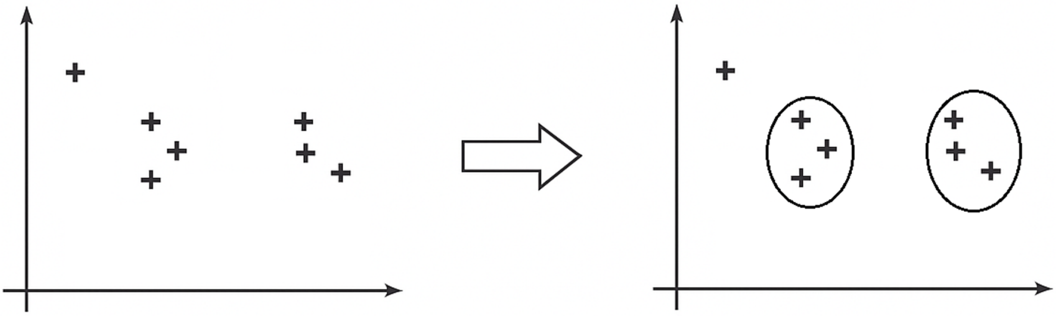


Figure 6. HDBSCAN

3.2.3. The OPTICS clustering method

The OPTICS (Ordering Points To Identify the Clustering Structure) algorithm builds upon the ideas introduced in DBSCAN, allowing the detection of clusters with varying densities by ordering points and dynamically selecting parameters. The algorithm requires setting a minimum number of neighbors (MinPts) and optionally an ε-radius. Compared to DBSCAN, OPTICS is less sensitive to the exact choice of these parameters.

After setting the parameters, the algorithm identifies the neighbors of each data point within the ε-radius and calculates the density of each point based on the number of its neighbors. OPTICS then generates an ordered list of points by iteratively traversing the dataset, starting from an random unvisited point and proceeding through its neighbors. This list forms the basis for constructing a dendrogram and selecting a density threshold that separates clusters [15] (see Fig. 7).

OPTICS offers several advantages, including the ability to detect clusters of arbitrary shape, robustness to noise and outliers, and no need to specify the number of clusters in advance. It also supports a variety of distance metrics for cluster formation.

However, the algorithm still requires careful tuning of ε and MinPts, and its computational complexity becomes significant on large datasets [14].

3.3. Comparative table

Let’s make a summary table of the data (Table 1).

Table 1

Comparison of Clustering Methods

Method	Core Principle	Parameters	Outliers	Advantages	Disadvantages
k-means	Finds cluster centers and minimizes deviation from points.	Number of clusters	Sensitive to outliers	Simplicity, scalability, interpretability, versatility	Sensitive to initial conditions and outliers, no guarantee of optimality
k-medoids	Selects and updates medoids	Number of clusters	Less sensitive than k-means	Clusters defined by actual data points, supports various distance metrics	Sensitive to initial medoid selection, random replacement
Hierarchical Clustering	Merges and splits clusters based on inter-point distances	Number of clusters (optional)	Outliers affect hierarchy formation	High interpretability, versatility	High computational complexity, sensitive to distance metric
DBSCAN	Separates high- and low-density regions	ε, MinPts	Isolates outliers into separate clusters	Good performance on large datasets	Sensitive to parameters, computational complexity
HDBSCAN	Builds hierarchy of density-based clusters	*min_cluster_size*, *cluster_selection_epsilon*	Detects and ignores outliers	Automatic cluster number selection, robust to parameters	Higher computational demands than DBSCAN
OPTICS	Estimates density and performs ordered traversal	ε, MinPts	Robust to noise and outliers	Noise resilience, various distance metrics, arbitrary shape detection	Sensitive to parameters, computationally intensive

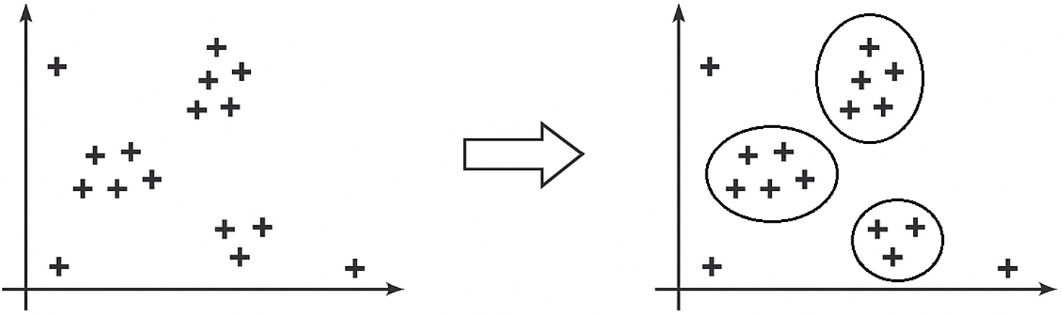


Figure 7. OPTICS

4. Practical application of clustering methods in network anomaly detection systems

This chapter presents a comparative analysis of six clustering methods described in the previous chapter that is conducted through experimental application to the task of separating network traffic into two categories: normal and anomalous. The analysis is based on the NSL-KDD dataset, which is an improved version of the well-known KDD Cup 1999 dataset [16, 17]. The NSL-KDD dataset contains 41 attributes and includes a label indicating whether the connection is normal or an anomaly [18]. The evaluation metrics and visualizations obtained from the experiments allow assessing the effectiveness, advantages, and limitations of each method, as well as identifying the most suitable approach for a Network Intrusion Detection System (NIDS).

4.1. Experiment description

The experiment was conducted using the NSL-KDD dataset, which contains both numerical and categorical features of network traffic, such as connection duration, number of bytes sent and received, as well as categorical features like protocol type and flags. The data underwent preprocessing: numerical features were normalized using StandardScaler, and categorical features were encoded using OneHotEncoder. Dimensionality reduction was performed using PCA, retaining 10 principal components to accelerate computation and facilitate analysis [19].

The clustering procedure was used to partition the data into groups corresponding to normal and anomalous traffic. The quality of clustering was evaluated using the following metrics: precision, recall, F1-score, execution time, silhouette coefficient, Calinski-Harabasz index (*ch_index*), number of clusters (*n_clusters*), noise ratio, cluster purity, and Precision-Recall AUC ($PR - AUC$). A cluster was classified as anomalous if more than 50% of the traffic within it was anomalous.

4.2. The analysis of clustering methods

4.2.1. K-means

In the K-means method, the number of clusters was fixed at $n_{\text{clusters}} = 20$. As shown in Figure 8, the clusters exhibit clear separation; however, their sizes vary significantly, with a few large clusters dominating the distribution. The algorithm demonstrates sensitivity to outliers, which can distort centroid positions and reduce clustering quality, as it does not isolate noise into a separate category.

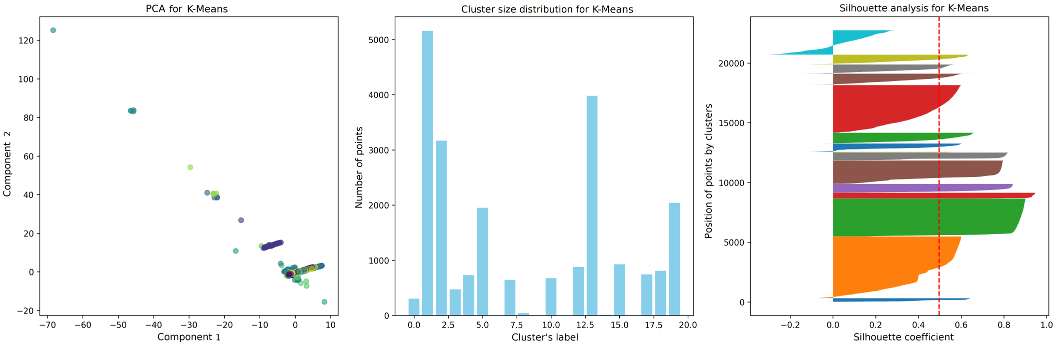


Figure 8. K-means—results

To evaluate clustering performance, the experiment used F1-score, execution time, and the silhouette coefficient. The F1-score reached 0.858, indicating a balanced correlation between precision and recall. The algorithm achieved exceptional computational efficiency, with an execution time of just 0.032 seconds. A silhouette coefficient of 0.496 suggests a tolerable level of compactness and separation among clusters.

The Calinski–Harabasz index further supports the presence of well-defined cluster structures. This method proves effective for processing large-scale datasets and performs well when anomalies form dense groups. Also, it demonstrates high speed and ease of implementation. However, its inability to explicitly handle noise stints its applicability in scenarios with significant outlier presence.

4.2.2. K-medoids

In the experiment with the k-medoids method, the number of clusters was set to 20, and the Manhattan distance metric was used for evaluating distances between objects. Unlike k-means, this method relies on medoids instead of centroids, making it less sensitive to outliers. The clustering visualizations (see Fig. 9) show a more balanced cluster size distribution, although the compactness becomes lower due to the heterogeneous data density.

The evaluation used F1-score, execution time, cluster purity, and silhouette coefficient. The F1-score reached 0.883, surpassing k-means and indicating higher clustering quality. However, the execution time amounted to 28.499 seconds, highlighting a significant drawback in computational efficiency. Cluster purity reached 0.868, reflecting a strong correspondence between the formed clusters and the actual data labels. However, the method's sensitivity to the initial medoid selection introduced variability, resulting in a low silhouette value of 0.221.

This approach achieves a strong balance between precision and anomaly coverage, as evidenced by high F1-score and purity. It proves more robust than k-means in noisy environments, although it similarly lacks explicit mechanisms for isolating noise points. The method's resource intensity must also be taken into account when applying it to large-scale data.

4.2.3. Hierarchical clustering

Hierarchical clustering begins by treating each data point as an individual cluster and gradually merges them based on distances between objects using the Ward linkage method until the desired number of clusters (10) is formed. As illustrated in Fig. 10, the resulting clusters exhibit a clear

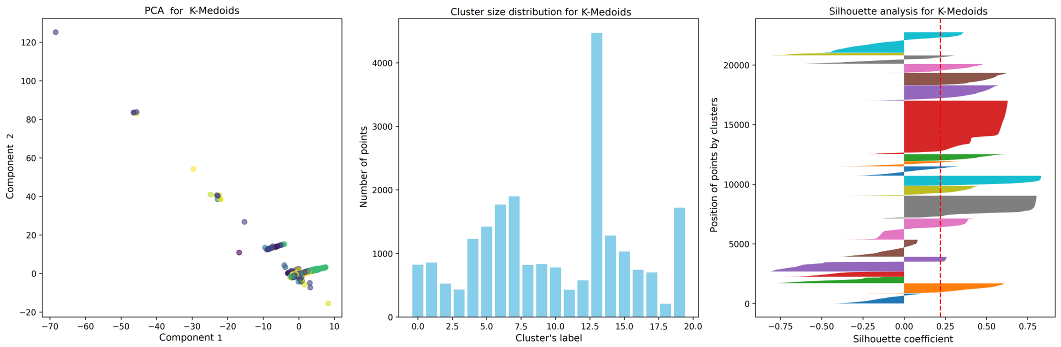


Figure 9. K-medoids—results

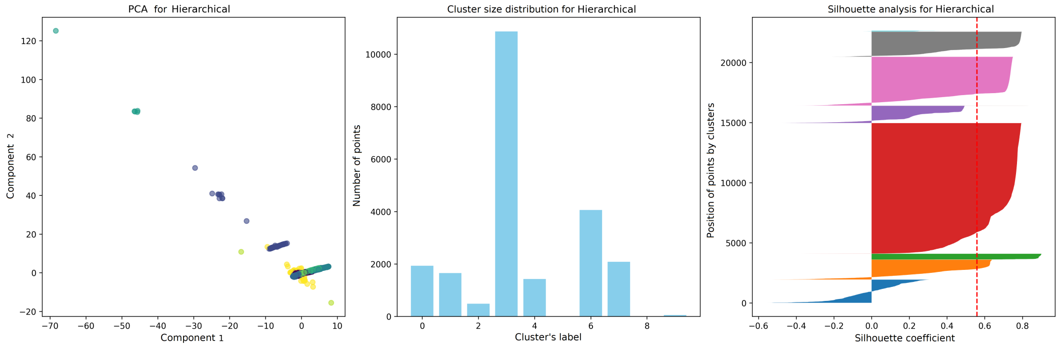


Figure 10. Hierarchical clustering—results

hierarchical structure; however, the cluster size distribution remains imbalanced, with one dominant large cluster.

Evaluation used the following metrics: F1-score, precision, Calinski–Harabasz index, silhouette coefficient, and execution time. The F1-score reached 0.824, indicating a reasonable balance between precision and recall. Precision achieved a high value of 0.939, reflecting strong classification accuracy. The Calinski–Harabasz index was 10159.3, suggesting excellent cluster separability and compactness. The silhouette coefficient was 0.559, which confirms passable intra-cluster cohesion and inter-cluster separation. However, the method required 17.538 seconds to complete, limiting its suitability for time-sensitive applications.

Hierarchical clustering allows us to identify internal data structure and relationships between clusters. Despite its interpretability and clustering effectiveness, the high computational complexity constrains its applicability to large-scale datasets or real-time systems.

4.2.4. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies clusters based on data density using the predefined parameters $\epsilon = 0.3$ (radius) and $min\_samples = 20$ (minimum number of neighbors). With these settings, DBSCAN detected a noise fraction of 0.14 relative to the total number of data points. As shown in Fig. 11, the resulting cluster size distribution is unbalanced,

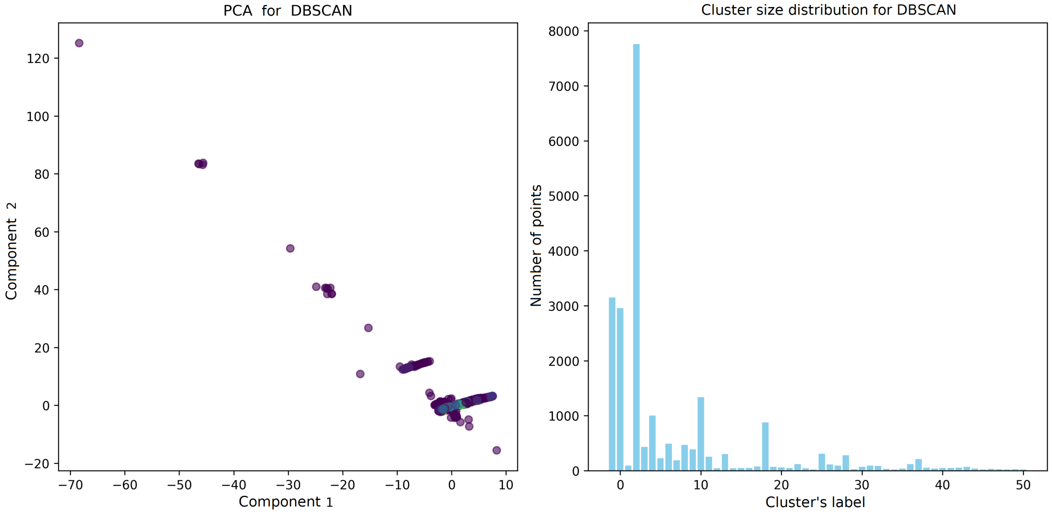


Figure 11. DBSCAN—results

with a single dominant cluster. Nevertheless, the remaining clusters maintain a moderate degree of compactness.

The clustering quality was assessed using F1-score, precision, execution time, and noise ratio. The F1-score reached 0.889, indicating high overall clustering performance. Precision was 0.86, suggesting an acceptable level of errors in the distribution of objects into clusters. Execution time amounted to 2.536 seconds, demonstrating good computational efficiency. However, the method's performance is highly sensitive to parameter tuning, which complicates its practical application. The silhouette coefficient is not a reliable measure for density-based methods and thus was not used as a primary evaluation metric.

DBSCAN offers an effective balance between computational speed and clustering quality. Yet, the requirement for careful parameter selection imposes additional complexity in real-world deployments.

4.2.5. HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) extends DBSCAN by building a hierarchy of clusters with automatic determination of the optimal number of clusters, which amounted to 243 in this case. The method employs parameters $min\_cluster\_size = 5$ and $cluster\_selection\_epsilon = 0.08$. Unlike DBSCAN, HDBSCAN replaces the fixed radius ε with a minimum cluster size, thereby offering greater flexibility in clustering process. With these parameters, the method identified noise accounting for 0.131 of the total number of instances.

Figure 12 shows that the cluster size distribution is relatively balanced, with several large groups and a moderate number of smaller ones. However, cluster compactness remained limited due to heterogeneous density across groups.

The evaluation relied on F1-score, recall, PR-AUC, and execution time. The F1-score reached 0.924, the highest among all methods considered. Recall was 0.988, reflecting excellent sensitivity in anomaly detection. The PR-AUC value of 0.932 further confirmed outstanding overall performance. Execution time was 4.194 seconds, acceptable for mid-scale problems.

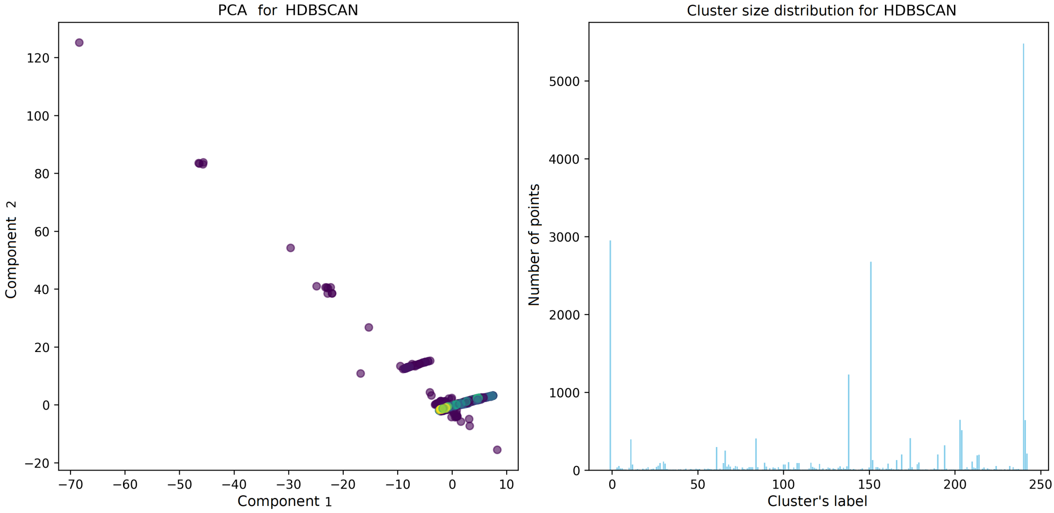


Figure 12. HDBSCAN—results

The combination of high recall and F1-score demonstrates the method's ability to detect nearly all anomalies, while the presence of a noise cluster assists outlier identification. Overall, HDBSCAN provides strong performance in clustering tasks involving heterogeneous densities and noisy data.

4.2.6. OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) is a density-based clustering method that utilizes the parameters *min_samples* $= 4$ and $\xi = 0.01$, resulting in the formation of 2066 clusters and a high noise level with a noise ratio of 0.386. As shown in Figure 13, the cluster size distribution is imbalanced, with one dominant large cluster and numerous small groups. Despite this, the main cluster maintains high compactness.

Clustering quality was assessed using F1-score, recall, execution time, and the silhouette coefficient. The F1-score reached 0.853, indicating a satisfactory result. Recall achieved 0.987, highlighting the method's strong anomaly detection capability. However, the execution time amounted to 41 seconds, which significantly reduces the method's feasibility for real-time applications. Furthermore, the high noise proportion of 0.39 emphasizes its sensitivity to data structure.

OPTICS effectively identifies clusters of arbitrary shapes and exhibits robustness to noise. Nevertheless, its computational complexity and sensitivity to parameter tuning limitations for deployment in performance-critical environments.

5. Results

The comparative analysis of clustering methods was conducted based on evaluation metrics presented in the heatmap (Figure 14) and the clustering visualizations. Table 2 summarizes the key performance indicators across all methods.

For the task of distinguishing between normal and anomalous traffic, it is essential to balance precision and recall, which is reflected in the F1-score, while also considering computational

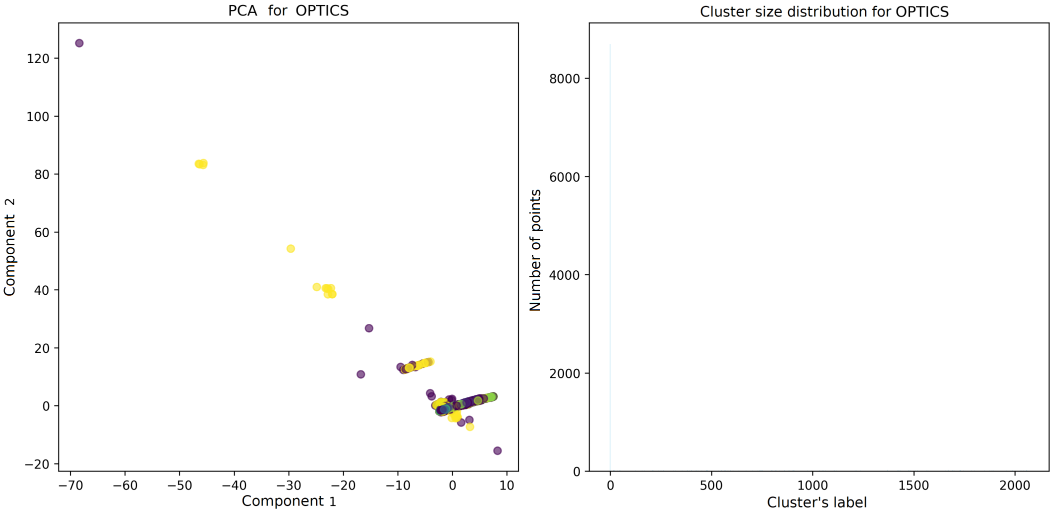


Figure 13. OPTICS—results

Table 2

Comparison of clustering metrics

	HDBSCAN	DBSCAN	K-Medoids	K-Means	OPTICS	Hierarchical
precision	0,868	0,860	0,895	0,896	0,751	0,939
recall	0,988	0,920	0,870	0,823	0,987	0,734
F1	0,924	0,889	0,883	0,858	0,853	0,824
time	4,194	2,536	28,499	0,032	41,203	17,538
silhouette	$-0,088$	0,192	0,221	0,496	$-0,038$	0,559
ch_index	229,125	829,498	1396,012	12286,905	15,215	10159,276
n_clusters	243	51	20	20	2066	10
noise_ratio	0,131	0,140	0	0	0,386	0
purity	0,846	0,774	0,868	0,845	0,601	0,822
pr_auc	0,932	0,913	0,783	0,692	0,873	0,912

efficiency, robustness to noise, and ease of parameter setting. Among the evaluated methods, HDBSCAN achieved the highest F1-score (0.92), followed by DBSCAN (0.89), K-Medoids (0.88), K-Means (0.86), OPTICS (0.85), and Hierarchical Clustering (0.82). In terms of execution time, K-Means (0.03 seconds) and DBSCAN (2.5 seconds) demonstrated the fastest performance, whereas OPTICS (41 seconds), K-Medoids (28 seconds), and Hierarchical Clustering (17 seconds) required significantly more time. HDBSCAN ranked second in speed with a runtime of 4 seconds. Regarding noise handling, HDBSCAN (0.13) and DBSCAN (0.14) effectively isolated outliers, while OPTICS (0.39) marked a substantial portion of data as noise. K-Means, K-Medoids, and Hierarchical Clustering

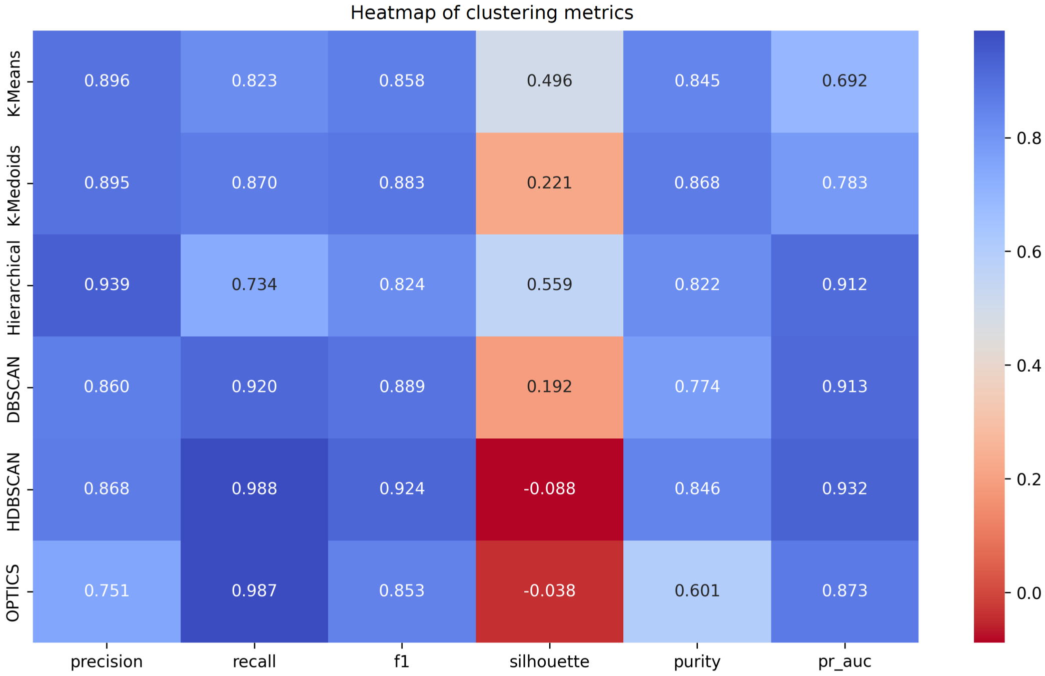


Figure 14. Heatmap

do not separate noise explicitly. In terms of configuration simplicity, K-Means, K-Medoids, and Hierarchical Clustering are more straightforward, requiring only the number of clusters as input, whereas DBSCAN, HDBSCAN, and OPTICS involve more complex parameter setting.

Thus, HDBSCAN represents the most suitable method for this task: it identifies nearly all anomalies (recall of 0.99), yields a strong F1-score (0.92), and effectively isolates suspicious points via noise detection. However, if execution time is a critical constraint, K-Means offers a viable alternative, achieving a runtime of just 0.03 seconds and high precision (0.94), albeit with lower recall (0.69).

6. Discussion

The analysis demonstrated that the most effective clustering methods for separating network traffic into normal and anomalous categories are HDBSCAN and K-Means. These methods achieved high F1-scores in the range of 0.85-0.92 and exhibited low execution times (0.03-4 seconds), making them suitable for use in network intrusion detection systems (NIDS). K-Means stands out due to its simplicity and speed, but its sensitivity to initial conditions can result in instability. K-Medoids is more robust to outliers but suffers from slow performance on large datasets. Hierarchical clustering offers a high degree of interpretability, yet its computational complexity limits its scalability for large-scale applications.

Density-based methods such as DBSCAN and OPTICS identified a substantial proportion of data as noise, which complicates their direct application to binary classification tasks in network traffic analysis. In contrast, HDBSCAN achieved an optimal balance between clustering quality (F1-score of 0.923) and anomaly detection capability (recall of 0.988), while maintaining a rapid runtime. These results make HDBSCAN the most suitable clustering method for practical deployment in network intrusion detection systems.

In conclusion, for practical applications, HDBSCAN and K-Means are recommended, depending on the specific requirements for computational efficiency and robustness to noise.

7. Conclusion

This study conducted a comparative analysis of six clustering methods for the task of separating network traffic into normal and anomalous categories. Experimental results indicate that HDBSCAN is the most suitable method according to several criteria, including precision, recall, F1-score, and the ability to detect outliers. HDBSCAN achieved the highest F1-score (0.92) and recall (0.988), effectively identifying nearly all anomalies and handling noise robustly.

At the same time, the K-Means algorithm, having much lower computational complexity and execution time (0.03 seconds), is also an effective solution for time-sensitive applications. However, its sensitivity to the initial centroid selection and initial conditions, as well as its inability to detect noise stint its applicability. K-Medoids offers better robustness to outliers compared to K-Means, but its computational cost makes it less attractive for large-scale datasets.

Hierarchical clustering, OPTICS, and DBSCAN exhibit advantages such as the ability to detect clusters of arbitrary shape and considering noise. Nevertheless, their high computational complexity and sensitivity to parameter selection restrict their use in scenarios requiring rapid analysis of large datasets.

Future research directions include the development of hybrid approaches that combine the high accuracy and recall of density-based methods (e.g., HDBSCAN) with the speed and simplicity of partitioning-based methods (e.g., K-Means). Moreover, integrating clustering techniques with neural network architectures may further enhance the overall performance of anomaly detection systems.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, Aleksandr S. Baklashov and Dmitry S. Kulyabov; methodology, Dmitry S. Kulyabov; software, Aleksandr S. Baklashov; validation, Aleksandr S. Baklashov and Dmitry S. Kulyabov; formal analysis, Dmitry S. Kulyabov; investigation, Aleksandr S. Baklashov; resources, Dmitry S. Kulyabov; data curation, Aleksandr S. Baklashov; writing—original draft preparation, Aleksandr S. Baklashov; writing—review and editing, Aleksandr S. Baklashov and Dmitry S. Kulyabov; visualization, Aleksandr S. Baklashov; supervision, Dmitry S. Kulyabov.; project administration, Dmitry S. Kulyabov. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kosmacheva, I., Davidyuk, N., Belov, S., Kuchin, Y. S., Kvyatkovskaya, Y., Rudenko, M. & Lobeyko, V. I. Predicting of cyber attacks on critical information infrastructure. *Journal of Physics: Conference Series* **2091** (2021).
2. Bhuyan, M. H., Bhattacharyya, D. K. & Kalita, J. K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials* **16,** 303–336 (2014).
3. Schynol, L. & Pesavento, M. *Deep Unrolling for Anomaly Detection in Network Flows* in (Dec. 2023), 61–65. doi:10.1109/CAMSAP58249.2023.10403513.

4. Maheswari, G., Vinith, A., Sathyanarayanan, A. S., Sowmi, S. M. & Sambath, M. An Ensemble Framework for Network Anomaly Detection Using Isolation Forest and Autoencoders. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS),* 1–6 (2024).
5. Olateju, O., Okon, S., Igwenagu, U., Salami, A., Oladoyinbo, T. & Olaniyi, O. Combating the Challenges of False Positives in AI-Driven Anomaly Detection Systems and Enhancing Data Security in the Cloud. *Asian Journal of Research in Computer Science* **17,** 264–292. doi:10.9734/ajrcos/2024/v17i6472 (June 2024).
6. Lavanya, A. & Sekar, D. Traditional Methods and Machine Learning for Anomaly Detection in Self-Organizing Networks. *International Journal of Scientific Research in Science, Engineering and Technology* **10,** 352–360. doi:10.32628/IJSRSET2310662 (Dec. 2023).
7. Sheela, S. N., Prasad, E., Srinath, M. V. & Basha, M. S. Intrusion Detection Systems, Tools and Techniques – An Overview. *Indian journal of science and technology* **8** (2015).
8. Al-Ghamdi, M. An Assessment of Intrusion Detection System (IDS) and Data-Set Overview: A Comprehensive Review of Recent Works. *Journal of Scientific Research and Development* **5,** 979–982 (Feb. 2021).
9. Rozendaal, K., Mailewa, A. & Dissanayake Mohottalalage, T. Neural Network Assisted IDS/IPS: An Overview of Implementations, Benefits, and Drawbacks. *International Journal of Computer Applications* **184,** 21–28. doi:10.5120/ijca2022922098 (May 2022).
10. Satilmiş, H., Akleylek, S. & Tok, Z. A Systematic Literature Review on Host-Based Intrusion Detection Systems. *IEEE Access* **PP,** 1–1. doi:10.1109/ACCESS.2024.3367004 (Jan. 2024).
11. Mahfuz, N. M., Yusoff, M. & Ahmad, Z. Review of single clustering methods. *IAES International Journal of Artificial Intelligence* **8,** 221–227 (2019).
12. Burkov, A. *Machine learning engineering* (True Positive, Sept. 2020).
13. Park, H.-S. & Jun, C.-H. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications* **36,** 3336–3341. doi:10.1016/j.eswa.2008.01.039 (2009).
14. Campello, R., Kröger, P., Sander, J. & Zimek, A. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10.** doi:10.1002/widm.1343 (Oct. 2019).
15. Ankerst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. OPTICS: ordering points to identify the clustering structure. *SIGMOD Rec.* **28,** 49–60. doi:10.1145/304181.304187 (June 1999).
16. Sahli, Y. Comparison of the NSL-KDD dataset and its predecessor the KDD Cup '99 dataset. *International Journal of Scientific Research and Management* **10,** 832–839. doi:10.18535/ijsrm/v10i4.ec05 (Apr. 2022).
17. L.Dhanabal & Shantharajah, D. S. P. *A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms* in. **4** (June 2015), 446–452.
18. Kunhare, N. & Tiwari, R. *Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques* in (Nov. 2018), 127–131. doi:10.1109/CSNT.2018.8820244.
19. Gorban, A., Kégl, B., Wunsch, D. & Zinovyev, A. *Principal Manifolds for Data Visualisation and Dimension Reduction, LNCSE 58* 338 pp. (Jan. 2008).

Information about the authors

Aleksandr S. Baklashov—Master's degree student Department of Probability Theory and Cybersecurity of RUDN University; Mathematician, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (e-mail: 1132239133@pfur.ru, phone: +7(977)4573881, ORCID: 0009-0000-9046-3225, ResearcherID: KLZ-4503-2024)

Dmitry S. Kulyabov—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Probability Theory and Cyber Security of RUDN University; Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: kulyabov_ds@pfur.ru, phone: +7(495)9520250, ORCID: 0000-0002-0877-7063, ResearcherID: I-3183-2013, Scopus Author ID: 35194130800)

УДК 004.85
PACS 07.05.Tp,
DOI: 10.22363/2658-4670-2025-33-1-27-45 EDN: AFZDUC

Статистические и плотностные методы кластеризации в задачах обнаружения аномалий сетевых систем: сравнительный анализ

А. С. Баклашов[1, 2], Д. С. Кулябов[1, 3]

[1] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[2] Институт проблем управления им. В. А. Трапезникова Российской академии наук, ул. Профсоюзная, д. 65, Москва, 117997, Российская Федерация

[3] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. В современном мире количество данных, хранящихся в электронном виде и передающихся по сети, непрерывно растёт. Это создаёт потребность в разработке эффективных методов защиты информации, передающейся в виде сетевого трафика. Выявление аномалий играет ключевую роль в обеспечении безопасности сетей и защите информации от кибератак.

Цель данной работы заключается в проведении обзора статистических и плотностных методов кластеризации, применяемых для определения аномалий в сетевых системах, и проведении их сравнительного анализа на конкретной задаче.

Для достижения цели исследования использовались методы анализа существующих подходов к обнаружению аномалий с помощью методов кластеризации. В исследовании рассматривались различные алгоритмы и методы кластеризации, применяемые в сетевых системах.

Результаты проведённого сравнительного анализа продемонстрировали высокую эффективность методов кластеризации в задачах обнаружения аномалий сетевого трафика, что позволяет рекомендовать их для интеграции в системы обнаружения вторжений с целью повышения уровня информационной безопасности.

Был проведён сравнительный анализ различных методов, выявлены их общие черты, различия, достоинства и недостатки.

Полученные результаты могут быть использованы для улучшения систем обнаружения вторжений и повышения уровня защиты информации в сетевых системах.

Ключевые слова: системы обнаружения вторжений, сетевые системы, методы кластеризации

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 46–56
http://journals.rudn.ru/miph

Research article

UDC 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-46-56 EDN: BLFUDE

Symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices

Milena Veneva[1, 2]

[1] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russian Federation

[2] RIKEN Center for Computational Science, R-CCS, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

(received: November 8, 2024; revised: November 30, 2024; accepted: December 12, 2024)

Abstract. Systems of linear algebraic equations with multi-diagonal coefficient matrices may arise after many different scientific and engineering problems, as well as problems of the computational linear algebra where finding the solution of such a system of linear algebraic equations is considered to be one of the most important problems. This paper presents a generalised symbolic algorithm for solving systems of linear algebraic equations with multi-diagonal coefficient matrices. The algorithm is given in a pseudocode. A theorem which gives the condition for correctness of the algorithm is formulated and proven. Formula for the complexity of the multi-diagonal numerical algorithm is obtained.

Key words and phrases: numerical analysis, computational methods for sparse matrices, numerical mathematical programming methods, complexity of numerical algorithms

For citation: Veneva, M. Symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 46–56. doi: 10.22363/2658-4670-2025-33-1-46-56. edn: BLFUDE (2025).

1. Introduction

Systems of linear algebraic equations (SLAEs) with multi-diagonal coefficient matrices may arise after many different scientific and engineering problems, as well as problems of the computational linear algebra where finding the solution of a SLAE is considered to be one of the most important problems. For instance, the resultant SLAE after discretization of partial differential equations (PDEs), using finite difference methods (FDM) or finite element methods (FEM) has a banded coefficient matrix. The methods for solving such SLAEs known in the literature usually require the matrix to possess special characteristics so as the method to be numerically correct and stable, e. g. diagonal dominance, positive definiteness, etc. Another possible approach which ensures numerically correct formulae without adding special additional requirements or using pivoting is the symbolic algorithms.

By definition, a band SLAE is a SLAE with band coefficient matrix. The lower band width p is the number of sub-diagonals, the upper band width q is the number of super-diagonals, and the band width of the matrix is defined as $p + q + 1$ (we should add 1 because of the main diagonal), that is, the total number of non-zero diagonals in the matrix [1]. Here, we are going to focus on SLAEs with matrices for which $p = q = M$. The author of [2] presents a generalised numerical algorithm for solving multi-diagonal SLAEs with pivoting (implemented in `Fortran`) in the case $p \neq q$, and has applied it for solving boundary value problems discretized by finite difference approximations.

A whole branch of symbolic algorithms for solving systems of linear algebraic equations with different coefficient matrices exists in the literature. For instance, in [3] the author considers a tridiagonal matrix and a symbolic version of the Thomas method [4–7] is formulated. The authors of [8] build an algorithm in the case of a general bordered tridiagonal SLAE, while in [9] the coefficient matrix taken into consideration is a general opposite-bordered tridiagonal one.

A pentadiagonal coefficient matrix is of interest in [10], while a cyclic pentadiagonal coefficient matrix is considered in [11]. The latter algorithm can be applied to periodic tridiagonal and periodic pentadiagonal SLAE either by setting the corresponding matrix terms to be zero.

In [12] a symbolic method for the case of a cyclic heptadiagonal SLAEs is presented.

What is common for all these symbolic algorithms, is that they are implemented using Computer Algebra Systems (CASs) such as `Maple` [13], `Mathematica` [14], and `Matlab` [15].

Finally, [16] presents a symbolic method for the case of a pure heptadiagonal SLAE.

A performance analysis of symbolic methods (and numerical as well) for solving band matrix SLAEs (with three and five diagonals) being implemented in `C++` and run on modern (as of 2018) computer systems is made in [17]. Different strategies (symbolic included) for solving band matrix SLAEs (with three and five diagonals) are explored in [18]. A performance analysis of effective symbolic algorithms for solving band matrix SLAEs with coefficient matrices with three, five and seven diagonals being implemented in both `C++` and `Python` and run on modern (as of 2018) computer systems is made in [19].

Having in mind all these introductory notes, it is clear that a generalised multi-diagonal symbolic algorithm is the novelty that addresses the need of a direct method which solves multi-diagonal systems of linear algebraic equations without putting any requirements for the characteristics of the coefficient matrix. Thus, the aim of this paper, which is a logical continuation of [16–19], is to present such a generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. The symbolic algorithms investigated in [16–19] are actually particular cases of the generalised multi-diagonal symbolic method when $p = q = M = 1, 2,$ and 3.

The layout of the paper is as follows: in the next section, we outline the multi-diagonal numerical algorithm, and introduce the multi-diagonal symbolic algorithm in pseudocode. Afterwards, we make some correctness remarks for the symbolic method, and present a generalised formula for the complexity of the multi-diagonal numerical algorithm. Finally, some conclusions are drawn.

The novelties of this work are as follows: suggested multi-diagonal symbolic algorithm for solving SLAEs, formulation and proof of a correctness theorem, and an additionally obtained formula for the complexity of the multi-diagonal numerical method.

2. Multi-diagonal symbolic algorithm

Let us consider a SLAE $Ax = y$, where $A = \mathrm{diag}(\mathbf{b^0}, \mathbf{b^1}, \mathbf{b^2}, \dots, \mathbf{b^M}, \mathbf{b^{M+1}}, \mathbf{b^{M+2}}, \mathbf{b^{M+3}}, \dots, \mathbf{b^{2\times M}})$, A is a real $N \times N$ multi-diagonal matrix with M sub-diagonals, and M super-diagonals, and $2 \times M + 1 < N$, that is, the number of diagonals should be smaller than the number of equations within the SLAE; x

and y are real column vectors with N elements:

$$\begin{bmatrix}
b_0^M & b_0^{M+1} & b_0^{M+2} & b_0^{M+3} & \dots & b_0^{2M} & 0 & \dots & \dots & \dots & \dots & 0 \\
b_1^{M-1} & b_1^M & b_1^{M+1} & b_1^{M+2} & \dots & b_1^{2M-1} & b_1^{2M} & 0 & \dots & \dots & \dots & 0 \\
b_2^{M-2} & b_2^{M-1} & b_2^M & b_2^{M+1} & \dots & b_2^{2M-2} & b_2^{2M-1} & b_2^{2M} & 0 & \dots & \dots & 0 \\
b_3^{M-3} & b_3^{M-2} & b_3^{M-1} & b_3^M & \dots & b_3^{2M-3} & b_3^{2M-2} & b_3^{2M-1} & b_3^{2M} & 0 & \dots & 0 \\
b_4^{M-4} & b_4^{M-3} & b_4^{M-2} & b_4^{M-1} & \dots & b_4^{2M-4} & b_4^{2M-3} & b_4^{2M-2} & b_4^{2M-1} & b_4^{2M} & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & b_{N-4}^0 & b_{N-4}^1 & \dots & \dots & b_{N-4}^{M-1} & b_{N-4}^M & b_{N-4}^{M+1} & b_{N-4}^{M+2} & b_{N-4}^{M+3} \\
0 & \dots & \dots & 0 & b_{N-3}^0 & b_{N-3}^1 & \dots & \dots & b_{N-3}^{M-1} & b_{N-3}^M & b_{N-3}^{M+1} & b_{N-3}^{M+2} \\
0 & \dots & \dots & \dots & 0 & b_{N-2}^0 & b_{N-2}^1 & \dots & \dots & b_{N-2}^{M-1} & b_{N-2}^M & b_{N-2}^{M+1} \\
0 & \dots & \dots & \dots & \dots & 0 & b_{N-1}^0 & b_{N-1}^1 & \dots & \dots & b_{N-1}^{M-1} & b_{N-1}^M
\end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{N-4} \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \end{bmatrix}
=
\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-4} \\ y_{N-3} \\ y_{N-2} \\ y_{N-1} \end{bmatrix}.$$

The multi-diagonal numerical solver which we are going to formulate below is a generalization of the Thomas method for multi-diagonal SLAEs. The algorithm is based on LU decomposition, and requires forward reduction for reducing the initial matrix into a lower triangular one:

$$\begin{aligned}
&\mu_0 = b_0^M \\
&\alpha_0^{M+1} = \frac{b_0^{M+1}}{\mu_0} \\
&\alpha_0^{M+2} = \frac{b_0^{M+2}}{\mu_0} \\
&\dots \\
&\alpha_0^{2M} = \frac{b_0^{2M}}{\mu_0} \\
&z_0 = \frac{y_0}{\mu_0}
\end{aligned}$$

$$\begin{aligned}
&\alpha_1^j = 0, \quad i = 1, 2 \dots, M-1 \\
&\alpha_1^M = b_1^{M-1} \\
&\mu_1 = b_1^M - \alpha_0^{M+1} \times \alpha_1^M \\
&\alpha_1^{M+1} = \frac{b_1^{M+1} - \alpha_0^{M+2} \times \alpha_1^M}{\mu_1} \\
&\alpha_1^{M+2} = \frac{b_1^{M+2} - \alpha_0^{M+3} \times \alpha_1^M}{\mu_1} \\
&\dots \\
&\alpha_1^{2M} = \frac{b_1^{2M}}{\mu_1} \\
&z_1 = \frac{y_1 - z_0 \times \alpha_1^M}{\mu_1}
\end{aligned}$$

$$\begin{aligned}
&\text{For } i = 2, 3, \dots, M-1 \ : \\
&\text{counter} = M - i \\
&\alpha_i^{k-\text{counter}} = b_i^{k-1}, \quad k = M, M-1, \dots, 1, \quad k - \text{counter} \geq 1 \\
&\alpha_i^{k-\text{counter}} = \alpha_i^{k-\text{counter}} - \alpha_0^{M+k-\text{counter}-1} \times \alpha_i^1 - \alpha_1^{M+k-\text{counter}-2} \times \alpha_i^2 - \dots \\
&\qquad\qquad - \alpha_{k-\text{counter}-2}^{M+1} \times \alpha_i^{k-\text{counter}-1}, \quad k = 2, 3, \dots, M \\
&\mu_i = b_i^M - \alpha_0^{M+i} \times \alpha_i^1 - \alpha_1^{M+i-1} \times \alpha_i^2 - \cdots - \alpha_{i-1}^{M+1} \times \alpha_i^i \\
&\alpha_i^{M+1} = \frac{b_i^{M+1} - \alpha_0^{M+i+1} \times \alpha_i^1 - \alpha_1^{M+i} \times \alpha_i^2 - \cdots - \alpha_{i-1}^{M+2} \times \alpha_i^i}{\mu_i} \\
&\text{num\_sub} = min(i, M-2) \quad \text{number of subtractions for } \alpha_i^{M+2}
\end{aligned}$$

$$\alpha_i^{M+2} = \frac{1}{\mu_i} \Big(b_i^{M+2} - \alpha_{i-\text{num\_sub}}^{M+2+\text{num\_sub}} \times \alpha_i^{i-\text{num\_sub}+1}$$

$$-\alpha_{i-\text{num\_sub}+1}^{M+2+\text{num\_sub}-1} \times \alpha_i^{i-\text{num\_sub}+2} - \dots \alpha_{i-1}^{M+3} \times \alpha_i^{i}\Big)$$

$$\dots$$

$$\text{num\_sub} = min(i, M-1-(k-1)) \quad \text{number of subtractions for } \alpha_i^{M+k}$$

$$\dots$$

$$\alpha_i^{2M} = \frac{b_i^{2M}}{\mu_i}$$

$$z_i = \frac{y_i - z_0 \times \alpha_i^1 - z_1 \times \alpha_i^2 - \dots - z_{i-1} \times \alpha_i^i}{\mu_i}$$

For $i = M, M+1, \dots, N-1$:

$$\alpha_i^j = b_i^{j-1}, \quad j = 1, 2 \dots, M-1$$

$$\alpha_i^k = \alpha_i^k - \alpha_{i-M}^{M+k-1} \times \alpha_i^1 - \alpha_{i-M+1}^{M+k-2} \times \alpha_i^2 - \dots - \alpha_{i-M+\text{iter}}^{M+k-1-\text{iter}} \times \alpha_i^{1+\text{iter}},$$
$$k = 2, 3, \dots, M, \quad \text{iter} = 0, 1, \dots, k-2, \quad i = M, M+1 \dots, N-1$$

$$\mu_i = b_i^M - \alpha_{i-M}^{2M} \times \alpha_i^1 - \alpha_{i-M+1}^{2M-1} \times \alpha_i^2 - \dots - \alpha_{i-1}^{M+1} \times \alpha_i^M,$$
$$i = M, M+1, \dots, N-1$$

$$\alpha_i^{M+1} = \frac{b_i^{M+1} - \alpha_{i-1}^{M+2} \times \alpha_i^M - \alpha_{i-2}^{M+3} \times \alpha_i^{M-1} - \alpha_{i-3}^{M+4} \times \alpha_i^{M-2} - \dots - \alpha_{i-M+1}^{2M} \times \alpha_i^2}{\mu_i},$$
$$i = M, M+1, \dots, N-2$$

$$\alpha_i^{M+2} = \frac{b_i^{M+2} - \alpha_{i-1}^{M+3} \times \alpha_i^M - \alpha_{i-2}^{M+4} \times \alpha_i^{M-1} - \alpha_{i-3}^{M+5} \times \alpha_i^{M-2} - \dots - \alpha_{i-M+2}^{2M} \times \alpha_i^3}{\mu_i},$$
$$i = M, M+1, \dots, N-3$$

$$\dots$$

$$\alpha_i^{2M} = \frac{b_i^{2M}}{\mu_i}, \qquad i = M, M+1, \dots, N-1-M$$

$$z_i = \frac{y_i - z_{i-M} \times \alpha_i^1 - z_{i-M+1} \times \alpha_i^2 - \dots - z_{i-1} \times \alpha_i^M}{\mu_i}, \qquad i = M, M+1, \dots, N-1$$

and a backward substitution for finding the unknowns x in a reverse order:

$$x_{N-1} = z_{N-1}$$

$$x_{N-k} = z_{N-k} - \alpha_{N-k}^{M+1} \times x_{N-k+1} - \dots - \alpha_{N-k}^{M+1+k-2} \times x_{N-1}, \quad k = 2, 3, \dots, M$$

$$x_i = z_i - \alpha_i^{M+1} \times x_{i+1} - \alpha_i^{M+2} \times x_{i+2} - \alpha_i^{M+3} \times x_{i+3} - \dots - \alpha_i^{2M} \times x_{i+M},$$
$$i = N-(M+1), N-(M+2), \dots, 0.$$

In order to cope with the stability issue of the Thomas method in the case of non-diagonally dominant matrices, in the case of a zero (or numerically zero) quotient of two subsequent leading principal minors within the symbolic method a symbolic variable is assigned instead and the calculations are continued. At the end of the algorithm, this symbolic variable is substituted with zero. The same approach is suggested in [3].

The full multi-diagonal symbolic method in pseudocode is given in Algorithm 1. There, ε plays the role of a numerical zero, and was set to 1.0e−20 in our code.

Algorithm 1: Multi-diagonal symbolic algorithm for solving a SLAE $Ax = y$.

```
Input: $N, \mathbf{b^0}, \mathbf{b^1}, \dots, \mathbf{b^M}, \mathbf{b^{M+1}}, \dots, \mathbf{b^{2M}}, \mathbf{y}, \varepsilon$
Output: $\mathbf{x}$
if $det(A) == 0$ then
    exit
end if
bool flag = False
$\mu_0 = b_0^M$                                              ▷ Step 1.(0)
if $|\mu_0| < \varepsilon$ then
    $\mu_0$ = symb; flag = True
end if
for $k = \overline{M+1,\dots,2M}$ do
    $\alpha_0^k = \dfrac{b_0^k}{\mu_0}$
end for
$z_0 = \dfrac{y_0}{\mu_0}$
for $k = \overline{1,2,\dots,M}$ do
    $\alpha_1^k = b_1^{k-1}$
end for                                                      ▷ (1)
$\mu_1 = b_1^M - \alpha_0^{M+1} \times \alpha_1^M$
if !flag then
    if $|\mu_1| < \varepsilon$ then
        $\mu_1$ = symb; flag = True
    end if
end if
for $k = \overline{M+1,\dots,2M}$ do
    $\alpha_1^k = b_1^k$
    if $M > 1$ and $k < 2M$ then
        $\alpha_1^k = \alpha_1^k - \alpha_0^{k+1} \times \alpha_1^M$
    end if
    $\alpha_1^k = \dfrac{\alpha_1^k}{\mu_1}$
end for
$z_1 = \dfrac{y_1 - z_0 \times \alpha_1^M}{\mu_1}$
for $i = \overline{2,\dots,N-1}$ do
    counter = 0 ▷ number of non-zero helping $\alpha_1^k$,
                                           ▷ where $k = \overline{1,2,\dots,M}$
    if $i < M$ then
        counter = $M - i$
    end if
    for $k = \overline{1,\dots,M}$ do
        if $k$ − counter ≥ 1 then
            $\alpha_i^{k-\text{counter}} = b_i^{k-1}$
        end if
    end for
    ▷ above we shift the non-zero $\alpha_i^j, j \le M$ in order
          ▷ to have them in the interval $j \in [0;\dots]$
    for $k = \overline{M+1,\dots,2M}$ do
        $\alpha_i^k = b_i^k$
    end for
    $\mu_i = b_i^M$
    $z_i = y_i$
    iter = 0
      ▷ number of iterations for $\alpha_i^k$, where $k \le M$
    coeff = 0
▷ the biggest distance between the lower coeff of
                                  ▷ $\alpha_i^k$ and $\alpha_{\text{coeff+iter}}^{M+\dots}$
    if $i >= M$ then
        coeff = $i - M$
    end if
    for $k = \overline{2,\dots,M}$ do
        iter = 0
        for $l = \overline{2,\dots,k-\text{counter}}$ do
            $\alpha_i^{k-\text{counter}} = \alpha_i^{k-\text{counter}}$
                $-\alpha_{\text{coeff+iter}}^{M+k-1-\text{counter}-\text{iter}} \times \alpha_i^{1+\text{iter}}$
            iter = 1 + iter
        end for
    end for
    iter = 0
                  ▷ number of iterations for $\mu_i$ and $z_i$
    if $i < M$ then
        mu_max_iter = $i - 1$
    else
        mu_max_iter = $M - 1$
    end if
    for iter = $\overline{0,1,\dots,\text{mu\_max\_iter}}$ do        ▷ $\mu_i, z_i$
        $\mu_i = \mu_i - \alpha_{\text{coeff+iter}}^{2M-\text{counter}-\text{iter}} \times \alpha_i^{1+\text{iter}}$
        $z_i = z_i - z_{\text{coeff+iter}} \times \alpha_i^{1+\text{iter}}$
    end for
    if !flag then
        if $|\mu_i| < \varepsilon$ then
            $\mu_i$ = symb; flag = True
        end if
    end if
    $z_i = \dfrac{z_i}{\mu_i}$
    iter = 0                          ▷ number of iterations for a
                                      ▷ particular $\alpha_i^m, m \ge M+1$
    alpha_counter = 0                            ▷ number of
$\alpha_i^m, m \ge M+1$
    for $m = \overline{0,1,\dots,M-1}$ do
        num_sub[$m$] = 0
                              ▷ number of subtractions in $\alpha_i^m$
    end for
    for $m = \overline{M+1,\dots,2M-1}$ do
        m_index = $m - M - 1$
                              ▷ shift the index from 0 to $M-2$
        if $i \le M-1$ then
```

```
            num_sub[m_index] =
                $min(i, M - 1 -$ alpha_counter$)$
        else
            num_sub[m_index] =
                $M - 1 -$ alpha_counter
        end if
        iter = 0
        for $k = \overline{0, 1, \dots, \text{num\_sub[m\_index]} - 1}$ do
            coeff = $i -$ num_sub[m_index] + iter
            coeff_1 = 0
            if $i \geq M$ then
                coeff_1 = ($M$ − num_sub[m_index]
                                +iter)%$M$ + 1
            else
                coeff_1 = $i$ − num_sub[m_index]
                                +iter + 1
            end if
            ▷ the helping $\alpha_i^{\text{coeff\_1}}$ are with upper index
            ▷ up to $M$, therefore we need to find the
                                            ▷ module($M$)
            $\alpha_i^m = \alpha_i^m$
                $-\alpha_{\text{coeff}}^{m+\text{num\_sub[m\_index]}-\text{iter}} \times \alpha_i^{\text{coeff\_1}}$
            iter = iter + 1
        end for
        alpha_counter = alpha_counter + 1
        for $k = \overline{M + 1, \dots, 2M}$ do
            $\alpha_i^k = \frac{\alpha_i^k}{\mu_i}$
        end for
    end for
end for
$x_{N-1} = z_{N-1}$                              ▷ Step 2. Solution
for $i = \overline{N - 2, \dots 0}$ do
    $x_i = z_i$
    iter = 0
    for $k = \overline{0, \dots M - 1}$ do
        if $i$ + iter > $n - 2$ then
            break
        end if
        $x_i = x_i - \alpha_i^{M+1+k} \times x_{i+1+k}$
        iter = iter + 1
    end for
end for
Cancel the common factors in the numerators and
denominators of $\mathbf{x}$, making them coprime. Substi-
tute symb := 0 in $\mathbf{x}$ and simplify.
```

Remark: If any μ_i expression has been evaluated to be zero or numerically zero, then it is assigned to be a symbolic variable. We cannot compare any of the next μ_i expressions with ε, because any further μ_i is going to be a symbolic expression. To that reason, we use a boolean flag which tells us if any previous μ_i is a symbolic expression. In that case, comparison with ε is not conducted as being not needed.

3. Justification of the algorithm

Let us make some observations on the correctness of the proposed algorithm. In case the algorithm assigns μ_i for any $i = \overline{0, N-1}$ to be equal to a symbolic variable (in case μ_i is zero or numerically zero), this ensures correctness of the formulae for computing the solution of the considered SLAE (because we are not dividing by (numerical) zero). However, this does not add any additional requirements to the coefficient matrix so as to keep the algorithm stable.

Theorem 2. *The only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct is nonsingularity.*

Proof. As a direct consequence of the transformations done so as the matrix A to be factorized and then the downwards sweep to be conducted, it follows that the determinant of the matrix A in the terms of the introduced notation is:

$$\det(A) = \prod_{i=0}^{N-1} \mu_i|_{\text{symb}=0},$$

because the determinant of an upper triangular matrix is equal to the product of all its diagonal elements [20]. (This formula could be used so as the nonsingularity of the coefficient matrix to be checked.) If μ_i for any i is assigned to be equal to a symbolic variable, then it is going to appear in both the numerator and the denominator of the expression for the determinant and so it can be cancelled:

$$\begin{aligned}\det(A) &= \mu_0\,\mu_1\,\mu_2 \dots \mu_{N-2}\,\mu_{N-1} = M_0\,\frac{M_1}{\mu_0}\,\frac{M_2}{\mu_0\,\mu_1}\dots\frac{M_{N-2}}{\mu_0\,\mu_1 \dots \mu_{N-3}}\,\frac{M_{N-1}}{\mu_0\,\mu_1 \dots \mu_{N-2}} = \\ &= \frac{\prod_{i=0}^{N-1} M_i}{\mu_0^{N-1}\,\mu_1^{N-2}\,\mu_2^{N-3}\dots\mu_{N-3}^{2}\,\mu_{N-2}^{1}} = \frac{\prod_{i=0}^{N-1} M_i}{M_0^{N-1}\,\frac{M_1^{N-2}}{\mu_0^{N-2}}\,\frac{M_2^{N-3}}{\mu_0^{N-3}\,\mu_1^{N-3}}\dots\frac{M_{N-3}^{2}}{\mu_0^2\,\mu_1^2\dots\mu_{N-4}^2}\,\frac{M_{N-2}^{1}}{\mu_0^1\,\mu_1^1\dots\mu_{N-3}^1}} = \\ &= \frac{\prod_{i=0}^{N-1} M_i}{M_0^{N-1}\,\frac{M_1^{N-2}}{\mu_0^{N-2}}\,\frac{M_2^{N-3}}{\mu_0^{N-3}\left(\frac{M_1}{\mu_0}\right)^{N-3}}\dots\frac{M_{N-3}^{2}}{\mu_0^2\left(\frac{M_1}{\mu_0}\right)^2\dots\left(\frac{M_{N-4}}{\mu_{N-3}}\right)^2}\,\frac{M_{N-2}^{1}}{\mu_0^1\left(\frac{M_1}{\mu_0}\right)^1\dots\left(\frac{M_{N-3}}{\mu_{N-4}}\right)^1}} = \frac{\prod_{i=0}^{N-1} M_i}{\prod_{i=0}^{N-2} M_i} = M_{N-1}\end{aligned}$$

where M_i is the i-th leading principal minor, and $\mu_0 = M_0$. This means that the only constraint on the coefficient matrix is $M_{N-1} \neq 0$. □

Remark: above, we have used the following recurrent formula $M_i = \prod_{i=0}^{i} \mu_i$.

Remark: this theorem coincides with the theorem we have proven in [16], because no matter what the number of diagonals ($2 \times M + 1$) within the coefficient matrix is, the logic remains.

The requirement on the coefficient matrix to be nonsingular is not limiting at all since this is a standard requirement so as the SLAE to have only one solution.

3.1. Number of computational steps

The calculation of $\alpha_i^k, \mu_i, \alpha_i^{M+1}, \alpha_i^{M+2}, \dots, \alpha_i^{2M}$, and z_i depends on the results of the calculation of α_{i-j}^{M+k}, and z_{i-j}. On the other hand, the calculation of x_i depends on the results of the calculation of $\alpha_i^{M+1}, \alpha_i^{M+2}, \dots, \alpha_i^{2M}, z_i$, and $x_{i+1}, x_{i+2}, \dots, x_{i+M}$. This makes the multi-diagonal numerical method inherently serial. It takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

3.2. Complexity

The amount of operations per expression are summarized in Table 1. Thus, the overall complexity of the multi-diagonal numerical algorithms is:

$$2NM^2 + 5NM + N - \frac{4M^3}{3} - \frac{7M^2}{2} - \frac{13M}{6},$$

where N is the number of rows in the initial coefficient matrix. Hence, the multi-diagonal numerical method requires only $O(N)$ operations (provided that $M << N$) for finding the solution, and beats the Gaussian elimination which requires $O(N^3)$ operations.

4. Results

Within this paper we formulated the multi-diagonal numerical solver which is a generalization of the Thomas method for multi-diagonal SLAEs. In Algorithm 1, we introduced the pseudocode of the the generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices.

It was proven that the only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct is nonsingularity.

The multi-diagonal numerical method takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

Table 1

Complexity per expression for the multi-diagonal numerical algorithm

expression	# operations	simplified form of # ops	examples		
			$M = 2$	$M = 3$	$M = 4$
$\alpha_i^k, i < M,$ $k = 2, \dots, M$	$\sum_{k=1}^{M-1}(2 \times (1 + 2 + \dots k - 1)) =$ $\sum_{k=1}^{M-1}\left(2 \times \frac{(k-1) \times k}{2}\right)$	$\frac{(M-1) \times M \times (2M-1)}{6}$ $-\frac{(M-1) \times M}{2}$	0	2	8
$\alpha_i^k, i \geq M,$ $k = 2, \dots, M$	$(N - 1 - M + 1)$ $\times \sum_{k=1}^{M}((k-1) \times 2)$	$(N - M) \times (M^2 - M)$	$2(N-2)$	$6(N-3)$	$12(N-4)$
$\mu_i, i < M$	$\sum_{k=0}^{M-1}(2k)$	$M^2 - M$	2	6	12
$\mu_i, i \geq M$	$(N - 1 - M + 1) \times 2M$	$2MN - 2M^2$	$4N - 8$	$6N - 18$	$8N - 32$
$\alpha_i^{M+k}, i < M,$ $k = 1, 2, \dots, M$	$\sum_{k=1}^{M}\Big(\sum_{i=0}^{M-k}(i \times 2 + 1)$ $+(k-1) \times ((M-k) \times 2 + 1)\Big)$	M^3 $-\frac{M \times (M+1) \times (2M+1)}{6}$ $+\frac{M \times (M+1)}{2}$	6	19	44
$\alpha_i^{M+k}, i \geq M,$ $k = 1, 2, \dots, M$	$\sum_{k=1}^{M}((N - (M + k))$ $\times((M-k) \times 2 + 1))$	$NM^2 - 2M^3 - M^2$ $+\frac{M \times (M+1) \times (2M+1)}{3}$ $-\frac{M \times (M+1)}{2}$	$4N - 13$	$9N - 41$	$16N - 94$
$z_i, i < M$	$\sum_{k=0}^{M-1}(2k+1)$	M^2	4	9	16
$z_i, i \geq M$	$(N - 1 - M + 1) \times (2M + 1)$	$2NM + N - M - 2M^2$	$5N - 10$	$7N - 21$	$9N - 36$
$x_{N-k},$ $k = 1, \dots, M$	$\sum_{k=1}^{M}((k-1) \times 2)$	$M^2 - M$	2	6	12
$x_{N-k},$ $k = M + 1, \dots, N$	$(N - (M + 1) + 1) \times 2M$	$2NM - 2M^2$	$4N - 8$	$6N - 18$	$8N - 32$
Total	$2NM^2 + 5NM + N - \frac{4M^3}{3} - \frac{7M^2}{2} - \frac{13M}{6}$		$19N - 29$	$34N - 74$	$53N - 150$

The complexity of the multi-diagonal numerical algorithms was found to be:

$$2NM^2 + 5NM + N - \frac{4M^3}{3} - \frac{7M^2}{2} - \frac{13M}{6},$$

where N is the number of rows in the initial coefficient matrix. Hence, the multi-diagonal numerical method requires $O(N)$ operations (provided that $M << N$) for finding the solution. The amount of operations per expression were summarized in Table 1. In the Table 1 one can also find the complexity per expression in the cases when $M = 2$, $M = 3$, and $M = 4$.

5. Discussion

Within this paper we formulated the multi-diagonal numerical solver which is a generalization of the Thomas method for multi-diagonal SLAEs. Next, we introduced the pseudocode of the generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. There, as a remedy of the stability issue which arises within the Thomas method in the case of non-diagonally dominant matrices, if we obtain a zero (or numerically zero) quotient of two subsequent leading principal minors, a symbolic variable is assigned instead and the calculations are continued. At the end of the algorithm, this symbolic variable is substituted with zero. The generalised multi-diagonal symbolic algorithm is the novelty that addresses the need of a direct method which solves multi-diagonal SLAEs without putting any requirements for the characteristics of the coefficient matrix. This algorithm is a generalization of the algorithms presented in [3, 10, 16].

It was proven that the only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct is nonsingularity. Asking for nonsingularity of the coefficient matrix is a standard requirement so as the SLAE to have only one solution. Hence, this does not limit the significance of the formulated symbolic algorithm.

The multi-diagonal numerical method takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

The multi-diagonal numerical method requires $O(N)$ operations (provided that $M << N$) for finding the solution, and beats the Gaussian elimination which requires $O(N^3)$ operations.

6. Conclusion

A generalised symbolic algorithm for solving systems of linear algebraic equations with multi-diagonal coefficient matrices was formulated and presented in pseudocode. Some notes on the correctness of the algorithm were made. Formula for the complexity of the multi-diagonal numerical algorithm was obtained.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Acknowledgments: The author would like to thank Dr. Alexander Ayriyan (JINR), and Prof. Toshiyuki Imamura (R-CCS, RIKEN) for their precious comments.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Spiteri, R. J. *Notes in Numerical Analysis I. Chapter 2* (University of Saskatchewan, 2007).
2. Christov, C. I. Gaussian elimination with pivoting for multidiagonal systems. *University of Reading, Department of Mathematics, Numerical Analysis Report 5/94* (1994).
3. El-Mikkawy, M. A Generalized Symbolic Thomas Algorithm. *Applied Mathematics* **3,** 342–345. doi:10.4236/am.2012.34052 (2012).
4. Thomas, L. H. *Elliptic Problems in Linear Difference Equations over a Network* (Watson Sci. Comput. Lab. Rept., Columbia University, 1949).
5. Higham, N. J. *Accuracy and Stability of Numerical Algorithms* 2nd ed. 710 pp. (SIAM, 2002).
6. Samarskii, A. A. & Nikolaev, E. S. *Metody Resheniya Setochnyh Uravnenii* Russian (Nauka, 1978).
7. Samarskii, A. A. & Nikolaev, E. S. *Numerical Methods for Grid Equations. Direct Methods* (Birkhäuser Basel, 1989).
8. Karawia, A. A. & Rizvi, Q. M. On solving a general bordered tridiagonal linear system. *International Journal of Mathematical Sciences* **33** (2013).
9. Atlan, F. & El-Mikkawy, M. A new symbolic algorithm for solving general opposite-bordered tridiagonal linear systems. *American Journal of Computational Mathematics* **5,** 258–266. doi:10.4236/ajcm.2015.53023 (2015).
10. Askar, S. S. & Karawia, A. A. On Solving Pentadiagonal Linear Systems via Transformations. *Mathematical Problems in Engineering. Hindawi Publishing Corporation.* **9.** doi:10.1155/2015/232456 (2015).

11. Jia, J.-T. & Jiang, Y.-L. Symbolic algorithm for solving cyclic penta-diagonal linear systems. *Numerical Algorithms* **63,** 357–367. doi:10.1007/s11075-012-9626-2 (2012).
12. Karawia, A. A. A new algorithm for general cyclic heptadiagonal linear systems using Sherman-Morrisor-Woodbury formula. *ARS Combinatoria* **108,** 431–443 (2013).
13. Bernardin, L. e. a. *Maple Programming Guide* (Maplesoft, a division of Waterloo Maple Inc., 1996-2021.).
14. Wolfram Research, I. *System Modeler* (Version 13.2, Champaign, IL, 2022).
15. Higham, D. J. & Higham, N. J. *MATLAB guide* (SIAM, 2016).
16. Veneva, M. & Ayriyan, A. Symbolic Algorithm for Solving SLAEs with Heptadiagonal Coefficient Matrices. *Mathematical Modelling and Geometry* **6,** 22–29 (2018).
17. Veneva, M. & Ayriyan, A. *Performance Analysis of Effective Methods for Solving Band Matrix SLAEs after Parabolic Nonlinear PDEs* in *Advanced Computing in Industrial Mathematics, Revised Selected Papers of the 12th Annual Meeting of the Bulgarian Section of SIAM, December 20–22, 2017, Sofia, Bulgaria* (Studies in Computational Intelligence, 2019), 407–419. doi:10.1007/978-3-319-97277-0_33.
18. Veneva, M. & Ayriyan, A. Effective Methods for Solving Band SLAEs after Parabolic Nonlinear PDEs. *AYSS-2017, European Physics Journal – Web of Conferences (EPJ-WoC)* **177** (2018).
19. Veneva, M. & Ayriyan, A. *Performance Analysis of Effective Symbolic Methods for Solving Band Matrix SLAEs* in *23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018)* **214** (European Physics Journal – Web of Conferences (EPJ-WoC), 2019).
20. Kaye, R. & Wilson, R. *Linear Algebra* 242 pp. (Oxford University Press, 1998).

Information about the authors

Milena Veneva—Master of Sciences in Applied Mathematics, (e-mail: milena.p.veneva@gmail.com, ORCID: 0000-0002-6421-4716)

УДК 519.872, 519.217

PACS 07.05.Tp, 02.60.Pn, 02.70.Bf

DOI: 10.22363/2658-4670-2025-33-1-46-56 EDN: BLFUDE

Символьный алгоритм решения СЛАУ с многодиагональными матрицами коэффициентов

Милена Венева[1, 2]

[1] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

[2] Центр вычислительной науки RIKEN (R-CCS), Минатодзима-Минами-мачи 7-1-26, район Тюо-ку, Кобе, префектура Хиого, 650-0047, Япония

Аннотация. Системы линейных алгебраических уравнений с многодиагональными матрицами коэффициентов возникают во многих прикладных и теоретических задачах науки и техники, а также в задачах вычислительной линейной алгебры, где их решение представляет собой одну из ключевых проблем. В данной работе представлен обобщённый символьный алгоритм решения систем линейных алгебраических уравнений с многодиагональными матрицами коэффициентов. Алгоритм приведён в виде псевдокода. Сформулирована и доказана теорема, определяющая условие корректности алгоритма. Получена формула, описывающая вычислительную сложность соответствующего численного алгоритма для многодиагональных систем.

Ключевые слова: численные методы, вычислительные методы для разреженных матриц, методы численного математического программирования, вычислительная сложность численных алгоритмов

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 57–73
http://journals.rudn.ru/miph

Research article

UDC 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-57-73

EDN: ABHFKC

On the stable approximate solution of the ill-posed boundary value problem for the Laplace equation with homogeneous conditions of the second kind on the edges at inaccurate data on the approximated boundary

Evgeniy B. Laneev, Alexander V. Klimishin

RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

(received: November 20, 2024; revised: December 2, 2024; accepted: December 12, 2024)

Abstract. In this paper, we consider the ill-posed continuation problem for harmonic functions from an ill-defined boundary in a cylindrical domain with homogeneous boundary conditions of the second type on the side faces. The value of the function and its normal derivative (Cauchy conditions) is known approximately on an approximated surface of arbitrary shape bounding the cylinder. In this case, the Cauchy problem for the Laplace equation has the property of instability with respect to the error in the Cauchy data, that is, it is ill-posed. On the basis of an idea about the source function of the original problem, the exact solution is represented as a sum of two functions, one of which depends explicitly on the Cauchy conditions, and the second one can be obtained as a solution of the Fredholm integral equation of the first kind in the form of Fourier series on the eigenfunctions of the second boundary value problem for the Laplace equation. To obtain an approximate stable solution of the integral equation, the Tikhonov regularization method is applied when the solution is obtained as an extremal of the Tikhonov functional. For an approximated surface, we consider the calculation of the normal to this surface and its convergence to the exact value depending on the error with which the original surface is given. The convergence of the obtained approximate solution to the exact solution is proved when the regularization parameter is compared with the errors in the data both on the inexactly specified boundary and on the value of the original function on this boundary. A numerical experiment is carried out to demonstrate the effectiveness of the proposed approach for a special case, for a flat boundary and a specific initial heat source (a set of sharpened sources).

Key words and phrases: ill-posed problem, Tikhonov regularization method, Cauchy problem for the Laplace equation, integral equation of the first kind

For citation: Laneev, E. B., Klimishin, A. V. On the stable approximate solution of the ill-posed boundary value problem for the Laplace equation with homogeneous conditions of the second kind on the edges at inaccurate data on the approximated boundary. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 57–73. doi: 10.22363/2658-4670-2025-33-1-57-73. edn: ABHFKC (2025).

Introduction

Among non-invasive diagnostic methods [1], thermal imaging stands out for its efficiency and accuracy with proper data processing. When carried out with the help of a thermal imager, a thermogram of the surface of the object can be obtained, showing the heat distribution on the surface of the object under investigation in the infrared range.

Corrected for various interferences in heat exchange processes and surface inhomogeneities of the observed object, the thermogram image conveys the structure of the heat-generating object, which makes it possible to assess various abnormalities in the state of the patient's internal organs during medical diagnosis.

The paper proposes a method for correcting the image on the thermogram based on a mathematical model that considers a homogeneous heat-conducting body in the form of a rectangular cylinder containing heat sources with a distribution density function independent of time, bounded by an arbitrary surface S on which a boundary condition of the third kind corresponding to convective heat exchange with the medium is set. The proposed model considers the case when there is no heat exchange on the lateral faces of the body — homogeneous boundary conditions of the second kind take place.

As a result of processing, the temperature distribution function on the plane (corrected thermogram) near the heat sources is constructed as a result of continuation of the temperature distribution from an arbitrary surface from which the original thermogram is taken. The corrected thermogram more accurately conveys the structure of the heat sources than the image on the original thermogram.

When obtaining the corrected thermogram as a result of processing, approaches similar to the continuation of gravitational fields in geophysics problems were used [2–4].

To obtain the result, the inverse problem to the mixed boundary value problem for the Poisson equation is solved, since the goal is to obtain information about inaccessible heat sources from data on a given surface. The inverse problem is ill-posed, because small errors in the initial data (in the initial thermogram, in the data on the surface) can lead to significant distortions of the result. To construct its stable approximate solution, the Tikhonov regularisation method is used [5–7].

1. Problem statement

Physical model: we consider a homogeneous heat-conducting body in the form of a rectangular cylinder bounded by the surface S and containing heat sources with a distribution density function independent of time.

These heat sources create a stationary temperature distribution in the body.

The object of study is the density function of heat source distribution.

On the surface S there is convective heat exchange with the medium described by the given function.

On the side faces of the cylinder we assume that there is no heat exchange.

Mathematical model: in a rectangular cylinder

$$D^{\infty} = \{(x, y, z) \,:\, 0 < x < l_x, 0 < y < l_y, -\infty < z < +\infty\} \tag{1}$$

we consider a cylindrical domain

$$D(F, +\infty) = \{(x, y, z) \,:\, 0 < x < l_x, 0 < y < l_y, F(x, y) < z < +\infty\}$$

bounded by a surface

$$S = \{(x, y, z) \,:\, 0 < x < l_x, 0 < y < l_y, z = F(x, y) < H\},\ F \in C^2. \tag{2}$$

In the domain $D(F, +\infty)$ we consider the following mixed boundary value problem for the Poisson equation

$$\begin{aligned}
&\Delta u(M) = \rho(M),\ M \in D(F, +\infty)\\
&\frac{\partial u}{\partial n}\Big|_S = h(U_0 - u)\Big|_S = g,\\
&\frac{\partial u}{\partial n}\Big|_{x=0,\,l_x} = 0,\quad \frac{\partial u}{\partial n}\Big|_{y=0,\,l_y} = 0,\\
&u \text{ bounded when } z \to +\infty.
\end{aligned} \tag{3}$$

Problem (3) corresponds to a stable temperature distribution created by heat sources with the distribution density function ρ.

On the surface S a third boundary condition is set and corresponds to a convective heat exchange with a medium of temperature U_0 with a constant coeffcient h. In this case we will consider the temperature of the medium as $U_0 = 0$. On the side faces $D(F, +\infty)$ there is no heat transfer — boundary conditions of the second kind take place.

We assume that the functions ρ, g are such that the solution of problem (3) exists in $C^2\,(D(F, \infty)) \cap C^1\overline{(D(F, \infty))}$. In particular, solving problem (3) allows us to find $u|_S$.

In addition, we assume that the support density ρ is in the $z > H$ domain.

Statement of the inverse problem:

Let ρ not be known.

But $u|_S = f$ is the initial thermogram.

We consider the surface S to be arbitrary

$$S = \{(x, y, z) : 0 < x < l_x, 0 < y < l_y, z = F(x, y) < H\},\ F \in C^2.$$

We need to find a continuous function ρ.

To solve the inverse problem, we apply the approach [2] used in geophysics problems.

The source of information about the density ρ will be the function $u|_{z=H}$ on the plane $z = H$, which is closer to the support of the density ρ than the surface S.

Since the support of the function ρ is outside the domain

$$D(F, H) = \{(x, y, z) : 0 < x < l_x, 0 < y < l_y, F(x, y) < z < H\}$$

then the solution of problem (3) satisfies the Laplace equation in this domain.

Assume that the functions f and g are taken from the set of solutions to the forward problem (3), so the solution to the inverse problem exists in $C^2\,(D(F, H)) \cap C^1\overline{(D(F, H))}$.

Then we obtain the continuation problem

$$\begin{aligned}
&\Delta u(M) = 0,\ M \in D(F, H),\\
&u|_S = f,\quad \frac{\partial u}{\partial n}\Big|_S = -hu\Big|_S = g,\\
&\frac{\partial u}{\partial n}\Big|_{x=0,\,l_x} = 0,\quad \frac{\partial u}{\partial n}\Big|_{y=0,\,l_y} = 0.
\end{aligned} \tag{4}$$

from the boundary S with homogeneous boundary conditions of the second kind on the side faces of D.

Note that in problem (4) the Cauchy conditions on the surface S of the form (2) are given, i.e., the boundary values f of the desired function u and the values of its normal derivative are given, so problem (4) has a single solution.

The boundary $z = H$ of the domain $D(F, H)$ is free, and, as a Cauchy problem for the Laplace equation, is unstable with respect to errors in the data, i.e., ill-posed.

In the inverse problem, the function f corresponds to the original thermogram obtained with a thermal imager.

The function $u|_{z=H}$ will be considered as a corrected thermogram, i.e., a source of more accurate information about the density ρ.

2. Exact solution of the inverse problem

Based on the approach of [8], an explicit representation of the exact solution of problem (4) is constructed similarly to [9].

Let us consider the source function $\varphi(M, P)$ of the Neumann problem for the Laplace equation in an infinite cylinder D^∞ of the form (1), i.e. the solution of the problem

$$\begin{aligned}
&\Delta w(P) = -\delta_{MP}, \ P \in D^\infty, \\
&\frac{\partial w}{\partial n}\Big|_{x=0,\, l_x} = 0, \quad \frac{\partial w}{\partial n}\Big|_{y=0,\, l_y} = 0, \\
&\frac{\partial w}{\partial z} \to \frac{1}{l_x l_y} \text{ at } z \to +\infty, \quad \frac{\partial w}{\partial z} \to 0 \text{ at } z \to -\infty.
\end{aligned} \tag{5}$$

for which the necessary solvability condition is fulfilled

$$\int\limits_S \frac{\partial w}{\partial n} dS - \int\limits_V \Delta w \, dV = 0.$$

The source function $\varphi(M, P)$ of problem (5) can be represented as

$$\varphi(M, P) = \frac{1}{4\pi r_{MP}} + W(M, P)$$

where r_{MP} is the distance between points M and P, $W(M, P)$ is a harmonic function on P. The source function can also be obtained [9] as a Fourier series under the condition $z_M < \min\limits_{(x,y)} F(x, y) < z_P$

$$\begin{aligned}
\tilde{\varphi}(M, P) = \frac{1}{2 l_x l_y} C + \frac{2}{l_x l_y} \sum_{n,m=0, n^2+m^2\neq 0}^{\infty} \varepsilon_n \varepsilon_m \frac{e^{-k_{nm}|z_M - z_P|}}{k_{nm}} \times \\
\times \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} \cos\frac{\pi n x_P}{l_x} \cos\frac{\pi m y_P}{l_y}
\end{aligned} \tag{6}$$

where

$$k_{nm} = \pi \sqrt{\frac{n^2}{l_x^2} + \frac{m^2}{l_y^2}}, \qquad \varepsilon_n = \begin{cases} 1 & \text{n} \neq 0, \\ 0,5 & \text{n} = 0. \end{cases} \tag{7}$$

Taking into account homogeneous boundary conditions for $\tilde{\varphi}$ and u on the side faces of the cylindrical domain $D(F, H)$, we obtain

$$\begin{aligned}
u(M) = \int\limits_S \left[g(P)\tilde{\varphi}(M, P) - f(P) \frac{\partial \tilde{\varphi}}{\partial n_P}(M, P) \right] d\sigma_P + \\
+ \int\limits_{\Pi(H)} \left[\frac{\partial u}{\partial n}(P) \tilde{\varphi}(M, P) - u(P) \frac{\partial \tilde{\varphi}}{\partial n_P}(M, P) \right] d\sigma_P,
\end{aligned}$$

where

$$\Pi(H) = \{(x, y, z) : 0 < x < l_x, 0 < y < l_y, z = H\} \tag{8}$$

By introducing the notations

$$\Phi(M) = -\int\limits_S \left[g(P)\tilde{\varphi}(M,P) - f(P)\frac{\partial\tilde{\varphi}}{\partial n_P}(M,P)\right] d\sigma_P,\ M \in D(-\infty, H), \tag{9}$$

$$v(M) = \int\limits_{\Pi(H)} \left[\frac{\partial u}{\partial n}(P)\tilde{\varphi}(M,P) - u(P)\frac{\partial\tilde{\varphi}}{\partial n_P}(M,P)\right] d\sigma_P,\ M \in D(-\infty, H), \tag{10}$$

we obtain the solution of problem (4) in the form

$$u(M) = v(M) - \Phi(M), \quad M \in D(F, H) \tag{11}$$

where $\Phi(M)$ is computed from known functions f and g and can be considered as a known function.

The function $v(M)$ can be viewed as a solution to the problem

$$\begin{aligned}
&\Delta v(M) = 0,\ M \in D(-\infty, H),\\
&v|_{z=H} = v_H, \quad \frac{\partial v}{\partial n}\Big|_{x=0,l_x} = 0, \quad \frac{\partial v}{\partial n}\Big|_{y=0,\,l_y} = 0,\\
&v \text{ is bounded at } z \to -\infty,
\end{aligned}$$

which can be obtained by the Fourier method, and the function v can be expressed through v_H

$$v(M) = \sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty} (\tilde{v}_H)_{nm} e^{k_{nm}(z_M - H)} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y}, \tag{12}$$

$$(\tilde{v}_H)_{nm} = \frac{4\varepsilon_n\varepsilon_m}{l_x l_y} \int\limits_0^{l_x}\int\limits_0^{l_y} v_H(x,y) \cos\frac{\pi n x}{l_x} \cos\frac{\pi m y}{l_y} dxdy. \tag{13}$$

The function v_H satisfies the Fredholm integral equation of the first kind, taking into account homogeneous boundary conditions for $\tilde{\varphi}$ and u, and notations (9) and (10), we obtain

$$v(M) = \Phi(M), \quad M \in D(-\infty, F) \tag{14}$$

Let $a < \min\limits_{(x,y)} F(x,y)$ and $M \in \Pi(a)$, where $\Pi(a)$ is a domain of the form (8) at $z = a$, then from (14) and (12) we obtain a system of equations with respect to the Fourier coefficients of the function v_H

$$\sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty} (\tilde{v}_H)_{nm} e^{k_{nm}(a-H)} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} = \Phi(M). \tag{15}$$

Using (13), equation (15) can also be written as an integral equation of the first kind

$$\int\limits_{\Pi(H)} G(M,P) v_H(P) dx_P dy_P = \Phi(M),\ M \in \Pi(a), \tag{16}$$

where the kernel of the integral operator has the form

$$\begin{aligned}
G(M,P) = \frac{4}{l_x l_y} &\sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty} \varepsilon_n\varepsilon_m e^{-k_{nm}(H-a)} \times\\
&\times \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} \cos\frac{\pi n x_P}{l_x} \cos\frac{\pi m y_P}{l_y}.
\end{aligned} \tag{17}$$

Equation (16) will also be written in the form

$$Gv_H = \Phi(a).$$

From equation (16), taking into account the expansion (17) at $z_M = a$, we obtain the relation between the Fourier coefficients of the single solution v_H and the Fourier coefficients of the right-hand side

$$(\tilde{v}_H)_{nm} e^{-k_{nm}(H-a)} = \tilde{\Phi}_{nm}(a), \tag{18}$$

where $\tilde{\Phi}_{nm}(a)$ are the Fourier coefficients of the function $\Phi(M)|_{M\in\Pi(a)}$:

$$\tilde{\Phi}_{nm}(a) = \frac{4\varepsilon_n\varepsilon_m}{l_x l_y} \int\limits_{\Pi(a)} \Phi(x,y,a) \cos\frac{\pi n x}{l_x} \cos\frac{\pi m y}{l_y} dxdy.$$

Substituting the Fourier coefficients $(\tilde{v}_H)_{nm}$ from (18) into the series (12), we obtain the function v in the domain $D(-\infty, H)$

$$v(M) = \sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty} \tilde{\Phi}_{nm}(a) e^{k_{nm}(z_M-a)} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y}. \tag{19}$$

The series (19), like series (12), converges uniformly in $D(-\infty, H-\varepsilon)$ for any $\varepsilon > 0$, if the solution of problem (4) exists given f and g.

Formula (11), where the functions v and Φ are of the form (19) and (9), respectively, gives an explicit expression for the solution of problem (4).

$$\begin{gathered} u(M) = \sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty} \tilde{\Phi}_{nm}(a) e^{k_{nm}(z_M-a)} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} + \\ + \int\limits_S \left[g(P)\tilde{\varphi}(M,P) - f(P)\frac{\partial\tilde{\varphi}}{\partial n_P}(M,P) \right] d\sigma_P. \end{gathered} \tag{20}$$

3. Approximately given surface S. Calculation of the normal to the surface

Formula (20) gives an explicit expression for the solution of problem (4).

Since the surface S of the form (2)

$$S = \{(x,y,z) : 0 < x < l_x, 0 < y < l_y, z = F(x,y) < H\}$$

is given by the equation $z = F(x,y)$, the function f, defined on S, can be viewed as a function of the variables x and y on the rectangle Π:

$$\Pi = \{(x,y) : 0 < x < l_x, 0 < y < l_y\}. \tag{21}$$

In applied problems, the surface S, which is a part of the boundary of the domain, can be defined on the basis of measurements, i.e. approximated. If the surface S is defined with an error, the calculation of the integral (9) is complicated by the necessity to calculate the normal to such a surface. The problem of calculating the normal to the surface, in other words, the gradient of a function given approximatively, is ill-posed as a problem of numerical differentiation.

To compute the function Φ (9), it is necessary to compute [10] the vector function $\mathbf{n}_1$ of the normal to the surface S, which is the gradient of the function $F(x;y) - z$ of the following form

$$\mathbf{n}_1 = (F'_x, F'_y, -1) = iF'_x + jF'_y - k = grad(F(x,y) - z) = \nabla_{xy}F - k$$

$$\frac{\partial\tilde{\varphi}}{\partial n} = (\mathbf{n}, \nabla\tilde{\varphi}), \quad \mathbf{n} = \frac{\mathbf{n}_1}{n_1}$$

$$d\sigma_P = n_1(x_P, y_P)dx_Pdy_P$$

The integral (9) can be represented as

$$\Phi(M) = -\int\limits_{\Pi} \left[g(P)\tilde{\varphi}(M,P) - f(P)\left(\nabla_P\tilde{\varphi}(M,P), \mathbf{n}(P)\right)\right] n_1(P)dx_Pdy_P,$$

$$\Phi(M) = -\int\limits_{\Pi} \left[g(P)\tilde{\varphi}(M,P)n_1(P) - f(P)\left(\nabla_P\tilde{\varphi}(M,P), n_1(P)\right)\right] dx_Pdy_P. \tag{22}$$

Let us assume that the surface S is defined with some error, namely: instead of the exact function F, there is a function F^μ such that

$$\|F^\mu - F\|_{L_2(\Pi)} \leq \mu. \tag{23}$$

As an approximation to the function $\nabla_{xy}F$, calculated from the known function F^μ, related to F by the condition (23), consider the gradient from the extremum of the functional

$$N^\beta[W] = \|W - F^\mu\|^2_{L_2(\Pi)} + \beta\|\nabla W\|^2_{L_2(\Pi)} \tag{24}$$

where Π is a plane of the form (21).

We will consider such surfaces S for which

$$F|_{x=0,\,l_x} = F|_{y=0,\,l_y} = 0.$$

This condition, in particular, takes place in the case when S can be considered as a perturbation of the main plane $z = 0$. Then the extremal of the functional (24) satisfies the Euler equation

$$\begin{gathered} -\beta\Delta W + W = F^\mu, \\ W|_{x=0,\,l_x} = W|_{y=0,\,l_y} = 0. \end{gathered} \tag{25}$$

Solving the problem (25) by Fourier method, we obtain:

$$W^\mu_\beta(x,y) = \sum_{n,m=1}^{\infty} \frac{\tilde{F}^\mu_{nm}}{1 + \beta k^2_{nm}} \sin\frac{\pi nx}{l_x} \sin\frac{\pi my}{l_y}. \tag{26}$$

It is easy to see that the series (26) converges uniformly to Π (21).

As an approximation of the gradient of the function F^μ we will consider the vector function

$$\begin{gathered} \nabla_{xy}W^\mu_\beta(x,y) = \sum_{n,m=1}^{\infty} \frac{\tilde{F}^\mu_{nm}}{1 + \beta k^2_{nm}} \times \\ \times\left(\mathbf{i}\frac{\pi n}{l_x}\cos\frac{\pi nx}{l_x}\sin\frac{\pi my}{l_y} + \mathbf{j}\frac{\pi m}{l_y}\cos\frac{\pi my}{l_y}\sin\frac{\pi nx}{l_x}\right). \end{gathered} \tag{27}$$

The series (27) is also uniformly convergent on Π (21).

Let F^- be an odd-periodic continuation of a function F with period $2l_x$, on the variable x and with period $2l_y$ on the variable y, i.e.

$$\begin{aligned}&F^-(-x,y)=-F^-(x,y), \quad (x,y)\in\Pi,\\ &F^-(x,-y)=-F^-(x,y), \quad F^-(-x,y)=F^-(x,-y),\\ &F^-(x+2l_xn,y+2l_ym)=F^-(x,y)=F(x,y).\end{aligned}$$

Similarly to [11–13] it can be proved

Theorem 1. *Let $F^-\in C^2(R^2)$, $\beta=\beta(\mu)>0$, $\beta(\mu)\to 0$ and $\mu/\sqrt{\beta(\mu)}\to 0$ if $\mu\to 0$. Then*

$$\|\nabla_{xy}W^{\mu}_{\beta(\mu)}-\nabla_{xy}F\|_{L_2(\Pi)}\le \frac{\mu}{2\sqrt{\beta}}+\frac{\sqrt{\beta}}{2}\|\varDelta F\|_{L_2(\Pi)}\to 0 \; at\; \mu\to 0.$$

Based on the theorem, we can use formula (27) for approximate calculation of the normal to the surface:

$$\mathbf{n}^{\mu}_{1,\beta}=\nabla_{xy}W^{\mu}_{\beta}-k. \tag{28}$$

It follows from the proof of the theorem that under the conditions formulated in the theorem

$$\|\mathbf{n}^{\mu}_{1,\beta}-\mathbf{n}_1\|_{L_2(\Pi)}=\|\nabla_{xy}W^{\mu}_{\beta}-\nabla_{xy}F\|_{L_2(\Pi)}\le \frac{\mu}{2\sqrt{\beta}}+\frac{\sqrt{\beta}}{2}\|\varDelta F\|_{L_2(\Pi)}$$

the maximum in β of the expression on the right-hand side is reached when $\beta(\mu)=\frac{\mu}{\|\varDelta F\|}$, and thus, denoting by (28)

$$\mathbf{n}^{\mu}_{1}=\mathbf{n}^{\mu}_{1,\beta}=\nabla_{xy}W^{\mu}_{\beta(\mu)}-k, \tag{29}$$

we obtain:

$$\|\mathbf{n}^{\mu}_{1}-\mathbf{n}_1\|_{L_2(\Pi)}\le\sqrt{\|\varDelta F\|\mu}\xrightarrow[\mu\to 0]{}0. \tag{30}$$

It is also not difficult to obtain an approximate estimate

$$\|W_{\beta(\mu)}-F\|_{L_2(\Pi)}\le 2\mu.$$

The surface defined by the equation $z=W^{\mu}_{\beta(\mu)}(x,y)$ we denote as

$$S^{\mu}=\{(x,y,z)\,:\,0<x<l_x, 0<y<l_y, z=W^{\mu}_{\beta(\mu)}(x,y)\}.$$

4. Stable approximate solution in case of inaccurate data on the approximated boundary

Now let the functions f and g in problem (4) be approximated, namely: let f^{δ} and g^{δ} be given such that

$$\|f^{\delta}-f\|_{L_2(\Pi)}\le\delta, \quad \|g^{\delta}-g\|_{L_2(\Pi)}\le\delta. \tag{31}$$

Then the function $\varPhi$ of the form (22) at exactly given surface S can be calculated with some error:

$$\begin{aligned}\varPhi^{\delta}(M)=-\int\limits_{\Pi}&\left[g^{\delta}(P)\tilde{\varphi}(M,P)n_1(P)-\right.\\ -f^{\delta}(P)&\left.\left(\nabla_P\tilde{\varphi}(M,P),\mathbf{n}^{\mu}_1(P)\right)\right]_{P\in S}dx_Pdy_P.\end{aligned} \tag{32}$$

Obtaining an approximate solution of problem (4) under conditions (31) and its convergence to the exact solution (20) for a precisely defined surface S is considered in [9].

When approximating the surface S (23)

$$\|F^{\mu}-F\|_{L_2(\Pi)}\leq\mu$$

the right-hand side of (16) can be calculated approximating by formula (32) taking into account (29):

$$\begin{gathered}\Phi^{\delta,\mu}(M)=-\int\limits_{\Pi}\left[g^{\delta}(P)\tilde{\varphi}(M,P)n_1^{\mu}(P)-\right.\\ \left.-f^{\delta}(P)\left(\nabla_P\tilde{\varphi}(M,P),\mathbf{n}_1^{\mu}(P)\right)\right]_{P=P(x,y,W_{\beta}^{\mu})\in S^{\mu}}dx_Pdy_P.\end{gathered}\tag{33}$$

Let us evaluate the difference

$$\begin{gathered}|\Phi^{\delta,\mu}(M)-\Phi(M)|\leq|\Phi^{\delta,\mu}(M)-\Phi^{\delta}(M)|+\\ +|\Phi^{\delta}(M)-\Phi(M)|,\quad M\in\Pi(a),\end{gathered}\tag{34}$$

where $\Phi^{\delta,\mu}$, Φ^{δ}, Φ are functions of the form (33), (32), (22), respectively. We evaluate the first difference in (34) by subtracting and adding the function:

$$\begin{gathered}\Phi_1^{\delta,\mu}(M)=\int\limits_0^{l_x}dx\int\limits_0^{l_y}dy\left[g^{\delta}(x,y)\tilde{\varphi}(M,P^{\mu})n_1(x,y)-\right.\\ \left.-f^{\delta}(x,y)\left(\nabla_P\tilde{\varphi}(M,P^{\mu}),\mathbf{n}_1\right)\right],\ P^{\mu}=(x,y,W_{\beta}^{\mu}(x,y)),\ M=(x_M,y_M,a)\end{gathered}$$

differing from the function $\Phi^{\delta,\mu}(M)$ by the exact normal:

$$\begin{gathered}|\Phi^{\delta,\mu}(M)-\Phi^{\delta}(M)|=|\Phi^{\delta,\mu}(M)-\Phi_1^{\delta,\mu}(M)+\Phi_1^{\delta,\mu}(M)-\Phi^{\delta}(M)|=\\ =\left|\int\limits_0^{l_x}dx\int\limits_0^{l_y}dy\left[f^{\delta}(x,y)\left(\mathbf{n}_1^{\mu}(x,y)-\mathbf{n}_1(x,y),\nabla_P\tilde{\varphi}(M,P^{\mu})\right)-\right.\right.\\ -g^{\delta}(x,y)\tilde{\varphi}(M,P^{\mu})\left(n_1^{\mu}(x,y)-n_1(x,y)\right)+\\ +f^{\delta}(x,y)\left(\mathbf{n}_1(x,y),\nabla_P\left(\tilde{\varphi}(M,P^{\mu})-\tilde{\varphi}(M,P)\right)\right)-\\ \left.\left.-\ g^{\delta}(x,y)\left(\tilde{\varphi}(M,P^{\mu})-\tilde{\varphi}(M,P)\right)n_1(x,y)\right]\right|.\end{gathered}$$

Replacing the modulus by the sum of moduli and evaluating the difference of functions using the Lagrange formula, we obtain:

$$\begin{gathered}|\Phi^{\delta,\mu}(M)-\Phi^{\delta}(M)|\leq\int\limits_0^{l_x}dx\int\limits_0^{l_y}dy\left[|f^{\delta}(x,y)|\cdot|\mathbf{n}_1^{\mu}(x,y)-\mathbf{n}_1(x,y)|\cdot|\nabla_P\tilde{\varphi}(M,P^{\mu})|\ +\right.\\ +|g^{\delta}(x,y)|\cdot|\tilde{\varphi}(M,P^{\mu})|\cdot|n_1^{\mu}(x,y)-n_1(x,y)|+\\ +|f^{\delta}(x,y)|\cdot|\mathbf{n}_1(x,y)|\cdot\left|\frac{\partial}{\partial z_P}\nabla_P\tilde{\varphi}(M,P^*)\right|\cdot|W_{\beta}^{\mu}-F|+\\ \left.+\ |g^{\delta}(x,y)|\cdot\left|\frac{\partial\tilde{\varphi}(M,P^*)}{\partial z_P}\right||W_{\beta}^{\mu}-F|\cdot|n_1(x,y)|\right]\leq\end{gathered}$$

taking out the maxima, we use the Cauchy–Bunyakovsky inequality:

$$\leq \left(\max_{M\in\Pi(a)} |\nabla_P \tilde{\varphi}(M,P^\mu)| \cdot \|f^\delta\| + \max_{M\in\Pi(a)} |\tilde{\varphi}(M,P^\mu)| \cdot \|g^\delta\| \right) \|\mathbf{n}_1^\mu - \mathbf{n}_1\| +$$
$$+ \left(\max_{M\in\Pi(a)} \left| \frac{\partial}{\partial z_P} \nabla_P \tilde{\varphi}(M,P^*) \right| \cdot |n_1| \|f^\delta\| + \max_{M\in\Pi(a)} \left| \frac{\partial}{\partial z_P} \tilde{\varphi}(M,P^*) \right| \cdot |n_1| \|g^\delta\| \right) \|W_\beta^\mu - F\| \leq$$

By virtue of the inequalities $\|f^\delta\| - \|f\| \leq |\|f^\delta\| - \|f\|| \leq \|f^\delta - f\| \leq \delta$ we obtain $\|f^\delta\| \leq \|f\| + \delta$ and thus,

$$\leq \left(\max_{M\in\Pi(a)} |\nabla_P \tilde{\varphi}(M,P^*)| \cdot (\|f\| + \delta) + \max_{M\in\Pi(a)} |\tilde{\varphi}(M,P^\mu)| \cdot (\|g\| + \delta) \right) \|\mathbf{n}_1^\mu - \mathbf{n}_1\| +$$
$$+ \left(\max_{M\in\Pi(a)} \left| \frac{\partial}{\partial z_P} \nabla_P \tilde{\varphi}(M,P^*) \right| \cdot |n_1| (\|f\| + \delta) + \max_{M\in\Pi(a)} \left| \frac{\partial}{\partial z_P} \tilde{\varphi}(M,P^*) \right| \cdot |n_1| (\|g\| + \delta) \right) \|W_\beta^\mu - F\|.$$

The maximums are evaluated by constants. Since we are interested in the behaviour of the regularized solution of problem (4) when $\delta \to 0$, we can assume that $\delta \leq \delta_0$, and thus, taking into account (30)

$$|\Phi^{\delta,\mu}(M) - \Phi^\delta(M)|_{M\in\Pi(a)} \leq C_1 \|\mathbf{n}_1^\mu - \mathbf{n}_1\| + C_2 \|W_\beta^\mu - F\|.$$

Consider the difference

$$\|W_\beta^\mu - F\| \leq \|W_\beta^\mu - W_\beta\| + \|W_\beta - F\|, \tag{35}$$

where W_β is calculated by formula (26) at $\mu = 0$

$$W_\beta = \sum_{n,m=1}^{\infty} \frac{\tilde{F}_{nm}}{1 + \beta k_{nm}^2} \sin \frac{\pi n x}{l_x} \sin \frac{\pi m y}{l_y}.$$

The evaluation of the first difference in (35) gives:

$$\|W_\beta^\mu - W_\beta\|^2 = \int_0^{l_x} dx \int_0^{l_y} dy \left| \sum_{n,m=1}^{\infty} \frac{\tilde{F}_{nm}^\mu - \tilde{F}_{nm}}{1 + \beta k_{nm}^2} \sin \frac{\pi n x}{l_x} \sin \frac{\pi m y}{l_y} \right|^2.$$

Using the orthogonality of the trigonometric system, we obtain:

$$\|W_\beta^\mu - W_\beta\|^2 = \frac{l_x l_y}{4} \sum_{n,m=1}^{\infty} \frac{(\tilde{F}_{nm}^\mu - \tilde{F}_{nm})^2}{(1 + \beta k_{nm}^2)^2} \leq$$
$$\leq \frac{l_x l_y}{4} \sum_{n,m=1}^{\infty} (\tilde{F}_{nm}^\mu - \tilde{F}_{nm})^2 = \|\tilde{F}_{nm}^\mu - \tilde{F}_{nm}\|^2 \leq \mu^2.$$

Similarly, to evaluate the second difference in (35) when $\beta = \frac{\mu}{\|\Delta F\|}$ we obtain:

$$\|W_\beta^\mu - F\|^2 \leq \mu^2.$$

Combining the estimates, we obtain

$$\|W_\beta - F\| \leq 2\mu. \tag{36}$$

From (30), (35) and (36) we obtain

$$\begin{aligned} |\Phi^{\delta,\mu}(M) - \Phi^\delta(M)| &\leq C\|\mathbf{n}_1^\mu - \mathbf{n}_1\| + C\|W_\beta^\mu - F\| \leq \\ &\leq C\mu + C\sqrt{\mu} \leq C\sqrt{\mu}(1 + C\mu) = C_1\sqrt{\mu}, \quad M \in \Pi(a). \end{aligned} \tag{37}$$

To evaluate the second difference, we obtain the same way as in [14]:

$$\begin{gathered}|\Phi^{\delta}(M)-\Phi(M)| = \left|\int\limits_{\Pi}\left[(f^{\delta}(P)-f(P))(\mathbf{n}_1(P),\nabla_P\tilde{\varphi}(M,P))-\right.\right.\\ \left.\left.-(g^{\delta}(P)-g(P))\tilde{\varphi}(M,P)n_1(P)\right]_{P\in S}dx_Pdy_P\right| \leq\\ \leq Const_{n_1}\max_{M\in\Pi(a)}\|\nabla\tilde{\varphi}(M)\|_{L_2(S)}\cdot\|f^{\delta}-f\|_{L_2(\Pi)}+\\ +Const^1_{n_1}\max_{M\in\Pi(a)}\|\tilde{\varphi}(M)\|_{L_2(S)}\cdot\|g^{\delta}-g\|_{L_2(\Pi)}\leq C_2\delta,\quad M\in\Pi(a).\end{gathered}\tag{38}$$

From (37) and (38), we obtain for the estimate (34):

$$\max_{M\in\Pi(a)}|\Phi^{\delta,\mu}(M)-\Phi(M)|\leq C_1\sqrt{\mu}+C_2\delta=\Delta(\mu,\delta)\xrightarrow[\substack{\mu\to 0\\ \delta\to 0}]{}0\tag{39}$$

Thus, the right part of the integral equation (16) is known with some error Δ, having the structure (39). The stable approximate solution of the problem (4) is constructed on the basis of the search for extrema of the Tikhonov functional [5], and can be obtained in the form of

$$u^{\delta,\mu}_{\alpha}(M)=v^{\delta,\mu}_{\alpha}(M)-\Phi^{\delta,\mu}(M),\ M\in D(H,F)\tag{40}$$

where $\Phi^{\delta,\mu}$ is a function of the form (33), and $v^{\delta,\mu}_{\alpha}$ (19) has the form:

$$v^{\delta,\mu}_{\alpha}(M)=\sum_{n,m=0,\ n^2+m^2\neq 0}^{\infty}\frac{\tilde{\Phi}^{\mu,\delta}_{nm}(a)e^{k_{nm}(z_M-a)}}{1+\alpha e^{2k_{nm}(H-a)}}\cos\frac{\pi n x_M}{l_x}\cos\frac{\pi m y_M}{l_y}.\tag{41}$$

Here $\tilde{\Phi}^{\mu,\delta}_{nm}(a)$ is the Fourier coefficients of the function $\Phi^{\mu,\delta}_{nm}(a)|_{M\in\Pi(a)}$:

$$\tilde{\Phi}^{\mu,\delta}_{nm}(a)=\frac{4\varepsilon_n\varepsilon_m}{l_xl_y}\int\limits_0^{l_x}dx\int\limits_0^{l_y}dy\,\Phi^{\delta,\mu}(x,y,a)\cos\frac{\pi nx}{l_x}\cos\frac{\pi ny}{l_y}$$

and α is the regularisation parameter. According to the notations introduced above, the value of a is chosen such that

$$a<\min_{(x,y)\in\Pi}F(x,y).$$

For the considered boundary conditions of the second kind on the side faces of the cylinder, taking into account [8], there is a theorem of convergence of the approximate solution to the exact one when the regularisation parameter, consistent with the accuracy of the initial data, tends to zero.

Theorem 2. *Let the solution of problem* (4) *exist in the domain* $D(H,F)$, $\alpha=\alpha(\Delta)$, $\alpha(\Delta)\to 0$, $\Delta/\sqrt{\alpha(\Delta)}\to 0$ *at* $\Delta\to 0$. *Then the function* $u_{\alpha(\Delta)}$ *of the form* (40), *where according to* (39) $\Delta=\Delta(\mu,\delta)=C_1\sqrt{\mu}+C_2\delta$, *converges uniformly to the exact solution of problem* (4) *at* $\delta\to 0$, $\mu\to 0$ *in the domain* $D(F+\varepsilon,H-\varepsilon)$, *where* $\varepsilon>0$ *is some fixed arbitrarily small number.*

Proof. Let's evaluate the difference

$$|u^{\delta,\mu}_{\alpha(\delta)}-u|\leq|v^{\delta,\mu}_{\alpha}-v|+|\Phi^{\delta,\mu}-\Phi|$$

in the domain $G(H+\varepsilon, F-\varepsilon)$.

The second difference is evaluated similarly to (38) when replacing $\Pi(a)$ by $D(H+\varepsilon, F-\varepsilon)$.

For the difference $v^{\delta}_{\alpha} - v$ we obtain

$$|v^{\delta}_{\alpha} - v| \le |v^{\delta}_{\alpha} - v_{\alpha}| + |v_{\alpha} - v|,$$

where v_{α} is a function of the form (41) at $\delta = 0$.

We estimate $v^{\delta}_{\alpha} - v_{\alpha}$ in $D(H+\varepsilon, F-\varepsilon)$.

$$|v^{\delta}_{\alpha(M)} - v_{\alpha(M)}| \le \left| \sum_{n,m=0}^{\infty} \frac{e^{k_{nm}(z_M - a)}}{1 + \alpha e^{2k_{nm}(H-a)}} \right| \cdot 4 \max_{P \in \Pi(a)} |\Phi^{\delta}(P) - \Phi(P)| \le$$

$$\le C_1 \cdot \delta \sum_{n,m=0}^{\infty} \frac{e^{k_{nm}(H-\varepsilon-a)}}{1 + \alpha e^{2k_{nm}(H-a)}} \le C_1 \cdot \delta \cdot \max_{x'} \left[\frac{e^{x'}}{1 + \alpha e^{2x'}} \right] \sum_{n,m=0}^{\infty} e^{-k_{nm}\varepsilon} \le C_2 \cdot \frac{\Delta(\mu, \delta)}{\sqrt{\alpha}}.$$

For the difference $v_{\alpha} - v$ we obtain

$$|v_{\alpha} - v| = \left| \sum_{n,m=0}^{\infty} \frac{\alpha e^{2k_{nm}(H-a)} e^{\pi k_{nm}(z_M - H)}}{1 + \alpha e^{2k_{nm}(H-a)}} (v_H)_{nm} \cos \frac{\pi n x_M}{l_x} \frac{\pi m y_M}{l_y} \right| \le$$

$$\le \left[\sum_{n,m=0}^{\infty} \frac{\alpha e^{2k_{nm}(H-a)}}{1 + \alpha e^{2k_{nm}(H-a)}} e^{-\varepsilon k_{nm}} \right]^{\frac{1}{2}} \cdot \|v_H\|_{L_2}.$$

Since the parameter-dependent series is majorised by a convergent numerical series

$$\sum_{n,m=0}^{\infty} e^{-\varepsilon k_{nm}}$$

then a limit transition on α is possible, and thus,

$$|v^{\delta,\mu}_{\alpha} - v| \to 0 \text{ при } \alpha(\delta) \to 0.$$

Using 2, the convergence of the approximate solution (40) to the exact solution (20) of the problem (4) of continuation from the boundary S (2) is proved.

5. Numerical solution of the inverse problem for the case of flat boundary

Let us demonstrate the effectiveness of the proposed approach of solving the problem (4) of continuation from the boundary S, on which the third boundary condition corresponding to convective heat exchange with the medium of temperature U_0 with a constant coefficient h is defined

$$\frac{\partial u}{\partial n}\Big|_S = g = h(U_0 - f)\big|_S$$

and the surface S itself is the plane $\Pi(0)$ $z = 0$ for the following conditions: $U_0 = 0$, $h = 0.4$, $l_x = 60$, $l_y = 60$, $H = 1.5$

Let the function $\rho(M)$ in the direct problem (3) correspond to three point sources in the plane $\Pi(H)$: $(x_1, y_1) = (30, 32)$, $(x_2, y_2) = (30, 30)$, $(x_3, y_3) = (32, 30)$.

For this case, using the results of [15], the function specifying the boundary condition for problem (3) with accuracy up to a constant can be obtained in the form of

$$f(x,y) = \sum_{n,m=0}^{\infty} \sum_{i=0}^{3} q_i \varepsilon_n \varepsilon_m \frac{e^{-k_{nm}H}}{k_{nm}+h} \cos\frac{\pi m x_i}{l_x} \cos\frac{\pi m y_i}{l_y} \cos\frac{\pi m x}{l_x} \cos\frac{\pi m y}{l_y}, \tag{42}$$

where k_{nm}, ε_n and ε_m are calculated by formula (7) and $q_i = 100$, $i = 1,2,3$.

We will solve the inverse continuation problem (4), assuming that the value of the function on the boundary f^δ is given approximated, its values will be determined on the basis of the function $f(x,y)$ (42) and a randomly given relative error within 3%.

In the applied approach, the solution can be obtained by applying (40) and (41).

According to (32)

$$\Phi^\delta(M) = -\int\limits_{\Pi(0)} \left[g^\delta(P)\tilde{\varphi}(M,P)n_1(P) - \right.$$
$$\left. -f^\delta(P)\left(\nabla_P\tilde{\varphi}(M,P), \mathbf{n}_1^\mu(P)\right)\right]_{P\in S} dx_P dy_P$$

where $\mathbf{n}_1 = (F'_x, F'_y, -1)$ and $n_1 = |\mathbf{n}_1| = \sqrt{1+(F'_x)^2+(F'_y)^2}$.

For the assumption that S is a plane $z = F(x,y) = 0$, $\mathbf{n}_1 = (0,0,-1)$ and $n_1 = 1$.

According to (6) the source function with accuracy up to constant

$$\tilde{\varphi}(M,P) = \frac{2}{l_x l_y} \sum_{n,m=0, n^2+m^2\neq 0}^{\infty} \varepsilon_n \varepsilon_m \frac{e^{-k_{nm}|z_M - z_P|}}{k_{nm}} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} \cos\frac{\pi n x_P}{l_x} \cos\frac{\pi m y_P}{l_y}$$

and

$$grad_P\varphi(M,P)|_{P\in S} = \frac{-2}{l_x l_y} \sum_{n,m=0}^{\infty} \varepsilon_n \varepsilon_m e^{-k_{nm}|z_M - z_P|} \cos\frac{\pi n x_M}{l_x} \cos\frac{\pi m y_M}{l_y} \cos\frac{\pi n x_P}{l_x} \cos\frac{\pi m y_P}{l_y}.$$

To obtain numerical results, problems (3), (4) are discretised.

We will assume that the rectangles $\Pi(0)$, $\Pi(H)$ and $\Pi(a)$, $a = -0.6$, are covered by a uniform grid $(N_x+1)\times(N_y+1)$ of points such that

$$x_i = i\frac{l_x}{N_x},\ i = 0,\dots,N_x,$$
$$y_j = j\frac{l_y}{N_y},\ j = 0,\dots,N_y$$

We will consider $N_x = N_y = 60$.

As a result of descretisation, using the approach [16], we obtain

$$\tilde{\Phi}^\delta_{nm}(a) = \left[1 + \frac{h}{k_{nm}}\right] \frac{2}{N_x N_y} \varepsilon_n \varepsilon_m e^{k_{nm}a} \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} f^\delta(x_i, y_i) \cos\frac{\pi n i}{N_x} \cos\frac{\pi m j}{N_y} \tag{43}$$

$$v^N_\delta(x_i, y_j, H) = -\sum_{m=0}^{N_x-1} \sum_{n=0}^{N_y-1} \frac{\tilde{\Phi}^\delta_{nm}(a) e^{k_{nm}(H-a)}}{1+\alpha e^{2k_{nm}(H-a)}} \cos\frac{\pi n i}{N_x} \cos\frac{\pi m j}{N_y}, \tag{44}$$
$$i = 0,\dots,N_x, \quad j = 0,\dots,N_y.$$

And, thus, according to (40), as a result of function recovery at $z = H$ we obtain

$$u^N_\delta(H) = v^N_\delta(H) - \Phi^N(H).$$

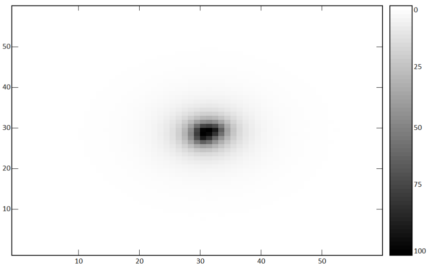


Figure 1. Initial thermogram on the surface S

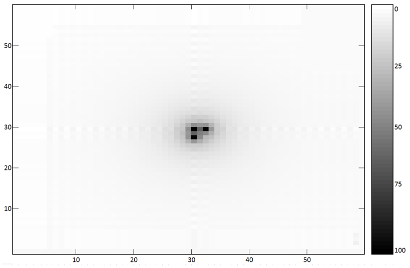


Figure 2. Adjusted thermogram obtained as an approximate solution of the inverse problem $u|_{z=H}$

The results of calculations are shown in Fig. 1 and Fig. 2.

Fig. 1 shows the initial data of the inverse problem — the function f^δ calculated by the discrete analogue of the formula (42) with the addition of a randomly specified error within 3%. The three sources are perceived as a single unit.

Fig. 2 shows the result of function recovery using (43), (44). Three sources are clearly visible.

When computing (44) discrete Fourier series, the algorithms described in [17–20] can be used.

The value of the obtained solution is calculated for the boundary conditions of the problem (4) with accuracy to a constant. Accordingly, Fig. 1 and Fig. 2 show the values normalised from 0 to 100.

Conclusion

When solving the inverse problem (4) of continuation from the boundary S, the function f can be interpreted as the original image obtained with the thermal imager or as the original thermogram.

The thermogram obtained with the help of a thermal imager reproduces with a certain degree of reliability the image of the structure of heat sources located inside the body. Then the solution of the inverse problem obtained as a result of the proposed approach can be considered as a mathematical processing of the thermogram, the obtained function $u|_{z=H}$ represents the temperature distribution on the plane located closer to the investigated heat sources than the initial surface S, we can expect a more accurate reproduction of the image of the sources on the calculated thermogram $u|_{z=H}$.

The above calculations show the effectiveness of the proposed method based on the stable solution of the inverse continuation problem (40) and (41) and its applicability for processing thermographic images, in particular, in medicine [1].

Author Contributions: Conceptualization, Evgeniy B. Laneev and Alexander V. Klimishin; methodology, Evgeniy B. Laneev and Alexander V. Klimishin; software, Evgeniy B. Laneev and Alexander V. Klimishin; validation, Evgeniy B. Laneev and Alexander V. Klimishin; formal analysis, Evgeniy B. Laneev and Alexander V. Klimishin; investigation, Evgeniy B. Laneev and Alexander V. Klimishin; resources, Evgeniy B. Laneev and Alexander V. Klimishin; data curation, Evgeniy B. Laneev and Alexander V. Klimishin; writing—original draft preparation, Evgeniy B. Laneev and Alexander V. Klimishin; writing—review and editing, Evgeniy B. Laneev and Alexander V. Klimishin; visualization, Evgeniy B. Laneev and Alexander V. Klimishin; supervision, Evgeniy B. Laneev and Alexander V. Klimishin; project administration, Evgeniy B. Laneev and Alexander V. Klimishin; funding acquisition, Evgeniy B. Laneev and Alexander V. Klimishin. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ivanitskii, G. R. Thermovision in medicine. Russian. *Vestnik RAN* **76,** 44–53 (2006).
2. Tikhonov, A. N. & Glasko, V. B. Use of the regularization method in non-linear problems. *U.S.S.R. Comput. Math. Math. Phys.* **5,** 93–107. doi:10.1016/0041-5553(65)90150-3 (1965).
3. Ivanov, V. K., Vasin, V. V. & Tanana, V. P. *The theory of linear ill-posed problems and its applications* Russian (Nauka, Moscow, 1978).
4. Vasin, V. V. The stable evaluation of a derivative in space $C(-\infty, +\infty)$. *U.S.S.R. Comput. Math. Math. Phys.* **13,** 16–24. doi:10.1016/0041-5553(73)90002-5 (1973).
5. Tihonov, A. N. & Arsenin, V. Y. *Methods for solving ill-posed problems* Russian (Nauka, Moscow, 1979).
6. Tihonov, A. N., Glasko, V. B., Litvinenko, O. K. & Melihov, V. R. On the continuation of the potential towards disturbing masses based on the regularization method. Russian. *Vestnik RAN* **1,** 30–48 (1968).
7. Laneev, E. B., Chernikova, N. Y. & Baaj, O. Application of the minimum principle of a Tikhonov smoothing functional in the problem of processing thermographic data. *Advances in Systems Science and Applications* **1,** 139–149. doi:10.25728/assa.2021.21.1.1055 (2021).
8. Baaj, O., Chernikova, N. Y. & Laneev, E. B. Correction of Thermographic Images Based on the MIinimization Method of Tikhonov Functional. *Yugoslav Journal of Operations Research* **32,** 407–424. doi:10.2298/YJOR211015026B (2022).
9. Laneev, E. B. & Klimishin, A. V. On an approximate solution to an ill-posed mixed boundary value problem for the Laplace equation in a cylindrical domain with homogeneous conditions of the second kind on the lateral surface of the cylinder. Russian. *Russian Universities Reports. Mathematics* **29,** 164–175. doi:10.20310/2686-9667-2024-29-146-164-175 (2024).
10. Laneev, E. B. & Baaj, O. On a stable calculation of the normal to a surface given approximately. *Discrete and Continuous Models and Applied Computational Science* **3,** 228–241. doi:10.22363/2658-4670-2023-31-3-228-241 (2023).
11. Laneev, E. B. & Muratov, M. N. On the stable solution of a mixed boundary value problem for the Laplace equation with an approximately given boundary. Russian. *Vestnik RUDN. Seriya Matematika* **1,** 102–111 (2002).
12. Laneev, E. B. & Muratov, M. N. Ob odnoy obratnoy zadache k kraevoy zadache dlya uravneniya Laplasa s usloviem tret'ego roda na netochno zadannoy granitse. Russian. *Vestnik RUDN. Seriya Matematika* **1,** 100–110 (2003).
13. Chernikova, N. Y., Laneev, E. B., Muratov, M. N. & Ponomarenko, E. Y. On an Inverse Problem to a Mixed Problem for the Poisson Equation. *Mathematical Analysis With Applications. CONCORD-90 2018* **318,** 141–146 (2020).
14. Laneev, E. B. & Bhuvana, V. Ob ustojchivom reshenii odnoj smeshannoj zadachi dlya uravneniya Laplasa. Russian. *Vestnik RUDN. Seriya Prikladnaya matematika i informatika* **1,** 128–133 (1999).
15. Laneev, E. B., Lesik, P. A., Klimishin, A. V., Kotyukov, A. M. & Romanov A. A. and, K. A. G. On a stable approximate solution of an ill-posed boundary value problem for the metaharmonic equation. Russian. *Russian Universities Reports. Mathematics* **25,** 156–164. doi:10.20310/2686-9667-2020-25-130-156-164 (2020).
16. Laneev, E. B., Mouratov, M. N. & Zhidkov, E. P. Discretization and its proof for numerical solution of a Cauchy problem for Laplace equation with inaccurately given Cauchy conditions on an inaccurately defined arbitrary surface. *Phys. Part. Nuclei Lett* **5,** 164–167. doi:10.1134/S1547477108030059 (2008).

17. Baaj, O. On the application of the Fourier method to solve the problem of correction of thermographic images. *Discrete and Continuous Models and Applied Computational Science* **30,** 205–216. doi:10.22363/2658-4670-2022-30-3-205-216 (2022).
18. Hamming, R. W. *Numerical methods for scientists and engineers* (McGraw-Hill Book Company, New York, 1962).
19. Laneev, E. B. & Baaj, O. On a modification of the Hamming method for summing discrete Fourier series and its application to solve the problem of correction of thermographic images. *Discrete and Continuous Models and Applied Computational Science* **30,** 342–356. doi:10.22363/2658-4670-2022-30-4-342-356 (2022).
20. Morozov, V. A. On a stable method for computing the values of unbounded operators. Russian. *Dokladi AN SSSR* **185,** 267–270 (1969).

Information about the authors

Laneev, Evgeniy B.—Doctor of Physical and Mathematical Sciences, Professor of the Mathematical Institute named after S. M. Nikolsky of RUDN University (e-mail: elaneev@yandex.ru, phone: +7 (903) 133-36-22, ORCID: 0000-0002-4255-9393, ResearcherID: G-7887-2016, Scopus Author ID: 24366681900)

Klimishin, Alexander V.—Post-Graduate student of the Mathematical Institute named after S. M. Nikolsky of RUDN University (e-mail: sa-sha-02@yandex.ru, phone: +7 (916) 788-94-39)

УДК 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-57-73 EDN: ABHFKC

Об устойчивом приближённом решении некорректно поставленной краевой задачи для уравнения Лапласа с однородными условиями второго рода на краях при неточных данных на приближённо заданной границе

Е. Б. Ланеев, А. В. Климишин

Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

Аннотация. В работе рассматривается некорректно поставленная задача продолжения гармонических функций с неточно заданной границы в цилиндрической области с однородными краевыми условиями второго рода на боковых гранях. Значение функции и её нормальной производной (условия Коши) — известны приближённо на приближённо заданной поверхности произвольного вида, ограничивающей цилиндр. В данном случае задача Коши для уравнения Лапласа обладает свойством неустойчивости по отношению к погрешности в данных Коши, т. е. является некорректно поставленной. На основе представлений о функции источника исходной задачи, точное решение представляется в виде суммы двух функций, одна из которых явно зависит от условий Коши, вторая может быть получена как решение интегрального уравнения Фредгольма первого рода в виде ряда Фурье по собственным функциям второй краевой задачи для уравнения Лапласа. Для получения приближённого устойчивого решения интегрального уравнения применён метод регуляризации Тихонова, когда решение получается как экстремаль функционала Тихонова. Для приближённо заданной поверхности рассматривается вычисление нормали к этой поверхности и её сходимость к точному значению в зависимости от погрешности, с которой задана исходная поверхность. Доказывается сходимость полученного приближённого решения к точному решению при сопоставлении параметра регуляризации с ошибками в данных как по неточно заданной границе, так и по значению исходной функции на этой границе. Проводится численный эксперимент, который демонстрирует эффективность предложенного подхода для частного случая — для плоской границы и конкретного исходного источника тепла (набора точеных источников).

Ключевые слова: некорректно поставленная задача, метод регуляризации Тихонова, задача Коши для уравнения Лапласа, интегральное уравнение первого рода

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 74–102
http://journals.rudn.ru/miph

Review

UDC 004.021:519.2:519.6

DOI: 10.22363/2658-4670-2025-33-1-74-102

EDN: AAAAHV

Analytic projective geometry for computer graphics

Migran N. Gevorkyan[1], Anna V. Korolkova[1], Dmitry S. Kulyabov[1,2], Leonid A. Sevastianov[1,2]

[1] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[2] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: June 25, 2024; revised: August 10, 2024; accepted: August 12, 2024)

Abstract. The motivation of this paper was the development of computer geometry course for students of mathematical specialties. The term “computer geometry” hereafter refers to the mathematical foundations of machine graphics. It is important to emphasize separately that this course should be designed for second-year students and, therefore, they can only be required to have prior knowledge of a standard course in algebra and mathematical analysis. This imposes certain restrictions on the material presented. When studying the thematic literature, it was found out that the de facto standard in modern computer graphics is the use of projective space and homogeneous coordinates. However, the authors faced a methodological problem—the almost complete lack of suitable educational literature in both Russian and English. This paper was written to present the information collected by the authors on this issue.

Key words and phrases: projective geometry, Asymptote system, Plücker coordinates, proper and improper points, lines and planes

For citation: Gevorkyan, M. N., Korolkova, A. V., Kulyabov, D. S., Sevastianov, L. A. Analytic projective geometry for computer graphics. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 74–102. doi: 10.22363/2658-4670-2025-33-1-74-102. edn: AAAAHV (2025).

1. Introduction

Here are the main reasons that motivated us to write this paper.

1.1. Synthetic and analytical approaches to geometry

Historically, there have been two approaches to the presentation of geometry in mathematics, *synthetic* and *analytical*. In the synthetic presentation of geometry, sets of geometric elements of various kinds are initially introduced, such as points, lines and planes. Then the relationship between them is defined, formulated in the form of axioms that correspond to visual geometric representations. This approach is used in a simplified form when presenting Euclidean geometry in a school geometry course, therefore it is intuitively understandable to most students.

An alternative approach emerged later, with the development of algebra. Despite the fact that the name analytical geometry has been assigned to it, it would be more correct to call it *linear-algebraic*, "since linear algebra forms the basis and provides its own methods, not analysis" [1, p. 12]. The linear-algebraic approach is much more general, much more powerful, and therefore much simpler in describing complex structures than the synthetic approach. He received a particularly strong impetus for his development within the framework of the ideas formulated by Felix Klein in his Erlangen program [2].

It is quite natural for computer graphics algorithms to use a linear-algebraic approach, since it allows you to write down and use algebraic formulas to calculate the necessary quantities. Consequently, projective geometry in the framework of this subject should be presented in this style. Note that we are talking about the projective space model $\mathbb{R}P^3$.

1.2. Lack of educational literature

The literature on projective geometry in Russian is extensive, not to mention English and other foreign languages. Most textbooks, where the presentation is conducted at a level accessible to undergraduates of physics and mathematics faculties, are essentially based on a synthetic approach [3–7]. This technique is justified, since the task of these authors is to provide an understanding of the essence of projective geometry, and the analytical approach "requires more ink and less thought" [8, p. 89]. However, when presenting the basics of projective geometry in a computer geometry course, this approach is questionable because it is too far from the final software implementation.

There are also a large number of monographs where projective geometry is presented in a linear-algebraic style, for example [9–11] and a list of sources in [1]. However, the style of presentation in them is rather abstract and most of them are textbooks for undergraduates, graduate students and researchers working in the framework of theoretical mathematics.

Looking at textbooks on computer graphics, machine vision and robotics, then a different problem arises. In many textbooks, projective geometry begins and ends with an exposition of the concept of homogeneous coordinates, which are introduced exclusively in the context of projective transformations. For example, let's list the sources with the pages: [12, pp. 101, 115][13, p. 18][14, pp. 192, 220][15, p. 146] [16, p. 85] [17, p. 20] [18, p. 176] [19, p. 211] [20, p. 438] [21, p. 56]. In all these books, homogeneous coordinates are introduced ad hoc and used to represent affine transformations as a linear transformation (3×3 matrices on the plane and 4×4 matrices in space). The representation of a straight line and a plane using homogeneous coordinates is not considered, and projective geometry is not applied to standard problems of analytical geometry.

The list of textbooks in Russian concerning the mathematical foundations of computer graphics is extremely limited [12–14, 17, 19, 20, 22, 23]. None of these manuals consistently use projective geometry as a tool for solving computer geometry problems. Some information, mainly about homogeneous coordinates, is available in books [12–14, 17, 19, 24], however, they do not consistently describe how to represent straight lines and planes in a homogeneous form, and homogeneous coordinates are also used only to represent affine transformations as linear.

As an exception, the textbook should be noted [25], which adopted a non-standard approach to the presentation of analytic geometry using Grassmann algebra using non-standard notation. We will also mention several sources on the theory of screws [26–28], in which, in particular, the moment of a sliding vector is introduced, which is directly related to the representation of straight lines in Plucker coordinates (in a homogeneous form).

The situation with books in English is only slightly better. You can specify the book [29], where the presentation is not limited to the introduction of homogeneous coordinates only for points, on the contrary, homogeneous coordinates for straight lines and planes are considered, as well as solutions to some standard problems [29, pp. 25, 65]. As a disadvantage, we note that all formulas are written mainly in component rather than vector form, and we also note the focus on the field of computer vision rather than computer graphics. These are different areas, despite their proximity.

Of particular note is the extremely capacious but extremely informative book [30] by Eric Lengyel, which stands out in several ways.

- The presentation is conducted at a good mathematical level, but with an emphasis on practical application. For almost every formula, an example of its implementation is given in the form of programs in C-like pseudocode.
- Due attention is paid to the application of the principles of projective geometry in computer graphics problems. In the third chapter, the author provides an extremely useful table 3.1 with a summary of basic formulas using homogeneous coordinates for points, lines and planes.
- The fourth chapter is devoted to Grassmann algebra, which is the basis of geometric algebra. The author gives a description of projective spaces in terms of n–vectors.

This concludes the list of textbooks found by the authors, where analytical projective geometry is somehow touched upon.

1.3. Paper structure

In this article and its continuation, we attempt to eliminate this drawback and provide a detailed description of analytic projective geometry, or rather the model of the projective space $\mathbb{R}\mathrm{P}^3$. At the same time, we focus on practical applications in the field of computer graphics and implement all proven formulas programmatically in the language Asymptote [31–33]. The article contains a large number of drawings and all of them are created programmatically using Asymptote. Writing points, lines, and planes in a projective form allows you to associate each of these geometric entities with a certain algebraic entity.

In the first article, we describe the theory in detail, based on the notation from the book [30]. To prove the formulas, we use both projective space and three-dimensional Euclidean space with a Cartesian coordinate system. For the sake of completeness, we repeat some things from classical analytic geometry, but they are also interpreted within the framework of projective space. We write down all the proven formulas in the table, which is an expanded and supplemented version of the above-mentioned Table 3.1. This table allows you to solve any problem about the relative position of points, lines and planes.

In the second article, the proven formulas will be translated into the Asymptote programming language. This language is designed to create two-dimensional and three-dimensional vector illustrations, has a C-like syntax, and allows you to set custom data structures by attaching member functions (methods) to them and overloading existing functions for these structures. This made it possible to create structures for a projective point, a straight line, and a plane and to implement standard tasks for studying their relative positions. The results are immediately visualized.

2. Projective space and homogeneous coordinates

In art, the concept of *perspective* (Latin *perspicere* - to look through) has emerged since ancient times as a technique for depicting spatial objects in accordance with the distortion of proportions and shapes of depicted bodies during their visual perception. Visual means for conveying perspective

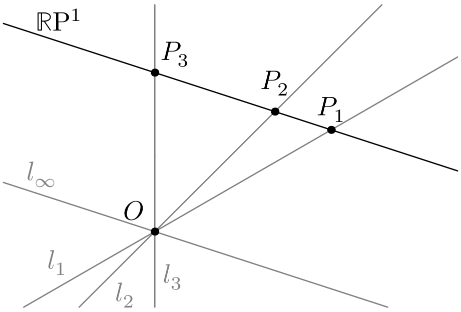


Figure 1. Real projective line $\mathbb{R}\mathrm{P}^1$. The points on the projective line correspond to the lines of the space $\mathbb{R}^2$ passing through the origin O. An irregular (ideal) point corresponds to a parallel line l_∞

were initially passed on as practical skills in the fine arts from experienced craftsmen to apprentices. The mathematical foundations of perspective were laid by the French geometer and architect Girard Desargues (1591–1661), under the name *projective geometry*. Later, one of the main theorems of which bears his name (Desargues' theorem) was named after him. Further contributions to the development of projective geometry were made by Jean-Victor Poncelet, M. Chasles, K. G. H. von Staudt, A. F. Möbius, J. Plücker, F. H. Klein. For more information about the history of the development of projective geometry, see the popular book [34].

A projective space or, more precisely, a model of the projective space $\mathbb{R}\mathrm{P}^n$ of dimension n over the field of real numbers $\mathbb{R}$ is a set of straight lines in an ordinary Euclidean affine space $\mathbb{R}^{n+1}$ passing through the origin point O. Three-dimensional computer graphics uses the projective space model $\mathbb{R}\mathrm{P}^3$. This model is based on the four-dimensional Euclidean space $\mathbb{R}^4$, which greatly limits visibility due to the natural complexity of illustrating four-dimensional spaces. Therefore, for greater clarity, we will give examples of the spaces $\mathbb{R}\mathrm{P}^1$ and $\mathbb{R}\mathrm{P}^2$ for which illustrations can be given on a plane and in three-dimensional space, respectively.

2.1. The real projective line

A model of a projective line, that is, a space of dimension 1, can be constructed if we define a certain line $\mathbb{R}\mathrm{P}^1$ on the plane $\mathbb{R}^2$ that does not pass through the origin. Each point P of this straight line will correspond to a straight line in the plane that passes through the origin and intersects $\mathbb{R}\mathrm{P}^1$ at the point P, as shown in the figure 1. Such points are called *endpoints* or *proper* points. The only straight line l_∞ passing through O and parallel to $\mathbb{R}\mathrm{P}^1$ will correspond to a point of a special kind called *improper* or *ideal point*.

Coordinates can be entered on the projective line. The most convenient way to do this is to draw a projective line $\mathbb{R}\mathrm{P}^1$ parallel to the Ox axis, through the point $(0, w)$ as shown in the figure 2. Since each projective point is defined by a straight line passing through the origin, the components of the non-normalized guide vector of this line can be taken as the projective coordinates, for example, $P = (x, w)$. When multiplying the components of the guide vector by the same number $\lambda \neq 0$, it will still set the same straight line, so you can write $P = (x, w) = (\lambda x, \lambda w)$. Therefore, the coordinates are a pair of numbers considered up to the proportionality of x/w. To emphasize this fact, the coordinates are written using a colon as a separator — the division symbol $(x : w)$. You can choose a fixed number as w. Traditionally, $w = 1$ is taken and such coordinates are called *homogeneous* or *homogeneous*. Points with coordinates like $(x : 1)$ correspond to proper points, and points of the form $(x : 0)$ are non-proper (in the one-dimensional case, there is one such point corresponding to the line Ox).

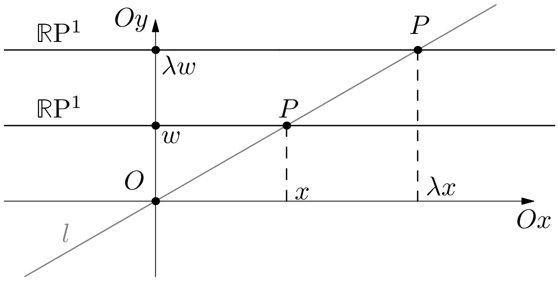


Figure 2. A homogeneous coordinate system for a one-dimensional projective space. The coordinates of the point are determined by the guide vector $\boldsymbol{v} = (x, w)^T = (\lambda x, \lambda)^T$ of the straight line l. In homogeneous coordinates, they are written as $\vec{\boldsymbol{v}} = (x : w)$

2.2. The real projective plane

The projective plane can be defined axiomatically [5, 35]. Thus, a set is called a projective plane, the elements of which are called points. It identifies subsets called straight lines and requires the following properties to be fulfilled.

1. For any two different points, there is a single line containing them.
2. Any two different lines have a single point in common.
3. There are four points, none of which lie on the same straight line.

These axioms define the projective plane in an non-constructive way, since they say nothing about how to construct a specific set of objects that satisfy these axioms — *model* of an abstract object.

Consider *real projective plane* $\mathbb{R}\mathrm{P}^2$ and introduce homogeneous coordinates on the projective plane. Such a space can be represented as a plane in a three-dimensional Euclidean affine space with a Cartesian coordinate system. The plane $\mathbb{R}\mathrm{P}^2$ passes through a point $(0, 0, 1)$, parallel to the Ox and Oy axes. As before, each endpoint of the plane corresponds to a straight line passing through the origin O and intersecting the plane at this point, as shown in the figure 3. The coordinates of all points on the plane have the form $(x, y, 1)$ and correspond to the homogeneous coordinates $(x : y : 1)$ of the point P.

Unlike a projective line, there are an infinite number of ideal points on the projective plane. Each such point corresponds to a straight line lying in the Oxy plane. These include, for example, the Ox and Oy axes themselves. The guide vectors of such lines have components of the form $(x, y, 0)$ and define the *direction* on the projective plane.

Due to this, two types of vectors are distinguished in the projective space $\mathbb{R}\mathrm{P}^2$ with component notation.

- Radius vectors or *point vectors* — vectors anchored to the origin, having homogeneous coordinates of the form $(x : y : 1)$ and defining endpoints on the plane.
- Direction vectors — loose, free vectors having homogeneous coordinates of the form $(x : y : 0)$ and defining ideal points lying at infinity.

In addition to points, straight lines are also present on the projective plane. Each straight line corresponds to a plane passing through the origin and intersecting $\mathbb{R}\mathrm{P}^2$ along a certain straight line, as shown in the figure 4. The plane Oxy parallel to $\mathbb{R}\mathrm{P}^2$ corresponds to a perfect straight line on which all ideal points lie. A model of an ideal straight line can be a circle lying infinitely far away in the plane $\mathbb{R}\mathrm{P}^2$.

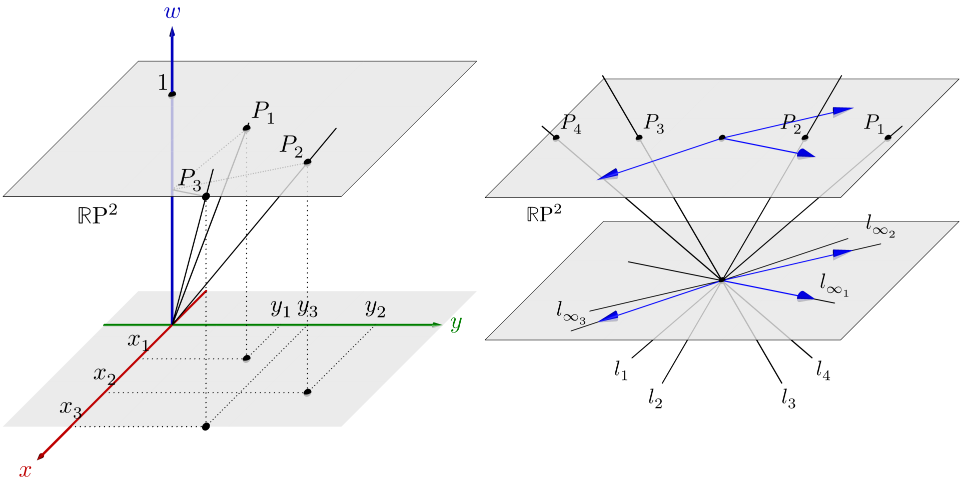


Figure 3. The model of the projective plane $\mathbb{R}P^2$. Each point in the plane corresponds to a line of $\mathbb{R}^3$ space passing through the origin. The z coordinate is traditionally denoted by the symbol w

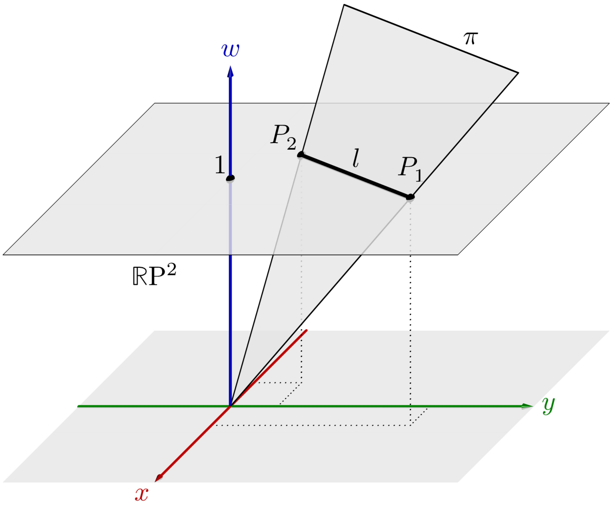


Figure 4. The model of the projective plane $\mathbb{R}P^2$. Each straight plane corresponds to a plane of space $\mathbb{R}^3$ passing through the origin. In this case, the plane is shown as a triangle

2.3. A model of three-dimensional projective space $\mathbb{R}\mathrm{P}^3$

Finally, let's consider the model of the projective space $\mathbb{R}\mathrm{P}^3$. Unfortunately, this model cannot be fully visualized, since it uses the four-dimensional space $\mathbb{R}^4$, where a Cartesian coordinate system with the axes Ox, Oy, Oz and Ow is introduced. Some intuitive understanding can be gained if we draw analogies with the projective plane, instead of which a hyperplane of dimension 3 is drawn through the point $w = 1$, which we will identify with $\mathbb{R}\mathrm{P}^3$.

- Endpoints in $\mathbb{R}\mathrm{P}^3$ correspond to lines intersecting $\mathbb{R}\mathrm{P}^3$, and ideal points correspond to lines parallel to $\mathbb{R}$.

- Finite lines correspond to planes intersecting $\mathbb{R}\mathrm{P}^3$, and ideal lines correspond to planes parallel to $\mathbb{R}\mathrm{P}^3$.
- The finite planes correspond to hyperplanes of dimension 3 intersecting $\mathbb{R}\mathrm{P}^3$ along planes (of dimension 2). The ideal plane corresponds to the hyperplane $Oxyz$ parallel to $\mathbb{R}\mathrm{P}^3$.

Some idea of a hyperplane of dimension 3 can be obtained if we imagine that an ordinary plane (of dimension 2) is intersected by some other plane that cuts off a straight line on the plane, that is, a geometric object one dimension smaller than itself. Similarly, a hyperplane of dimension 3 intersects another hyperplane of dimension 3 and cuts off the usual planes of dimension 2 on it, which in some way are located in “volume” $\mathbb{R}\mathrm{P}^3$.

Although a full-fledged visual representation is impossible in this case, it is still possible to work with $\mathbb{R}\mathrm{P}^3$ using algebra, in particular using homogeneous coordinates. Each point in this space can be matched in the same way as for $\mathbb{R}\mathrm{P}^1$ and $\mathbb{R}\mathrm{P}^2$ with homogeneous coordinates of the form $(x : y : z : 1)$ in the case of endpoints and the form $(x : y : z : 0)$ for ideal points. Unlike the projective plane, $\mathbb{R}\mathrm{P}^3$ contains planes and infinitely many ideal straight lines, all of which lie on an infinitely distant ideal plane, which can be conventionally modeled as a sphere with infinite radius.

To distinguish the vectors of the projective space $\mathbb{R}\mathrm{P}^3$ from the vectors of the Euclidean affine space $\mathbb{R}^3$, we will use bold font with the addition of an arrow icon at the top, for example:

$$\vec{\boldsymbol{p}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = (\boldsymbol{p} \mid 1) = (x : y : z : 1), \quad \vec{\boldsymbol{q}} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = (\boldsymbol{q} \mid w) = (x : y : z : w).$$

A point in $\mathbb{R}\mathrm{P}^3$ is considered normalized if $w = 1$. Geometrically, this means that it lies in a hyperplane drawn through the fourth axis Ow.

The introduction of homogeneous coordinates into the projective space model makes it possible to represent not only points, but also straight lines and planes in linear algebraic form. In other words — construct *projective analytic geometry*. The linear-algebraic representation of lines and planes with the introduced homogeneous coordinates will be called *homogeneous representation* of lines and planes. For the case of a straight line, the terms *Plucker representation* of a straight line or *Plucker coordinates* of a straight line are also used. Both of these terms in this article will be synonymous with the homogeneous representation of a straight line. For the case of a plane, the term Plucker coordinates is not usually used.

3. A line on plane

Before proceeding to the representation of a line in a projective form, let us list the main ways of algebraic representation of a line on an ordinary Euclidean plane. These methods are studied in standard analytical geometry courses, so we will focus only on the main points that are important for the topic of the article. For additional information, we refer the reader, for example, to [36, Chapter 5, §1] and [37].

Consider a line l on a plane. Let’s use the letter P to denote an arbitrary point on a line with the radius vector $\boldsymbol{p} = (x, y)^T$. Let us know some fixed point P_0 with a radius vector $\boldsymbol{p}_0 = (x_0, y_0)^T$, and the guiding vector $\boldsymbol{v}$ is also specified. You can write *parametric equation* in a line l:

$$\boldsymbol{r}(t) = \begin{bmatrix} x \\ y \end{bmatrix} = \boldsymbol{p}_0 + \boldsymbol{v}t = \begin{bmatrix} x_0 + v_x t \\ y_0 + v_y t \end{bmatrix}.$$

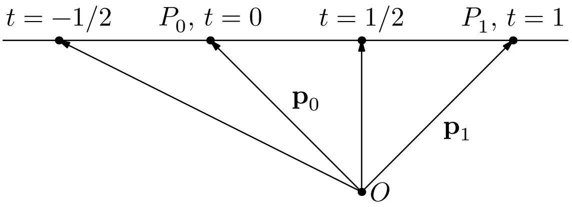


Figure 5. Illustration of the linear interpolation process. By setting the segment P_1P_2, you can move between points P_1 and P_2 by changing the parameter from 0 to 1. With parameter values outside the segment $[0,1]$, the point will slide along a line (P_1P_2) going beyond the segment

Where $\boldsymbol{r}(t)$ is the radius vector of a line point, t is a parameter running through all real numbers. The guiding vector $\boldsymbol{v}$ can be geometrically interpreted as a tangent vector to a line that lies on l, since the tangent to the l line coincides with the line itself.

The guide vector can be calculated if any two points of a line are known, for example P_0 and P_1, then $\boldsymbol{v} = \boldsymbol{p}_1 - \boldsymbol{p}_0$. The order of the radius vectors in the difference affects the direction of movement of the point. Usually, a guiding vector is chosen so that as t increases, the point moves from left to right, and as it decreases from right to left.

Substitute $\boldsymbol{v} = \boldsymbol{p}_1 - \boldsymbol{p}_0$ into the parametric equation and rearrange the terms:

$$\boldsymbol{r}(t) = \boldsymbol{p}_0 + (\boldsymbol{p}_1 - \boldsymbol{p}_0)t = \boldsymbol{p}_0(1-t) + \boldsymbol{p}_1 t.$$

In this form, the equation of a line defines the segment between the points P_0 and P_1, if $0 \leqslant t \leqslant 1$, as shown in the figure 5. The term *lerp* is well-established in computer geometry from the phrase linear interpolation (**l**inear int**erp**olation). This is the simplest version of interpolation, but it is very widely used in animation, curve drawing, etc.

The guide vector $\boldsymbol{v}$ can have an arbitrary length, since it is enough to specify any two points of a line P_1 and P_2 to calculate it, however, it is convenient to use a single guide vector $\|\boldsymbol{v}\| = 1$ for calculations, which can always be obtained by reparametrization:

$$\boldsymbol{r}(s) = \boldsymbol{p}_0 + \boldsymbol{v}\frac{t}{\|\boldsymbol{v}\|} = \boldsymbol{p}_0 + \boldsymbol{v}s, \quad s = \frac{t}{\sqrt{v_x^2 + v_y^2}}.$$

The parameter s is called *natural parameter* and is interpreted as the length of a line measured from some fixed point, in this case from P_0, since for $s = 0$ we get $r(0) = \boldsymbol{p}_0$. Sometimes the letter l is used to denote a natural parameter.

Directly from the parametric equation, one can obtain the *canonical equation* of a line:

$$\left.\begin{aligned} t &= \frac{x - x_0}{v_x} \\ t &= \frac{y - y_0}{v_y} \end{aligned}\right\} \Rightarrow \frac{x - x_0}{v_x} = \frac{y - y_0}{v_y}.$$

If we put $\boldsymbol{v} = (x_1 - x_0, y_1 - y_0)T$, then we can write the equation of a line passing through two points P_0 and P_1:

$$\frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0}.$$

Now consider a normal drawn to a line at some point and having a guiding vector $\boldsymbol{N} = (A, B)$. By definition, a normal is perpendicular to a line, hence $\boldsymbol{N} \perp \boldsymbol{v} = \boldsymbol{p} - \boldsymbol{p}_0$:

$$(\boldsymbol{N}, \boldsymbol{p} - \boldsymbol{p}_0) = 0 \Leftrightarrow (\boldsymbol{N}, \boldsymbol{p}) - (\boldsymbol{N}, \boldsymbol{p}_0) = 0 \Leftrightarrow Ax + By - Ax_0 - By_0 = 0.$$

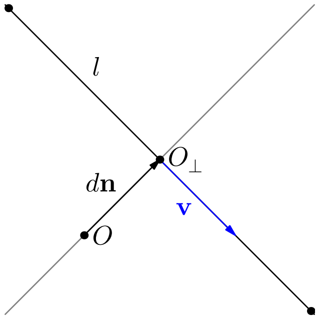


Figure 6. The geometric meaning of the normal equation of a line on a plane. The figure shows a line l, its guiding vector $\boldsymbol{v}$, the normal vector $d\boldsymbol{n}$, where d is the directional distance from the origin to the line or, in other words, the length of the perpendicular $\boldsymbol{OO}_\perp$

Denoting $C = -Ax_0 - By_0$, we write *general equation* of a line on a plane:

$$Ax + By + C = 0.$$

It can also be rewritten in a different form by dividing all terms by $-C$ and writing:

$$-\frac{A}{C}x - \frac{B}{C}y - 1 = 0 \Leftrightarrow \frac{x}{-C/A} - \frac{y}{-C/B} = 1 \Rightarrow \frac{x}{a} - \frac{y}{b} = 1, \quad a = -C/A, \quad b = -C/B.$$

This form of the equation is called the *equation in segments*.

In the equation $Ax + By + C = 0$, we divide all terms by the norm of the guide vector of the normal $\|\boldsymbol{N}\| = \sqrt{A^2 + B^2}$ and write:

$$\frac{A}{\sqrt{A^2+B^2}}x + \frac{B}{\sqrt{A^2+B^2}}y + \frac{C}{\sqrt{A^2+B^2}} = 0, \quad \boldsymbol{n} = \left(\frac{A}{\sqrt{A^2+B^2}}, \frac{B}{\sqrt{A^2+B^2}}\right) = (n_x, n_y).$$

We have introduced the notation $\boldsymbol{n}$ for the unit direction vector of the normal (ort of the normal) to the line. In addition, since

$$\frac{C}{\sqrt{A^2+B^2}} = \frac{-Ax_0 - By_0}{\sqrt{A^2+B^2}} = -\frac{A}{\sqrt{A^2+B^2}}x_0 - \frac{B}{\sqrt{A^2+B^2}}y_0 = -(\boldsymbol{n}, \boldsymbol{p}_0) = d.$$

as a result, the general equation of the line is written as

$$n_x x + n_y y + d = 0.$$

In this form, the equation of a line is called *normalized equation*. The coefficient d has the geometric meaning of the distance from the line to the origin. In general, the distance from a point to a line is defined as the length of the perpendicular lowered from the point to the line. The direction of the perpendicular coincides with the direction of the normal vector, since the normal is also perpendicular to the line.

The components of the unit normal vector can be written using trigonometric functions. Since $\|\boldsymbol{n}\|^2 = n_x^2 + n_y^2 = 1$, there will always be a θ such that $n_x = \cos\theta$, $n_y = \sin\theta$, where $\theta = \measuredangle(\boldsymbol{e}_x, \boldsymbol{n})$ — the value of the angle between the ort $\boldsymbol{e}_x$ of the Ox axis and the vector $\boldsymbol{n}$. For an unnormalized vector $\boldsymbol{N}$, you can also write $\boldsymbol{N} = \|\boldsymbol{N}\|(\cos\theta, \sin\theta)$ or else $A = \|\boldsymbol{N}\|\cos\theta$, $B = \|\boldsymbol{N}\|\sin\theta$.

It should be emphasized here that the vector $\boldsymbol{n}$ is a free vector that defines the direction of the normal. When visualizing $\boldsymbol{n}$, it can be moved both along the normal and the normal itself can be moved along a line. The specific image method depends on what exactly we want to illustrate. For example, in the figure 6, the vector $d\boldsymbol{n}$ is shifted away from the origin, since it indicates the point of the plane closest to the origin.

It is interesting to note that the equation $Ax + By + D = 0$ is already a projective equation of a line on the plane $\mathbb{RP}^2$, since within the framework of this model, lines on the projective plane are three-dimensional planes passing through the origin, which are given by the equation $Ax + By + Dw = 0$. If we put $w = 1$, we get just the general equation of the line. For a line on the projective plane, the term Plucker coordinates is not used, but their analog is the numbers A, B, C or in normalized form n_x, n_y, d.

4. A line in space

4.1. The parametric equation

Here we briefly recall how the *parametric equation* of a line is written in Euclidean space, and immediately proceed to write the Plucker coordinates of a line, that is, the linear algebraic representation of a line in a projective space. For information about the canonical equation of a line, we refer the reader to [36, Ch. 5, §4].

Let there also be some point P_0 lying on the line l and the guiding vector of the line $\boldsymbol{v}$, then the parametric equation will be written as

$$\boldsymbol{l}(t) = \boldsymbol{p}_0 + \boldsymbol{v}t = \begin{cases} x_0 + v_x t, \\ y_0 + v_y t, \\ z_0 + v_z t. \end{cases}$$

Where $\boldsymbol{l}(t)$ is the radius vector of an arbitrary point P belonging to a line. By writing the guiding vector through the radius of the vectors of the two points of the line, we obtain *canonical equation*:

$$\boldsymbol{p} - \boldsymbol{p}_0 = (\boldsymbol{p}_1 - \boldsymbol{p}_0)t \Rightarrow \frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0} = \frac{z - z_0}{z_1 - z_0}.$$

Just as for the plane case, a formula can be derived from the parametric equation for linear interpolation of a segment lying between points with radius vectors $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$:

$$\boldsymbol{l}(t) = \boldsymbol{p}_0(1 - t) + \boldsymbol{p}_1 t.$$

The question arises, is it possible to write down the equivalent of the normal equation of a line on a plane for the three-dimensional case? Even from visual geometric considerations, it becomes clear that it is not possible to do this, since the distance from the origin and the guiding vector of the normal do not uniquely define a line in space, unlike in the flat case.

4.2. The moment and coordinates of the Plucker line in space

Consider a line l, the points of which are set parametrically using the radius vector $\boldsymbol{l}(t) = \boldsymbol{p} + \boldsymbol{v}t$, where $\boldsymbol{p}$ is the radius vector of some point P belonging to the line l, and $\boldsymbol{v}$ is the guiding vector of the line. Let's introduce the vector $\boldsymbol{m}$, which we define as

$$\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{v}$$

and let's call *the moment of the line l.*

It can be shown that the moment $\boldsymbol{m}$ does not depend on the choice of a point on a line. To do this, find the vector product $\boldsymbol{l}(t) \times \boldsymbol{v}$:

$$\boldsymbol{l}(t) \times \boldsymbol{v} = (\boldsymbol{p} + \boldsymbol{v}t) \times \boldsymbol{v} = \boldsymbol{p} \times \boldsymbol{v} + \boldsymbol{v} \times \boldsymbol{v}t = \boldsymbol{p} \times \boldsymbol{v}.$$

Therefore, the moment $\boldsymbol{m}$ characterizes a line and does not depend on the choice of a point on it, but depends on the guiding vector $\boldsymbol{v}$. In this case, the vector $\boldsymbol{v}$ is *sliding*, that is, its origin is not tied to a single point and can move along the line l in any direction, including in the opposite direction. To emphasize the importance of the vector $\boldsymbol{v}$, the following terminology is used: the point P is called the *application point* of the vector $\boldsymbol{v}$, and the line itself l is called the *line vector* $\boldsymbol{v}$. We also note that the moment $\boldsymbol{m}$ is the main object in the theory of screws [26, 28].

The line is completely characterized by its guiding vector $\boldsymbol{v}$ and the moment $\boldsymbol{m}$. These two vectors allow you to define a line without reference to a specific point in space, as is done in the two-dimensional case using the normal equation of a line. The six parameters $\{v_x, v_y, v_z \mid m_x, m_y, m_z\}$ are called *Plucker coordinates* of a line in honor of Julius Plucker (1801-1868) [38, Ch. 3, §3] [29, 30]. The Plucker coordinates of a line are a linear algebraic representation of a line within a system of homogeneous coordinates, and just like homogeneous coordinates, they are defined up to a common factor. As a result, we get the opportunity to define any line in a homogeneous form.

$$\{\boldsymbol{v} \mid \boldsymbol{m}\} = \{v_x, v_y, v_z \mid m_x, m_y, m_z\} = \{\boldsymbol{v} \mid \boldsymbol{p} \times \boldsymbol{v}\}. \tag{1}$$

The notation for writing Plucker coordinates in the form of curly brackets is used in the book [30], and we will follow them in this tutorial as well.

The moment of a line can also be calculated using two points $P_1, P_2 \in l$ with radius vectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. To do this, note that $\boldsymbol{p}_2 - \boldsymbol{p}_1$ is the guiding vector of a line, hence

$$\boldsymbol{m} = \boldsymbol{p}_1 \times (\boldsymbol{p}_2 - \boldsymbol{p}_1) = \boldsymbol{p}_1 \times \boldsymbol{p}_2 - \boldsymbol{p}_1 \times \boldsymbol{p}_1 = \boldsymbol{p}_1 \times \boldsymbol{p}_2 \Longrightarrow$$

$$\boldsymbol{m} = \boldsymbol{p}_1 \times \boldsymbol{p}_2 \tag{2}$$

This entry corresponds to the formula (B) of the table 1.

Emphasis should be placed on the fact that the moment $\boldsymbol{m}$ and the guiding vector $\boldsymbol{v}$ are mutually orthogonal, which follows directly from our definition of the moment as the vector product of $\boldsymbol{p} \times \boldsymbol{v}$. Ratio

$$(\boldsymbol{v}, \boldsymbol{m}) = v_x m_x + v_y m_y + v_z m_z = 0,$$

It is called *Plucker equations* or *Plucker condition*. It imposes a restriction on the parameters $v_x, v_y, v_z, m_x, m_y, m_z$ so that not every six numbers can set a line, but only those for which the specified equality holds. The condition itself is a second-order algebraic expression, which means that four parameters are sufficient to define a line.

If the coordinates of the two points through which the line l passes are given in a homogeneous form, that is, P_1 as $(\boldsymbol{p}_1 \mid w_1)$ and P_2 as $(\boldsymbol{p}_2 \mid w_2)$, then normalize the coordinates of the point and calculate the guide vector, which is written as $\boldsymbol{v} = \boldsymbol{p}_2/w_2 - \boldsymbol{p}_1/w_1$, $w_1 w_2 \boldsymbol{v} = w_1 \boldsymbol{p}_2 - w_2 \boldsymbol{p}_1$. The moment of the line is calculated as follows:

$$\frac{\boldsymbol{p}_1}{w_1} \times w_1 w_2 \boldsymbol{v} = \frac{\boldsymbol{p}_1}{w_1} \times (w_1 \boldsymbol{p}_2 - w_2 \boldsymbol{p}_1) = \boldsymbol{p}_1 \times \boldsymbol{p}_2 - \frac{w_2}{w_1} \boldsymbol{p}_1 \times \boldsymbol{p}_1 = \boldsymbol{p}_1 \times \boldsymbol{p}_2 = \boldsymbol{m}.$$

Therefore, in a homogeneous form, the line can be written as

$$\{w_1 \boldsymbol{p}_2 - w_2 \boldsymbol{p}_1 \mid \boldsymbol{p}_1 \times \boldsymbol{p}_2\}, \tag{3}$$

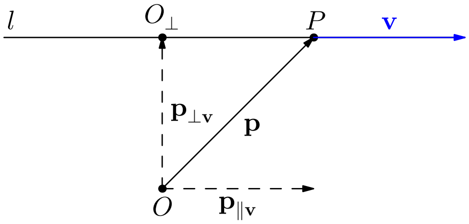


Figure 7. The distance from the origin O to the line l

which corresponds to formula (D) from table 1.

This formula allows you to calculate the Plucker coordinates of a line passing through non-matching points $(\boldsymbol{u}_1 \mid 0)$ and $(\boldsymbol{u}_2 \mid 0)$:

$$\{\mathbf{0} \mid \boldsymbol{u}_1 \times \boldsymbol{u}_2\}.$$

Such a line is called *improper line* or *ideal line*. In the visual arts, the horizon line corresponds to such line. The guiding vector of an improper line, therefore, is equal to the zero vector $\mathbf{0}$, and the moment can be nonzero.

4.3. Distance from a point to a line

Consider the problem of finding the distance from the origin point O to the line l, for which we must find the radius vector $\boldsymbol{OO}_\perp$, where the point $O_\perp$ is the base of the perpendicular omitted from O onto the line l as shown in the figure 7. The vector $\boldsymbol{OO}_\perp$ is found as the perpendicular part of the radius of the vector $\boldsymbol{p}$ relative to the guiding vector $\boldsymbol{v}$ of the line. Note also that the choice of the point of the line P is arbitrary. Using the formula $\boldsymbol{a} \times \boldsymbol{b} \times \boldsymbol{c} = \boldsymbol{b}(\boldsymbol{a}, \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a}, \boldsymbol{b})$ for a triple vector product, write an expression for $\boldsymbol{OO}_\perp$

$$\boldsymbol{OO}_\perp = \boldsymbol{p}_{\perp\boldsymbol{v}} = \boldsymbol{p} - \boldsymbol{p}_{\|\boldsymbol{v}} = \boldsymbol{p} - \frac{(\boldsymbol{p}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}\boldsymbol{v} = \frac{\boldsymbol{p}\|\boldsymbol{v}\|^2 - (\boldsymbol{p}, \boldsymbol{v})\boldsymbol{v}}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v} \times \boldsymbol{p} \times \boldsymbol{v}}{\|\boldsymbol{v}\|^2}.$$

Substitute $\boldsymbol{p} \times \boldsymbol{v} = \boldsymbol{m}$ and write:

$$\boldsymbol{OO}_\perp = \frac{\boldsymbol{v} \times \boldsymbol{p} \times \boldsymbol{v}}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v} \times \boldsymbol{m}}{\|\boldsymbol{v}\|^2}.$$

You can write the coordinates of the point $O_\perp$ in a homogeneous form. Since the radius vector is known, the homogeneous coordinates will look like:

$$\left(\frac{\boldsymbol{v} \times \boldsymbol{m}}{\|\boldsymbol{v}\|^2} \,\Big|\, 1\right) = (\boldsymbol{v} \times \boldsymbol{m} \mid (\boldsymbol{v}, \boldsymbol{v})), \tag{4}$$

which proves the formula (G) from the table 1.

Considering that $\boldsymbol{v} \perp \boldsymbol{m}$ by virtue of the definition of the vector product, we calculate the length of the vector $\boldsymbol{OO}_\perp$

$$d_{OO_\perp} = \|\boldsymbol{OO}_\perp\| = \frac{\|\boldsymbol{v} \times \boldsymbol{m}\|}{\|\boldsymbol{v}\|} = \frac{\|\boldsymbol{v}\|\|\boldsymbol{m}\| \sin\frac{\pi}{2}}{\|\boldsymbol{v}\|^2} = \frac{\|\boldsymbol{m}\|}{\|\boldsymbol{v}\|}. \tag{5}$$

Note that if a straight line passes through the origin, then the distance $d_{OO_\perp} = 0$ and $\|\boldsymbol{m}\|/\|\boldsymbol{v}\| = 0$, therefore $\boldsymbol{m} = 0$. In Plucker coordinates, such a straight line is written as

$$\{\boldsymbol{v} \mid 0\}. \tag{6}$$

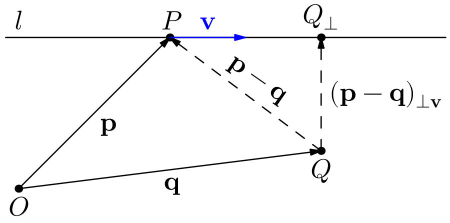


Figure 8. The distance from an arbitrary point Q to a straight line l. Vector $\boldsymbol{u} = \boldsymbol{p} - \boldsymbol{q}$

Indeed, a straight line passing through O is completely determined by its guiding vector, since in parametric form the radius vector of its points is written as $\boldsymbol{l}(t) = \boldsymbol{0} + \boldsymbol{v}t = \boldsymbol{v}t$.

Now let's complicate the problem and calculate the distance from an arbitrary point $Q \notin l$ to the straight line l. To do this, we find the length of the segment $QQ_\perp$, where $Q_\perp$ is the foot of the perpendicular dropped from the point Q onto the straight line l as this is shown in the figure 8.

We also use the radius vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ of the points P and Q, respectively, as well as the vector $\boldsymbol{u} = \boldsymbol{p} - \boldsymbol{q}$. The perpendicular vector $\boldsymbol{QQ}_\perp$ is calculated as the perpendicular part of the vector $\boldsymbol{u}$ relative to the guiding vector of the straight line $\boldsymbol{v}$. You can enter the moment of a straight line relative to the point Q as follows:

$$\boldsymbol{m}_Q = (\boldsymbol{p} - \boldsymbol{q}) \times \boldsymbol{v} = \boldsymbol{p} \times \boldsymbol{v} - \boldsymbol{q} \times \boldsymbol{v} = \boldsymbol{m}_O - \boldsymbol{q} \times \boldsymbol{v},$$

where $\boldsymbol{m}_O$ is the moment of a straight line relative to the origin. Note that

$$(\boldsymbol{m}_Q, \boldsymbol{v}) = (\boldsymbol{m}_O, \boldsymbol{v}) - (\boldsymbol{q} \times \boldsymbol{v}, \boldsymbol{v}) = 0, \text{ так как } \boldsymbol{m}_O \perp \boldsymbol{v} \text{ и } \boldsymbol{v} \perp \boldsymbol{q} \times \boldsymbol{v},$$

therefore, the moment $\boldsymbol{m}_Q$ is orthogonal to the vector $\boldsymbol{v}$.

Using the moment $\boldsymbol{m}_Q$, we can calculate the vector $\boldsymbol{QQ}_\perp$

$$\boldsymbol{QQ}_\perp = (\boldsymbol{p} - \boldsymbol{q})_{\perp\boldsymbol{v}} = (\boldsymbol{p} - \boldsymbol{q}) - \frac{(\boldsymbol{p} - \boldsymbol{q}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}\boldsymbol{v} = \frac{\boldsymbol{v} \times (\boldsymbol{p} - \boldsymbol{q}) \times \boldsymbol{v}}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v} \times \boldsymbol{m}_Q}{\|\boldsymbol{v}\|^2}.$$

To find the length of the vector $\boldsymbol{QQ}_\perp$, we write it as follows

$$\boldsymbol{QQ}_\perp = (\boldsymbol{p} - \boldsymbol{q}) - (\boldsymbol{p} - \boldsymbol{q})_{\|\boldsymbol{v}} = \boldsymbol{u} - \boldsymbol{u}_{\|\boldsymbol{v}}$$

and find the square of the norm $\|\boldsymbol{p} - \boldsymbol{q}\|$

$$\|\boldsymbol{p} - \boldsymbol{q}\|^2 = (\boldsymbol{u} - \boldsymbol{u}_{\|\boldsymbol{v}}, \boldsymbol{u} - \boldsymbol{u}_{\|\boldsymbol{v}}) = (\boldsymbol{u}\boldsymbol{u}) - (\boldsymbol{u}, \boldsymbol{u}_{\|\boldsymbol{v}}) - (\boldsymbol{u}_{\|\boldsymbol{v}}, \boldsymbol{u}) + (\boldsymbol{u}_{\|\boldsymbol{v}}, \boldsymbol{u}_{\|\boldsymbol{v}}) = \|\boldsymbol{u}\|^2 + \left\|\boldsymbol{u}_{\|\boldsymbol{v}}\right\|^2 - 2(\boldsymbol{u}, \boldsymbol{u}_{\|\boldsymbol{v}})$$

Since $\boldsymbol{u}_{\|\boldsymbol{v}} = \dfrac{(\boldsymbol{u}, \boldsymbol{v})\boldsymbol{v}}{\|\boldsymbol{v}\|^2}$, then

$$(\boldsymbol{u}, \boldsymbol{u}_{\|\boldsymbol{v}}) = \frac{(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}(\boldsymbol{u}, \boldsymbol{v}) = \frac{(\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^2},$$

$$\left\|\boldsymbol{u}_{\|\boldsymbol{v}}\right\|^2 = \left(\frac{(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}\boldsymbol{v}, \frac{(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}\boldsymbol{v}\right) = \frac{(\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^4}(\boldsymbol{v}, \boldsymbol{v}) = \frac{(\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^4}\|\boldsymbol{v}\|^2 = \frac{(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2}.$$

Let now substitute this formula into the expression for $\|\boldsymbol{u}\|^2$ to get:

$$\|\boldsymbol{u}\|^2 = \|\boldsymbol{u}\|^2 + \frac{(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|^2} - 2\frac{(\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^2} = \|\boldsymbol{u}\|^2 - \frac{(\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^2} = \frac{\|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2 - (\boldsymbol{u}, \boldsymbol{v})^2}{\|\boldsymbol{v}\|^2} = \frac{\|\boldsymbol{u} \times \boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2}.$$

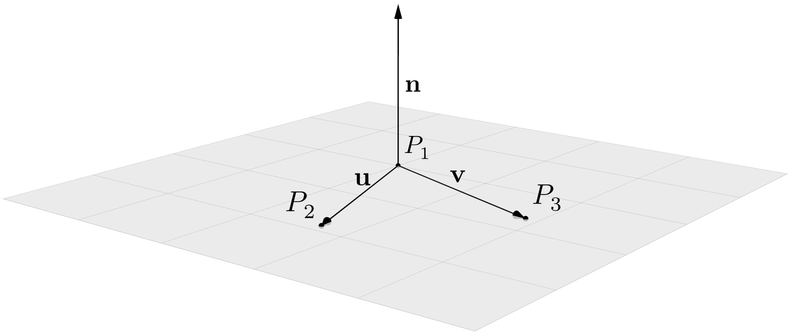


Figure 9. The plane π and its guide vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. The normal vector $\boldsymbol{n}$ is perpendicular to the guide vectors

In the last step, we again used the Lagrange identity of the vector product. Since $\boldsymbol{u} \times \boldsymbol{v} = (\boldsymbol{p} - \boldsymbol{q}) \times \boldsymbol{v} = \boldsymbol{p} \times \boldsymbol{v} - \boldsymbol{q} \times \boldsymbol{v} = \boldsymbol{m} - \boldsymbol{q} \times \boldsymbol{v} = \boldsymbol{m} + \boldsymbol{v} \times \boldsymbol{q}$, then we got the formula for calculating the distance from an arbitrary point Q to a straight line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ (formula (P) in the table 1):

$$d_{QQ_\perp} = \frac{\|\boldsymbol{m} + \boldsymbol{v} \times \boldsymbol{q}\|}{\|\boldsymbol{v}\|} = \frac{\|(\boldsymbol{p} - \boldsymbol{q}) \times \boldsymbol{v}\|}{\|\boldsymbol{v}\|}. \tag{7}$$

5. Plane equation

5.1. Parametric and general equations of the plane

Consider here *the parametric equation of the plane*. For information about other classical forms of the plane equation, we refer the reader to [36, Ch. 5, §3]. The parametric equation is given as follows:

$$\boldsymbol{r}(s,t) = \boldsymbol{p}_1 + (\boldsymbol{p}_2 - \boldsymbol{p}_1)s + (\boldsymbol{p}_3 - \boldsymbol{p}_1)t,$$

where $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ are the radius vectors of three points, s and t are some real numbers that are parameters. Unlike a line, a plane requires two parameters, since it is a two-dimensional object.

The differences $\boldsymbol{u} = \boldsymbol{p}_2 - \boldsymbol{p}_1$ and $\boldsymbol{v} = \boldsymbol{p}_3 - \boldsymbol{p}_1$ can be interpreted as tangent vectors to the plane, since

$$\boldsymbol{r}_s = \frac{\partial \boldsymbol{r}}{\partial s} = \boldsymbol{p}_2 - \boldsymbol{p}_1, \quad \boldsymbol{r}_t = \frac{\partial \boldsymbol{r}}{\partial t} = \boldsymbol{p}_3 - \boldsymbol{p}_1.$$

However, the tangent plane coincides with the π plane itself, which is why these tangent vectors lie completely in π and are the *guide vectors* of the plane.

Normal to a plane is a line perpendicular to any line lying completely on the plane and, therefore, perpendicular to the plane itself. The guiding vector of the normal is denoted as $\boldsymbol{N}$ and is called *the normal vector of the plane*. Its normalized version is more often used $\boldsymbol{n} = \boldsymbol{N}/\|\boldsymbol{N}\|$— the unit normal vector of the plane.

The plane π can be defined using the vector $\boldsymbol{N}$ and one fixed point belonging to the plane P_0 with the radius vector $\boldsymbol{p}_0 = (x_0, y_0, z_0)^T$. Let $\boldsymbol{p} = (x, y, z)^T$ be the radius vector of an arbitrary point in the plane, then $\boldsymbol{p} - \boldsymbol{p}_0$ also belongs to the plane, since the beginning and end of the vector lie in π. It follows from the perpendicularity of the normal vector to the plane that

$$(\boldsymbol{N}, \boldsymbol{p} - \boldsymbol{p}_0) = 0 \Rightarrow (\boldsymbol{N}, \boldsymbol{p}) - (\boldsymbol{N}, \boldsymbol{p}_0) = 0.$$

Table 1

Formulas in terms of homogeneous coordinates [30, Table 3.1]. The duality relation holds between formulas D and E, F and L, G and M, H and N. To simplify the search for the formula output, the first column contains the number under which it appears in the main text

		Formula	**Description**
A	(1)	$\{\boldsymbol{v} \mid \boldsymbol{p} \times \boldsymbol{v}\}$	A line passing through the point P in the direction of $\boldsymbol{v}$.
B	(2)	$\{\boldsymbol{p}_2 - \boldsymbol{p}_1 \mid \boldsymbol{p}_1 \times \boldsymbol{p}_2\}$	A line passing through two points P_1 and P_2.
C	(6)	$\{\boldsymbol{p} \mid \boldsymbol{0}\}$	A line passing through the origin and the point P.
D	(3)	$\{w_1\boldsymbol{p}_2 - w_2\boldsymbol{p}_1 \mid \boldsymbol{p}_1 \times \boldsymbol{p}_2\}$	A line passing through two points $\vec{\boldsymbol{p}}_1 = (\boldsymbol{p}_1 \mid w_1)$ and $\vec{\boldsymbol{p}}_2 = (\boldsymbol{p}_2 \mid w_2)$.
E	(12)	$\{\boldsymbol{n}_1 \times \boldsymbol{n}_2 \mid d_1\boldsymbol{n}_2 - d_2\boldsymbol{n}_1\}$	A line of intersection of two layers $[\boldsymbol{n}_1 \mid_1]$ and $[\boldsymbol{n}_2 \mid_2]$.
F	(13)	$(\boldsymbol{m} \times \boldsymbol{n} + d\boldsymbol{v} \mid -(\boldsymbol{n}, \boldsymbol{v}))$	The point of intersection of the plane $[\boldsymbol{n} \mid]$ and the line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$.
F.a	(21)	$[\boldsymbol{m}_1 \times \boldsymbol{m}_2 \mid (\boldsymbol{v}_2, \boldsymbol{m}_1)]$	The point of intersection of two lines $\{\boldsymbol{v}_1 \mid \boldsymbol{m}_1\}$ и $\{\boldsymbol{v}_2 \mid \boldsymbol{m}_2\}$
F.b	(11)	$\big(d_1\boldsymbol{n}_3 \times \boldsymbol{n}_2 + d_2\boldsymbol{n}_1 \times \boldsymbol{n}_3 + d_3\boldsymbol{n}_2 \times \boldsymbol{n}_1 \mid (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)\big)$	The point of intersection of three planes $[\boldsymbol{n}_1 \mid d_1]$, $[\boldsymbol{n}_2 \mid d_2]$ и $[\boldsymbol{n}_3 \mid d_3]$
G	(4)	$(\boldsymbol{v} \times \boldsymbol{m} \mid (\boldsymbol{v}, \boldsymbol{v}))$	The point closest to the origin on the line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$.
H	(14)	$(-d\boldsymbol{n} \mid (\boldsymbol{n}, \boldsymbol{n}))$	The point closest to the origin on the plane $[\boldsymbol{n} \mid d]$.
I	(16)	$[\boldsymbol{v} \times \boldsymbol{u} \mid -(\boldsymbol{u}, \boldsymbol{m})]$	A plane containing a line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ and a direction $\boldsymbol{u}$.
J	(17)	$[\boldsymbol{v} \times \boldsymbol{p} + \boldsymbol{m} \mid -(\boldsymbol{p}, \boldsymbol{m})]$	A plane containing a line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ and a point $(\boldsymbol{p} \mid 1)$.
K	(18)	$[\boldsymbol{m} \mid 0]$	A plane containing a line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ and the origin.
L	(15)	$[\boldsymbol{v} \times \boldsymbol{p} + w\boldsymbol{m} \mid -(\boldsymbol{p}, \boldsymbol{m})]$	A plane containing a line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ and a point $(\boldsymbol{p} \mid w)$.
L.a	(19)	$[\boldsymbol{v} \times \boldsymbol{u} \mid (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{p})]$	The plane containing the point $(\boldsymbol{p} \mid 1)$ and the directions $\boldsymbol{v}$ and $\boldsymbol{u}$.
M		$[\boldsymbol{m} \times \boldsymbol{v} \mid (\boldsymbol{m}, \boldsymbol{m})]$	The plane with a line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$, the furthest from O.
N		$[-w\boldsymbol{p} \mid (\boldsymbol{p}, \boldsymbol{p})]$	The plane with the point $(\boldsymbol{p} \mid w)$ furthest from O.
O	(20)	$\dfrac{\lvert(\boldsymbol{v}_1, \boldsymbol{m}_2) + (\boldsymbol{v}_2, \boldsymbol{m}_1)\rvert}{\lVert\boldsymbol{v}_1 \times \boldsymbol{v}_2\rVert}$	The distance between lines $\{\boldsymbol{v}_1 \mid \boldsymbol{m}_1\}$ and $\{\boldsymbol{v}_2 \mid \boldsymbol{m}_2\}$.
P	(7)	$\dfrac{\lvert\boldsymbol{v} \times \boldsymbol{p} + \boldsymbol{m}\rvert}{\lVert\boldsymbol{v}\rVert}$	The distance between the line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ and point $(\boldsymbol{p} \mid 1)$.
Q	(5)	$\lVert\boldsymbol{m}\rVert / \lVert\boldsymbol{v}\rVert$	The distance from the line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$ to the origin.
R	(9)	$\dfrac{\lvert(\boldsymbol{n}, \boldsymbol{p}) + d\rvert}{\lVert\boldsymbol{n}\rVert}$	Distance from the plane $[\boldsymbol{n} \mid d]$ to the point $(\boldsymbol{p} \mid 1)$.
S	(8)	$\lvert d\rvert / \lVert\boldsymbol{n}\rVert$	The distance from the plane $[\boldsymbol{n} \mid d]$ to the origin.

Table 2

Normalization of a point, line, and plane

	General form	**Normalized form**
Point	$(\boldsymbol{p} \mid w)$	$(\boldsymbol{p}/w \mid 1)$
Line	$(\boldsymbol{v} \mid \boldsymbol{m})$	$[\boldsymbol{v}/\|\boldsymbol{v}\| \mid \boldsymbol{m}/\|\boldsymbol{v}\|]$
Plane	$[\boldsymbol{n} \mid d]$	$[\boldsymbol{n}/\|\boldsymbol{n}\| \mid d/\|\boldsymbol{n}\|]$

Table 3

Duality of a point, a line, and a plane

Original	**Dual**
Point $(\boldsymbol{p} \mid w)$	Plane $[\boldsymbol{p} \mid w]$
Line $\{\boldsymbol{v} \mid \boldsymbol{m}\}$	Line $\{\boldsymbol{m} \mid \boldsymbol{v}\}$
Plane $[\boldsymbol{n} \mid d]$	Point $(\boldsymbol{n} \mid d)$

Since the point P_0 is fixed, then $(\boldsymbol{N}, \boldsymbol{p}_0) = \text{const}$.

Let's introduce the components of the normal vector $\boldsymbol{N} = (A, B, C)$ and denote

$$D = -(\boldsymbol{N}, \boldsymbol{p}_0),$$

then, in the Cartesian coordinate system, the plane will be uniquely defined by a linear equation of the following form:

$$Ax + By + Cz + D = 0.$$

The resulting equation is called *general equation of the plane*. It is assumed that A, B, C do not simultaneously vanish, that is, $A^2 + B^2 + C^2 \neq 0$.

5.2. The projective representation of the plane

In the $\mathbb{R}\mathrm{P}^3$ model of a three-dimensional projective space, planes are modeled using hyperplanes of the $\mathbb{R}^4$ space. When using homogeneous coordinates in the $\mathbb{R}^4$ space, a Cartesian coordinate system with $Oxyzw$ axes is introduced, and a three-dimensional projective space is modeled as a hyperplane passing through a point $(0, 0, 0, 1)$ that is, through a point on the Oz axis, parallel to the hyperplane $Oxyz$. The hyperplanes $Ax+By+Cz+Dw = 0$ passing through the coordinate center and intersecting the plane $w = 1$ cut off the three-dimensional planes defined by the equation $Ax+By+Cz+D = 0$. In fact, this is the general equation of the plane from classical analytical geometry, which we discussed above. Let's now consider its normalized version.

Instead of the vector $\boldsymbol{N}$, you can use the unit normal vector $\boldsymbol{n}$, the components of which are calculated as follows:

$$\boldsymbol{n} = \frac{\boldsymbol{N}}{\|\boldsymbol{N}\|} = \left(\frac{A}{\sqrt{A^2 + B^2 + C^2}}, \frac{B}{\sqrt{A^2 + B^2 + C^2}}, \frac{B}{\sqrt{A^2 + B^2 + C^2}}\right) = (n_x, n_y, n_z), \quad n_x^2 + n_y^2 + n_z^2 = 1.$$

The values n_x, n_y, n_z are uniquely determined by the guiding cosines $n_x = \cos\alpha$, $n_y = \cos\beta$, $n_z = \cos\gamma$, where α, β, γ are the angles between the vector $\boldsymbol{n}$ and the axes Ox, Oy and Oz. You can also normalize D by dividing it by the norm of the normal vector $\boldsymbol{N}$

$$d = \frac{D}{\sqrt{A^2 + B^2 + C^2}}.$$

As a result, the general equation of the plane will be written as *normal equation of the plane* in the following form:

$$n_x x + n_y y + n_z z + d = \cos\alpha x + \cos\beta y + \cos\gamma z + d = (\boldsymbol{n}, \boldsymbol{p}) + d = 0.$$

If instead of the radius vector $\boldsymbol{p} = (x, y, z)^T$ of a three-dimensional Euclidean space defining the coordinates of a point $P \in \pi$, enter the radius vector $\vec{\boldsymbol{p}} = (\boldsymbol{p} \mid 1) = (x : y : z : 1)$ of a projective space defining the projective coordinates of a point P, then the normal equation of the plane can be rewritten as a scalar product of the vector $\vec{\boldsymbol{p}}$ and the vector $\vec{\boldsymbol{\pi}} = [\boldsymbol{n} \mid d] = (n_x : n_y : n_z : d)$:

$$(\vec{\boldsymbol{p}}, \vec{\boldsymbol{\pi}}) = \begin{bmatrix} n_x & n_y & n_z & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = n_x x + n_y y + n_z z + d = 0.$$

Writing down the coefficients of the normal equation of the plane in the form of $[\boldsymbol{n} \mid d]$, we follow the notation adopted in [30].

The vector $\vec{\boldsymbol{\pi}} = [\boldsymbol{n} \mid d]$ for a plane is an analog of homogeneous coordinates for a point and allows you to write formulas related to the plane in a homogeneous form. Indeed, multiplying the vector $\vec{\boldsymbol{n}}$ by a scalar does not change the equation of the plane and you can always return to the normalized form by dividing all the components of the vector by $\|\boldsymbol{n}\|$:

$$\left[\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|} \;\middle|\; \frac{d}{\|\boldsymbol{n}\|} \right]. \tag{8}$$

The value of $|d|/\|\boldsymbol{n}\|$ is equal to the distance from the origin to the plane. We have introduced the vector $\boldsymbol{n}$ as a unit, however, an error may accumulate in the process of computer calculations and $\boldsymbol{n}$ will cease to be a unit. Therefore, the formula (8) is divided by $\|\boldsymbol{n}\|$, which provides renormalization, eliminating the accumulated error.

5.3. The distance from the point to the plane

Let's return to the general equation of the π plane and write it in a normalized form:

$$(\boldsymbol{n}, \boldsymbol{p}) + d = 0, \quad n_x x + n_y y + n_z z + d = 0, \quad \|\boldsymbol{n}\| = 1.$$

The geometric meaning of the value d is the projection of the radius vector of an arbitrary point P_1 lying on a plane onto the unit vector of the normal $\boldsymbol{n}$, as shown in the figure 10. The distance from an arbitrary point in space to the plane is defined as the length of the perpendicular lowered from this point onto the plane. The direction of the perpendicular to the plane coincides with the direction of the normal vector $\boldsymbol{n}$, so d can also be interpreted as the distance from the origin to the plane.

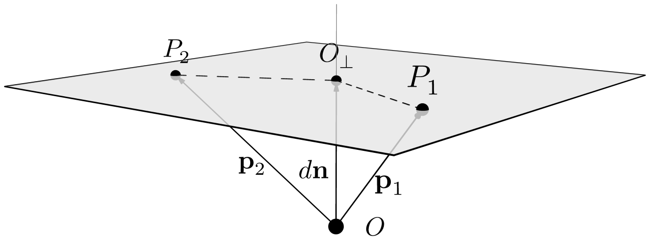


Figure 10. The geometric meaning of d is the directional distance from the origin to the plane. Radius vector $\boldsymbol{OO}_\perp = -d\boldsymbol{n}$, where $\boldsymbol{n}$ is the unit normal vector

We show that the value of d does not depend on the choice of a point on the plane. Any point in the plane is defined using the parametric equation radius vector

$$\boldsymbol{p}(t,s) = \boldsymbol{p}_1 + \boldsymbol{u}s + \boldsymbol{v}t,$$

where $\boldsymbol{u} = \boldsymbol{p}_2 - \boldsymbol{p}_1$ and $\boldsymbol{v} = \boldsymbol{p}_3 - \boldsymbol{p}_1$ are the guiding vectors of the plane perpendicular to the normal vector $\boldsymbol{n}$. Calculate the length of the projection of an arbitrary point $(\boldsymbol{p},\boldsymbol{n})$:

$$(\boldsymbol{p},\boldsymbol{n}) = (\boldsymbol{p}_1 + \boldsymbol{u}s + \boldsymbol{v}t, \boldsymbol{n}) = (\boldsymbol{p}_1,\boldsymbol{n}) + \cancelto{0}{(\boldsymbol{u},\boldsymbol{n})s} + \cancelto{0}{(\boldsymbol{v},\boldsymbol{n})t} = (\boldsymbol{p}_1,\boldsymbol{n}).$$

Since the choice of the point P_1 is also arbitrary, the parameter d is uniquely determined for this particular plane. So, the radius vector of the point $O_\perp$ — projections of the origin on the plane π can be calculated as $\boldsymbol{OO}_\perp = -d\boldsymbol{n}$ (see also (14)).

Let Q be an arbitrary point in space, and $\boldsymbol{q} = (x,y,z)^T$ be its radius vector. The value $\delta(\boldsymbol{q}) = n_x x + n_y y + n_z z + d$ is called *deviation of a point from the plane* and has the geometric meaning of the oriented distance from the point Q to the plane. The orientation of the distance makes it possible to determine by the sign on which side of the plane the point is located.

$$\delta(\boldsymbol{q}) = (\boldsymbol{n},\boldsymbol{q}) + d = \begin{cases} > 0, & \text{is the point is in front of the plane,} \\ = 0, & \text{is the point belongs to the plane,} \\ < 0, & \text{is the point is behind the plane.} \end{cases}$$

To clarify, the phrase “the point is in front of the plane” means that an observer at the point can see the front of the plane, and the phrase “the point is behind the plane” means that the same observer sees the inside of the plane. If the plane forms the face of some complex three-dimensional object, then in this case the observer is inside this object.

If we consider the homogeneous coordinates of the point Q given by the vector $\vec{\boldsymbol{q}} = (\boldsymbol{q} \mid 1) = (x : y : z : 1)$, then the deviation of the point Q from the plane reduces to finding the dot product of the vectors $\boldsymbol{q}$ and $\vec{\boldsymbol{\pi}}$:

$$\delta(\vec{\boldsymbol{q}}) = n_x x + n_y y + n_z z + d = \begin{bmatrix} n_x & n_y & n_z & d \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = (\vec{\boldsymbol{\pi}}, \vec{\boldsymbol{q}}). \tag{9}$$

Note that in the table 1, this formula is given in a normalized form on the assumption that the normal vector $\boldsymbol{n}$ will not necessarily be singular.

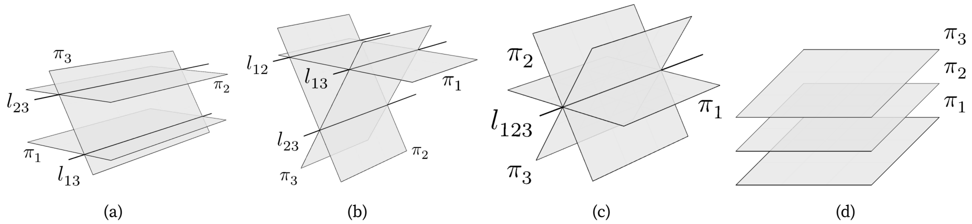


Figure 11. Options for intersecting planes at an incorrect point. In variant (a), the lines l_{13} and l_{23} are parallel and intersect at an irregular point, hence the planes intersect at the same point. Similarly, in case (b), three parallel lines intersect at an irregular point. In Figure (c), the planes intersect along their own straight line l_{123}, which contains, among other things, an irregular point at which the planes intersect. Finally, in (d), parallel planes intersect in an irregular straight line

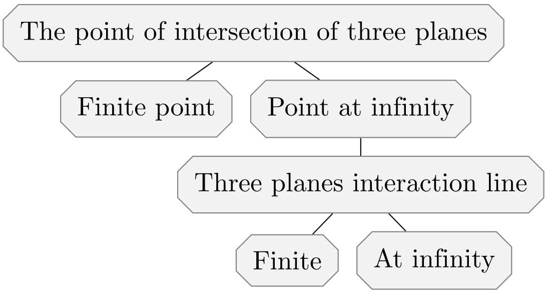

The point of intersection of three planes
Finite point
Point at infinity
Three planes interaction line
Finite
At infinity

Figure 12. Variants of the intersection of three planes in a projective space

The direction of the normal vector to the plane is extremely important in computer graphics, since modeling the reflection of light from three-dimensional objects depends on it. The normal vector allows you to specify the *front side of the plane*, that is, the side from which the normal vector is directed (the vector is directed into the eye of the observer looking at the plane) and the *back side of the plane*, opposite to the front. The front side can also be called face side if the plane forms the face of an object.

5.4. Intersection of two and three planes

In ordinary three-dimensional Euclidean space, three planes can be in six different positions.

1. Have a single common point, figure 13.
2. Have a single common straight line, Figure 11 (c).
3. Intersect in pairs, as shown in the figure, figure 11 (b).
4. Two planes can be parallel and intersect with the third, as shown in the figure, figure 11 (a).
5. All three planes can be mutually parallel, figure 11 (d).
6. All planes can match.

If we consider the location of three planes in a projective space, then the number of options is sharply reduced, since in a projective space any planes intersect, however, the intersection point can be either its own or improper. The general scheme is shown in the diagram 12.

Let's define three planes in a homogeneous form: $\vec{\boldsymbol{\pi}}_1 = [\boldsymbol{n}_1 \mid d_1]$, $\vec{\boldsymbol{\pi}}_2 = [\boldsymbol{n}_2 \mid d_2]$ and $\boldsymbol{\pi}_3 = [\boldsymbol{n}_3 \mid d_3]$. It is necessary to find a point $P = \pi_1 \cap \pi_2 \cap \pi_3$ whose homogeneous coordinates $(\boldsymbol{p} \mid 1) = (x : y : z : 1)$ satisfy each of the three equations of the planes π_1, π_2 and π_3:

$$(\vec{\boldsymbol{\pi}}_i, \vec{\boldsymbol{p}}) = [\boldsymbol{n}_i \mid 1] \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = 0 \;\Leftrightarrow\; (\boldsymbol{n}_i, \boldsymbol{p}) + d_i = 0, \quad i = 1, 2, 3.$$

Let's consider the matrix N, the rows of which make up the components of the normal vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ and use it to write a system of linear equations for the components of the radius vector $\boldsymbol{p}$:

$$N = \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \boldsymbol{n}_3 \end{bmatrix} = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ n_{3x} & n_{3y} & n_{3z} \end{bmatrix} \Longrightarrow \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ n_{3x} & n_{3y} & n_{3z} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = - \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad \boldsymbol{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

To solve this system, we will find the inverse matrix N^{-1}, for which we will use the formula to find the attached matrix $N^{\vee}$ using the cross product and the mixed product to find the determinant $\det N$, after which we write

$$N^{-1} = \frac{N^{\vee}}{\det N} = \frac{[\boldsymbol{n}_2 \times \boldsymbol{n}_3 \; \boldsymbol{n}_3 \times \boldsymbol{n}_1 \; \boldsymbol{n}_1 \times \boldsymbol{n}_2]}{(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)}, \quad N^{\vee} = \begin{bmatrix} | & | & | \\ \boldsymbol{n}_2 \times \boldsymbol{n}_3 & \boldsymbol{n}_3 \times \boldsymbol{n}_1 & \boldsymbol{n}_1 \times \boldsymbol{n}_2 \\ | & | & | \end{bmatrix},$$

$$(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3) = \det N = \begin{vmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ n_{3x} & n_{3y} & n_{3z} \end{vmatrix},$$

where the vectors $\boldsymbol{n}_2 \times \boldsymbol{n}_3$, $\boldsymbol{n}_3 \times \boldsymbol{n}_1$ and $\boldsymbol{n}_1 \times \boldsymbol{n}_2$ form the columns of the matrix $N^{\vee}$. Next, you can write the solution of a system of linear equations in the following form:

$$\boldsymbol{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_2)} \begin{bmatrix} | & | & | \\ \boldsymbol{n}_2 \times \boldsymbol{n}_3 & \boldsymbol{n}_3 \times \boldsymbol{n}_1 & \boldsymbol{n}_1 \times \boldsymbol{n}_2 \\ | & | & | \end{bmatrix} \begin{bmatrix} -d_1 \\ -d_2 \\ -d_3 \end{bmatrix} = \frac{-d_1 \boldsymbol{n}_2 \times \boldsymbol{n}_3 - d_2 \boldsymbol{n}_3 \times \boldsymbol{n}_1 - d_3 \boldsymbol{n}_1 \times \boldsymbol{n}_2}{(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)}.$$

Finally, we write down the formula for the radius vector of the point P of the intersection of three planes $\pi_1 : [\boldsymbol{n}_1 \mid d_1]$, $\pi_2 : [\boldsymbol{n}_2 \mid d_2]$ and $\pi_3 : [\boldsymbol{n}_3 \mid d_3]$:

$$\boldsymbol{p} = \frac{d_1 \boldsymbol{n}_3 \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{n}_3 + d_3 \boldsymbol{n}_2 \times \boldsymbol{n}_1}{(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)}. \tag{10}$$

In this form, the formula can be found, for example, in [22, p. 56] or in [30, p. 129]. In homogeneous coordinates, the same formula can be written without the division operation by moving the denominator to the place of the w coordinate:

$$\vec{\boldsymbol{p}} = \big(d_1 \boldsymbol{n}_3 \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{n}_3 + d_3 \boldsymbol{n}_2 \times \boldsymbol{n}_1 \mid (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3)\big).$$

To solve the system of equations $(n, * p) = 0$, you can use the more familiar Kramer method.

$$\begin{cases} n_{1x}x + n_{1y}y + n_{1z}z + d_1 = 0, \\ n_{2x}x + n_{2y}y + n_{2z}z + d_2 = 0, \\ n_{3x}x + n_{3y}y + n_{3z}z + d_3 = 0. \end{cases} \Leftrightarrow \begin{cases} n_{1x}x + n_{1y}y + n_{1z}z = -d_1, \\ n_{2x}x + n_{2y}y + n_{2z}z = -d_2, \\ n_{3x}x + n_{3y}y + n_{3z}z = -d_3. \end{cases}$$

$$\Delta = \begin{vmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ n_{3x} & n_{3y} & n_{3z} \end{vmatrix} \quad \Delta_x = \begin{vmatrix} -d_1 & n_{1y} & n_{1z} \\ -d_2 & n_{2y} & n_{2z} \\ -d_3 & n_{3y} & n_{3z} \end{vmatrix} \quad \Delta_y = \begin{vmatrix} n_{1x} & -d_1 & n_{1z} \\ n_{2x} & -d_2 & n_{2z} \\ n_{3x} & -d_3 & n_{3z} \end{vmatrix} \quad \Delta_z = \begin{vmatrix} n_{1x} & n_{1y} & -d_1 \\ n_{2x} & n_{2y} & -d_2 \\ n_{3x} & n_{3y} & -d_3 \end{vmatrix}.$$

As a result of solving this system, we obtain a point in homogeneous coordinates in the form:

$$\vec{\boldsymbol{p}} = \left(\frac{\Delta_x}{\Delta}, \frac{\Delta_y}{\Delta}, \frac{\Delta_z}{\Delta} \mid 1 \right) = (\Delta_x, \Delta_y, \Delta_z \mid \Delta).$$

We get that:

$$d_1 \boldsymbol{n}_3 \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{n}_3 + d_3 \boldsymbol{n}_2 \times \boldsymbol{n}_1 = \Delta_x \boldsymbol{e}_x + \Delta_y \boldsymbol{e}_y + \Delta_z \boldsymbol{e}_z.$$

Let's write down once again the formula for calculating the homogeneous coordinates of the intersection point of three planes $\pi_1 : [\boldsymbol{n}_1 \mid d_1]$, $\pi_2 : [\boldsymbol{n}_2 \mid d_2]$ and $\pi_3 : [\boldsymbol{n}_3 \mid d_3]$ in a homogeneous form in two versions:

$$\vec{\boldsymbol{p}} = \big(d_1 \boldsymbol{n}_3 \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{n}_3 + d_3 \boldsymbol{n}_2 \times \boldsymbol{n}_1 \mid (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3) \big) = (\Delta_x, \Delta_y, \Delta_z \mid \Delta). \tag{11}$$

The first option has an advantage, since it is written in a non-component form, and in the second option, the components participate in calculating the determinants Δ_x, Δ_y and Δ_z of the system.

For a full-fledged analytical study of the relative position of the three planes, we should first consider the problem of the location of the two planes, which we will do next. In the meantime, it follows directly from (11) that if $\Delta = (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3) \neq 0$, then the intersection point is its own (terminal), and if $(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3) = 0$, then the point is incorrect and this case covers all the remaining 5 plane locations.

Two planes in a projective space relative to each other can be in the following positions.

- Have their own common straight line (intersect along the end line).
- Should have a common irregular straight line (be parallel and intersect in a perfect straight line).
- Match.

When considering planes in a projective space, it is not necessary to consider the first two cases separately.

Find the straight line $l = \pi_1 \cap \pi_2$, which is obtained when two planes intersect $\pi_1 : [\boldsymbol{n}_1 \mid d_1]$, and $\pi_2 : [\boldsymbol{n}_2 \mid d_2]$. The guiding vector of such a straight line must be perpendicular to both the normal vector $\boldsymbol{n}_1$ and the normal vector $\boldsymbol{n}_2$, therefore it can be calculated as $\boldsymbol{v} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$, which means shown on the left side of the figure 13a.

To find the point of this straight line, take the third plane π, with a normal vector equal to the vector $\boldsymbol{n}_3 = \boldsymbol{v}$ and passing through the origin, that is, $d_3 = 0$. Then, using the formula (10), we immediately get:

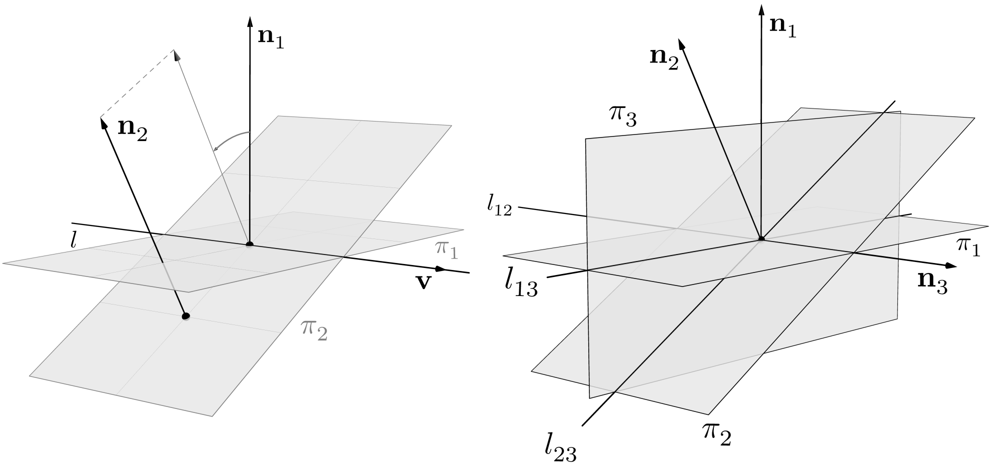
$$\boldsymbol{p} = \frac{d_1 \boldsymbol{v} \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{v}}{(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{v})},$$

where the triple product is simplified to the square of the norm of the cross product $\|\boldsymbol{n}_1 \times \boldsymbol{n}_2\|^2$

$$(\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{v}) = (\boldsymbol{n}_1, \boldsymbol{n}_2 \times \boldsymbol{v}) = (\boldsymbol{n}_1 \times \boldsymbol{n}_2, \boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{v}) = \|\boldsymbol{v}\|^2 = \|\boldsymbol{n}_1 \times \boldsymbol{n}_2\|^2.$$

Let's finally write it down:

$$\boldsymbol{p} = \frac{d_1 \boldsymbol{v} \times \boldsymbol{n}_2 + d_2 \boldsymbol{n}_1 \times \boldsymbol{v}}{\|\boldsymbol{v}\|^2}, \text{ where } \boldsymbol{v} = \boldsymbol{n}_1 \times \boldsymbol{n}_2.$$



(a)

(b)

Figure 13. In Figure (a), the line of intersection of two planes with the guide vector $\boldsymbol{v} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$. The triple of vectors $\langle \boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{v} \rangle$ is right, since the rotation from $\boldsymbol{n}_1$ to $\boldsymbol{n}_2$ from the end of $\boldsymbol{v}$ occurs counterclockwise. In Figure (b), three planes intersect along straight lines $l_{12} = \pi_1 \cos \pi_2$, $l_{13} = \pi_1 \cos \pi_3$, $l_{23} = \pi_2 \cos \pi_3$. The third plane passes through the origin and its normal vector is perpendicular to $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ and coincides with the guiding vector $\boldsymbol{v}$ of the line l_{12} of the intersection of π_1 and π_2.

To write the line l in Plucker coordinates, calculate the moment $\boldsymbol{m}$

$$\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{v} = \frac{d_1 \boldsymbol{v} \times \boldsymbol{n}_2 \times \boldsymbol{v} + d_2 \boldsymbol{n}_1 \times \boldsymbol{v} \times \boldsymbol{v}}{\|\boldsymbol{v}\|^2},$$

Using the “bac minus cab” property of the cross product, we transform the numerator to a simpler form:

$$\begin{aligned} &\boldsymbol{v} \times \boldsymbol{n}_2 \times \boldsymbol{v} = \boldsymbol{n}_2 \|\boldsymbol{v}\|^2 - \boldsymbol{v}(\boldsymbol{v}, \boldsymbol{n}) = \boldsymbol{n}_2 \|\boldsymbol{v}\|^2, \\ &\boldsymbol{n}_1 \times \boldsymbol{v} \times \boldsymbol{v} = \boldsymbol{v}(\boldsymbol{v}, \boldsymbol{n}) - \boldsymbol{n}_1(\boldsymbol{v}, \boldsymbol{v}) = -\boldsymbol{n}_1 \|\boldsymbol{v}\|^2, \end{aligned} \Rightarrow$$

$$\Rightarrow d_1 \boldsymbol{v} \times \boldsymbol{n}_2 \times \boldsymbol{v} + d_2 \boldsymbol{n}_1 \times \boldsymbol{v} \times \boldsymbol{v} = d_1 \boldsymbol{n}_2 \|\boldsymbol{v}\|^2 - d_2 \boldsymbol{n}_1 \|\boldsymbol{v}\|^2 = (d_1 \boldsymbol{n}_2 - d_2 \boldsymbol{n}_1) \|\boldsymbol{v}\|^2.$$

We used the fact that $\boldsymbol{v} \perp \boldsymbol{n}_1$ and $\boldsymbol{v} \perp \boldsymbol{n}_2$ since $\boldsymbol{v} = \boldsymbol{n}_1 \times \boldsymbol{n}_2$ which means $(\boldsymbol{v}, \boldsymbol{n}_1) = (\boldsymbol{v}, \boldsymbol{n}_2)$. The formula for the moment is simplified.

$$\boldsymbol{m} = \frac{(d_1 \boldsymbol{n}_2 - d_2 \boldsymbol{n}_1) \|\boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} = d_1 \boldsymbol{n}_2 - d_2 \boldsymbol{n}_1.$$

As a result, the line of intersection of the planes $\pi_1 : [\boldsymbol{n}_1 \ |_1]$ and $\pi_2 : [\boldsymbol{n}_2 \ |_2]$ in Plucker coordinates is written as

$$\{\boldsymbol{v} \mid \boldsymbol{m}\} = \{\boldsymbol{n}_1 \times \boldsymbol{n}_2 \mid d_1 \boldsymbol{n}_2 - d_2 \boldsymbol{n}_1\} \tag{12}$$

5.5. Intersection of a straight line and a plane

Consider the problem of the intersection of the plane $\pi : [\boldsymbol{n} \mid d]$ and the straight line l, given in parametric form by the radius vector $\boldsymbol{l}(t) = \boldsymbol{p} + \boldsymbol{v}t$. Let’s immediately exclude the case when the straight line is parallel to the plane, that is, $(\boldsymbol{n}, \boldsymbol{v}) = 0$, $\boldsymbol{n} \perp \boldsymbol{v}$ and there is no intersection point.

In the case of non-parallel lines and planes, they will necessarily intersect at some point Q with homogeneous coordinates $(\boldsymbol{q} \mid 1)$, where $\boldsymbol{q}$ is the radius vector of a point in Euclidean space. Since the point belongs to a straight line, then $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{v}t$ for a certain value of the parameter t. The task is to find the value of this parameter. To do this, we substitute the radius vector of the point Q into the normal equation of the plane, or, in other words, we find the dot product of homogeneous vectors:

$$[\boldsymbol{n} \mid d]\begin{bmatrix}\boldsymbol{q}\\1\end{bmatrix} = (\boldsymbol{n},\boldsymbol{q}) + d = 0 \Rightarrow (\boldsymbol{n},\boldsymbol{p}) + t(\boldsymbol{n},\boldsymbol{v}) + d = 0 \Rightarrow t = -\frac{(\boldsymbol{n},\boldsymbol{p}) + d}{(\boldsymbol{n},\boldsymbol{v})}.$$

Substituting this value of t into the parametric equation of the straight line, we write:

$$\boldsymbol{q} = \boldsymbol{p} - \frac{(\boldsymbol{n},\boldsymbol{p}) + d}{(\boldsymbol{n},\boldsymbol{v})}\boldsymbol{v}.$$

Let's transform this formula so as to express the moment $\boldsymbol{m}$ of a straight line:

$$(\boldsymbol{n},\boldsymbol{v})\boldsymbol{q} = \boldsymbol{p}(\boldsymbol{n},\boldsymbol{v}) - (\boldsymbol{n},\boldsymbol{p})\boldsymbol{v} - d\boldsymbol{v} = -(\boldsymbol{v}(\boldsymbol{n},\boldsymbol{p}) - \boldsymbol{p}(\boldsymbol{n},\boldsymbol{v}) + d\boldsymbol{v}) = -\boldsymbol{n}\times\boldsymbol{v}\times\boldsymbol{p} - d\boldsymbol{v}.$$

Here we used the “bac minus cab” property of the cross product again. Now consider that $\boldsymbol{m} = \boldsymbol{p}\times\boldsymbol{v}$ and write:

$$-\boldsymbol{n}\times\boldsymbol{v}\times\boldsymbol{p} - d\boldsymbol{v} = \boldsymbol{n}\times\boldsymbol{m} - d\boldsymbol{v} \Rightarrow (\boldsymbol{n},\boldsymbol{v})\boldsymbol{q} = \boldsymbol{n}\times\boldsymbol{m} - d\boldsymbol{v} = -(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v}) \Rightarrow \boldsymbol{q} = -\frac{(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v})}{(\boldsymbol{n},\boldsymbol{v})}.$$

The homogeneous coordinates of the point Q can be written as

$$\left(-\frac{(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v})}{(\boldsymbol{n},\boldsymbol{v})} \;\Big|\; 1\right) = (-(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v}) \mid (\boldsymbol{n},\boldsymbol{v})) = ((\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v}) \mid -(\boldsymbol{n},\boldsymbol{v})) \tag{13}$$

If the lines are parallel, then $(\boldsymbol{n},\boldsymbol{v}) = 0$ and the intersection point becomes ideal $(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v} \mid 0)$ i.e. the lines intersect in an infinitely distant point and the resulting coordinates indicate the direction where this point is located.

5.6. The point of the plane closest to the origin

If the plane is specified as $[\boldsymbol{n} \mid]$. The point of the plane closest to the origin lies on a straight line passing through the origin perpendicular to the plane. Such a straight line is defined by the guiding vector $\boldsymbol{v} = \boldsymbol{n}$ and a certain moment $\boldsymbol{m} = \boldsymbol{p}\times\boldsymbol{v}$, where $\boldsymbol{p}$ is the radius vector of an arbitrary point on a straight line. Since the straight line passes through the origin, you can choose $\boldsymbol{p} = \boldsymbol{0}$ and calculate the moment $\boldsymbol{m} = \boldsymbol{0}\times\boldsymbol{v} = \boldsymbol{0}$. Therefore, the line can be written as $\{\boldsymbol{v} \mid \boldsymbol{0}\}$. Using the formula (13) we write

$$(\boldsymbol{m}\times\boldsymbol{n} + d\boldsymbol{v} \mid -(\boldsymbol{n},\boldsymbol{v})) = (d\boldsymbol{n} \mid -(\boldsymbol{n},\boldsymbol{n})) = (d\boldsymbol{n} \mid -\|\boldsymbol{n}\|^2) = (-d\boldsymbol{n} \mid \|\boldsymbol{n}\|^2).$$

As a result, we obtain a formula for calculating the homogeneous coordinates of the point in the plane closest to the origin.

$$(-d\boldsymbol{n} \mid \|\boldsymbol{n}\|^2). \tag{14}$$

5.7. A plane passing through a straight line and a point

Let the plane π contain a straight line l defined by the Plucker coordinates in the form $\{\boldsymbol{v} \mid \boldsymbol{m}\}$, as well as a point P with homogeneous coordinates $(\boldsymbol{p} \mid w)$. The radius vector of this point in $\mathbb{R}^3$ will have the form $\boldsymbol{p}/w$. As the second point of the plane, we can take any point lying on a straight line, the equation of which we know. This point is the point of the straight line closest to the origin, which is calculated using the formula (4) $(\boldsymbol{v} \times \boldsymbol{m} \mid (\boldsymbol{v}, \boldsymbol{v}))$. Let's denote this point as Q and calculate the vector $\boldsymbol{PQ} = \boldsymbol{u}$:

$$\boldsymbol{u} = \frac{\boldsymbol{p}}{w} - \boldsymbol{q} = \frac{\boldsymbol{p}}{w} - \frac{\boldsymbol{v} \times \boldsymbol{m}}{\|\boldsymbol{v}\|^2}.$$

Now we can find the normal vector of the plane π as $\boldsymbol{n} = \boldsymbol{v} \times \boldsymbol{u}$, since $\boldsymbol{v} \perp \pi$ and $\boldsymbol{u} \perp \pi$.

$$\boldsymbol{n} = \boldsymbol{v} \times \boldsymbol{u} = \frac{\boldsymbol{v} \times \boldsymbol{p}}{w} - \frac{\boldsymbol{v} \times \boldsymbol{v} \times \boldsymbol{m}}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v} \times \boldsymbol{p}}{w} + \frac{\boldsymbol{m}\|\boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v} \times \boldsymbol{p}}{w} + \boldsymbol{m},$$

where we used the "bac minus cab" property of the cross product. As a result, we come to the expression for $\boldsymbol{n}$:

$$w\boldsymbol{n} = \boldsymbol{v} \times \boldsymbol{p} + w\boldsymbol{m}.$$

The distance from the origin to the plane is found as the length of the projection of the point $\boldsymbol{p}/w$ onto the vector $\boldsymbol{n}$:

$$\left\|\frac{1}{w}\boldsymbol{p}_{\|\boldsymbol{n}}\right\| = \frac{1}{w}(\boldsymbol{p}, \boldsymbol{n}) = \frac{1}{w}\left(\boldsymbol{p}, \boldsymbol{m} + \frac{\boldsymbol{v} \times \boldsymbol{p}}{w}\right) = \frac{1}{w}(\boldsymbol{p}, \boldsymbol{m}) + \frac{1}{w}\cancelto{0}{\left(\boldsymbol{p}, \frac{\boldsymbol{v} \times \boldsymbol{p}}{w}\right)} = \frac{1}{w}(\boldsymbol{p}, \boldsymbol{m})$$

and the directional distance from the plane to the straight line can be calculated as $d = -\dfrac{(\boldsymbol{p}, \boldsymbol{m})}{w}$.

Now we can write the plane in homogeneous coordinates as

$$[\boldsymbol{n} \mid d] = \left[\frac{\boldsymbol{v} \times \boldsymbol{p}}{w} + \boldsymbol{m} \,\middle|\, -\frac{(\boldsymbol{p}, \boldsymbol{m})}{w}\right],$$

multiplying by w, we finally write down the formula L from the table 1

$$[\boldsymbol{v} \times \boldsymbol{p} + w\boldsymbol{m} \mid -(\boldsymbol{p}, \boldsymbol{m})]. \tag{15}$$

If instead of a point on the plane, the direction is known, that is, a point in homogeneous coordinates $(\boldsymbol{u} \mid 0)$, then replacing the point $(\boldsymbol{p} \mid w)$ with $(\boldsymbol{u} \mid 0)$, we'll write it down immediately

$$[\boldsymbol{v} \times \boldsymbol{u} \mid -(\boldsymbol{u}, \boldsymbol{m})]. \tag{16}$$

If the coordinates of a point are given as $(\boldsymbol{p} \mid 1)$, that is, $w = 1$, then we also directly obtain a homogeneous representation of the plane from the formula (15)

$$[\boldsymbol{v} \times \boldsymbol{p} + \boldsymbol{m} \mid -(\boldsymbol{p}, \boldsymbol{m})]. \tag{17}$$

If the plane passes through the origin, then also substituting $(\boldsymbol{0} \mid 1)$ using the formula (15), we get

$$[\boldsymbol{m} \mid 0]. \tag{18}$$

Let's prove another formula that is not in the original table, but which can be useful when defining the plane π through vector guides. Let us know one point P of the plane $\vec{\boldsymbol{p}} = (\boldsymbol{p} \mid 1)$ and two guide vectors $\vec{\boldsymbol{v}} = (\boldsymbol{v} \mid 0)$ and $\vec{\boldsymbol{u}} = (\boldsymbol{u} \mid 0)$. Then, using the formula (1), we obtain the Plucker coordinates of a straight line lying in the plane $\{\boldsymbol{v} \mid \boldsymbol{p} \times \boldsymbol{v}\} \subset \pi$, and from (16) the final expression for the plane π in homogeneous form:

$$[\boldsymbol{v} \times \boldsymbol{u} \mid -(\boldsymbol{u}, \boldsymbol{p} \times \boldsymbol{v})] = [\boldsymbol{v} \times \boldsymbol{u} \mid (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{p})], \tag{19}$$

where $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{p})$ is the triple product of three vectors.

5.8. The position of straight lines in space

In a projective space, two straight lines l_1 and l_2 can be in three positions relative to each other.

- Lines can *intersect*, that is, they do not lie in the same plane.
- Lines can lie in the same plane (*coplanar* lines). In this case:
  - lines intersect at their own point;
  - lines intersect at an irregular point (parallel).

The distance between the lines is calculated as the length of the mutual perpendicular. If the lines intersect, the length of the perpendicular will be zero.

When using Plucker coordinates, the distance between two intersecting lines can be obtained as the distance between two parallel planes. Consider the lines $\{\boldsymbol{v}_1 \mid \boldsymbol{m}_1\}$ and $\{\boldsymbol{v}_2 \mid \boldsymbol{m}_2\}$ and construct two planes π_1 and π_2, where the plane π_1 contains the straight line $\{\boldsymbol{v}_1 \mid \boldsymbol{m}_1\}$ and the direction given by the vector $\boldsymbol{v}_2$. In turn, the plane π_2 contains the line $\{\boldsymbol{v}_2 \mid \boldsymbol{m}_2\}$ and the vector $\boldsymbol{v}_1$. The normal vectors of the planes match and are calculated as $\boldsymbol{v}_1 \times \boldsymbol{v}_2 = \boldsymbol{n}_1 = \boldsymbol{n}_2$.

We use the formula (16), where for π_1 we put $\boldsymbol{v} = \boldsymbol{v}_1$, $\boldsymbol{u} = \boldsymbol{v}_2$, and for π_2, on the contrary: $\boldsymbol{v} = \boldsymbol{v}_2$, $\boldsymbol{u} = \boldsymbol{v}_1$, as a result:

$$\pi_1 : [\boldsymbol{v}_1 \times \boldsymbol{v}_2 \mid -(\boldsymbol{v}_2, \boldsymbol{m}_1)], \quad \pi_2 : [\boldsymbol{v}_2 \times \boldsymbol{v}_1 \mid -(\boldsymbol{v}_1, \boldsymbol{m}_2)].$$

To find the distance from the origin to the planes, let's go to the normalized view by dividing by the norm of the normal vector $\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|$:

$$\pi_1 : \left[\frac{\boldsymbol{v}_1 \times \boldsymbol{v}_2}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|} \;\Big|\; -\frac{(\boldsymbol{v}_2, \boldsymbol{m}_1)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}\right], \quad \pi_2 : \left[-\frac{\boldsymbol{v}_1 \times \boldsymbol{v}_2}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|} \;\Big|\; -\frac{(\boldsymbol{v}_1, \boldsymbol{m}_2)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}\right] = \left[\frac{\boldsymbol{v}_1 \times \boldsymbol{v}_2}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|} \;\Big|\; \frac{(\boldsymbol{v}_1, \boldsymbol{m}_2)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}\right].$$

The distances from the planes π_1 and π_2 to the origin will be respectively:

$$d_1 = -\frac{(\boldsymbol{v}_2, \boldsymbol{m}_1)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}, \quad d_2 = -\frac{(\boldsymbol{v}_1, \boldsymbol{m}_2)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}.$$

The distance between the planes is calculated as the difference between d_2 and d_1:

$$d = |d_2 - d_1| = \left|\frac{(\boldsymbol{v}_1, \boldsymbol{m}_2) + (\boldsymbol{v}_2, \boldsymbol{m}_1)}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}\right|. \tag{20}$$

By the condition of constructing planes, this is the distance between the lines l_1 and l_2. Value $M = (\boldsymbol{v}_1, \boldsymbol{m}_2) + (\boldsymbol{v}_2, \boldsymbol{m}_1)$ is called *mutual moment* of two straight lines l_1 and l_2.

- If $M > 0$, then turn from l_1 to l_2 — right.
- If $M < 0$, then turn from l_1 to l_2 — left.
- If $M = 0$, then the lines lie in the same plane or, in other words, *are coplanar*.

The condition of line coplanarity $M = 0$ can be given a simple geometric interpretation. We have found the formula for the distance between straight lines as the distance between two parallel planes. If it turns to zero, then the planes coincide, and the mutual moment is zero.:

$$M = (\boldsymbol{v}_1, \boldsymbol{m}_2) + (\boldsymbol{v}_2, \boldsymbol{m}_1) = 0.$$

Meeting this condition allows us to assert that the lines lie in the same plane, but does not allow us to determine the point of their intersection (proper or improper).

To find the intersection point of the lines, we use the projective formula (11) and consider three planes: the first plane π_1 passes through l_1 and the origin, the second plane π_2— through l_2 and the origin, and finally the third plane π_3 contains the straight line l_1 and the direction $\boldsymbol{v}_2$. From the formula (16) we get the Plucker coordinates for π_3, and from (18) for π_1 and π_2:

$$\vec{\pi}_1 = [\boldsymbol{m}_1 \mid 0], \quad \vec{\pi}_2 = [\boldsymbol{m}_2 \mid 0], \quad \vec{\pi}_3 = [\boldsymbol{v}_1 \times \boldsymbol{v}_2 \mid -(\boldsymbol{v}_2, \boldsymbol{m}_1)].$$

Now we substitute the components of the planes in (11), which will give the following expression:

$$[-(\boldsymbol{v}_2, \boldsymbol{m}_1)\boldsymbol{m}_2 \times \boldsymbol{m}_1 \mid (\boldsymbol{v}_1 \times \boldsymbol{v}_2, \boldsymbol{m}_1 \times \boldsymbol{m}_2)].$$

Simplify $(\boldsymbol{v}_1 \times \boldsymbol{v}_2, \boldsymbol{m}_1 \times \boldsymbol{m}_2)$ using the Lagrange property of the vector product:

$$(\boldsymbol{v}_1 \times \boldsymbol{v}_2, \boldsymbol{m}_1 \times \boldsymbol{m}_2) = (\boldsymbol{v}_1, \boldsymbol{m}_1)(\boldsymbol{v}_2, \boldsymbol{m}_2) - (\boldsymbol{v}_1, \boldsymbol{m}_2)(\boldsymbol{v}_2, \boldsymbol{m}_1) = -(\boldsymbol{v}_1, \boldsymbol{m}_2)(\boldsymbol{v}_2, \boldsymbol{m}_1),$$

since the moment and the guiding vector of the straight line are orthogonal, then $(\boldsymbol{v}_1, \boldsymbol{m}_1) = (\boldsymbol{v}_2, \boldsymbol{m}_2) = 0$. Next, we use the equality of the mutual moment of the curves to zero: $(\boldsymbol{v}_1, \boldsymbol{m}_2) + (\boldsymbol{v}_2, \boldsymbol{m}_1) = 0$, whence $(\boldsymbol{v}_1, \boldsymbol{m}_2) = -(\boldsymbol{v}_2, \boldsymbol{m}_1)$. Finally we get:

$$(\boldsymbol{v}_1 \times \boldsymbol{v}_2, \boldsymbol{m}_1 \times \boldsymbol{m}_2) = (\boldsymbol{v}_2, \boldsymbol{m}_1)^2 = (\boldsymbol{v}_1, \boldsymbol{m}_2)^2.$$

Using the uniformity of coordinates, we divide all coordinates by $(\boldsymbol{v}_2, \boldsymbol{m}_1)$:

$$[-(\boldsymbol{v}_2, \boldsymbol{m}_1)\boldsymbol{m}_2 \times \boldsymbol{m}_1 \mid (\boldsymbol{v}_2, \boldsymbol{m}_1)^2] = [-\boldsymbol{m}_2 \times \boldsymbol{m}_1 \mid (\boldsymbol{v}_2, \boldsymbol{m}_1)] = [\boldsymbol{m}_1 \times \boldsymbol{m}_2 \mid (\boldsymbol{v}_2, \boldsymbol{m}_1)].$$

We have obtained that the intersection point of two straight lines $\{\boldsymbol{v}_1 \mid \boldsymbol{m}_1\}$ and $\{\boldsymbol{v}_2 \mid \boldsymbol{m}_2\}$ in homogeneous coordinates is calculated as

$$[\boldsymbol{m}_1 \times \boldsymbol{m}_2 \mid (\boldsymbol{v}_2, \boldsymbol{m}_1)] = [\boldsymbol{m}_2 \times \boldsymbol{m}_1 \mid (\boldsymbol{v}_1, \boldsymbol{m}_2)] \tag{21}$$

6. Conclusion

We have outlined the basics of the $\mathbb{R}\mathrm{P}^3$ model of projective space in an analytical form. As noted above, there are two aspects of this work that may be of value.

- Filling the gap in educational literature and specialized monographs. Analytical projective geometry is actively used for applied purposes in robotics and computer graphics, and a detailed description of its basic aspects is extremely useful in methodological and pedagogical plans.
- Summary of the basic formulas in the form of a table, which is an extended table from [30]. Based on this table, you can implement everything you need to work with projective points, straight lines, and planes in the form of software structures.

The latter aspect is covered in detail in the article [39].

Author Contributions: Methodology, writing—original draft preparation, Migran N. Gevorkyan; writing—review and editing, Anna V. Korolkova; conceptualization, Dmitry S. Kulyabov; supervision, Leonid A. Sevastianov. All authors have read and agreed to the published version of the manuscript.

Funding: This publication has been supported by the RUDN University Scientific Projects Grant System, project No 021934-0-000 (recipients Anna V. Korolkova; Migran N. Gevorkyan, Leonid A. Sevastianov) and has been supported by the RUDN University Strategic Academic Leadership Program (recipient Dmitry S. Kulyabov).

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sulanke, R. & Onishchik, A. L. *Projective and Cayley-Klein Geometries* 450 pp. (Springer, 2006).
2. Rosov, N. K. Felix Klein and his Erlangen program. Russian. *Mat. Pros.* **3,** 49–55 (3 2019).

3. Volberg, O. A. *Basic ideas of projective geometry* 3rd ed. Russian. 188 pp. (State Educational and Pedagogical Publishing House of thy Ministry of Education of the RSFSR, Moscow, 1949).
4. Coxeter, H. S. M. *The Real Projective Plane* (1955).
5. Ponarin, Y. P. *Affine and projective geometry* Russian. 288 pp. (MCCME, Moscow, 2019).
6. Prasolov, V. V. & Tikhomirov, V. M. *Geometry* Russian. 336 pp. (MCCME, Moscow, 2019).
7. Young, J. W. *Projective Geometry* 184 pp. (1938).
8. Shcherbakov, R. N. & Pichurin, L. F. *From projective geometry to non-Euclidean geometry (around the absolute)* Russian. 158 pp. (Prosveshchenie, Moscow, 1979).
9. Baer, R. *Linear Algebra and Projective Geometry* 399 pp. (New York, 1952).
10. Artin, E. *Geometric Algebra* 214 pp. (Interscience Publishers, New York, 1957).
11. Busemann, H. & Kelly, P. J. *Projective geometry and projective metrics* 352 pp. (Courier Corporation, New York, 2012).
12. Roger, D. F. & Adams, J. A. *Mathematical elements for computer graphics* 611 pp. (1990).
13. Ammeraal, L. *Programming Principles in Computer Graphics* 233 pp. (John Wiley and Sons, 1992).
14. Shikin, E. V. & Boreskov, A. V. *Computer graphics* Polygonal models. 464 pp. (Dialog-MIFI, Moscow, 2001).
15. Marschner, S. & Shirley, P. *Fundamentals of Computer Graphics* 5th ed. 717 pp. (CRC Press, 2022).
16. Vince, J. *Rotation transforms for computer graphics* 1st ed. 232 pp. (Springer-Verlag London, 2011).
17. Golovanov, N. N. *Geometric modeling* Russian. 472 pp. (Fizmatlit, Moscow, 2002).
18. Dunn, F. & Parberry, I. *3D Math Primer for Graphics and Game Development* 2nd ed. 846 pp. (CRC Press, 2011).
19. Nikulin, E. A. *Computer geometry and machine graphics algorithms* Russian. 560 pp. (BHV-Petersburg, Saint-Petersburg, 2003).
20. Cisarzh, V. V. & Marusik, R. I. *Mathematical methods of computer graphics* Russian. 464 pp. (Fakt, Kiev, 2004).
21. Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. *Robotics. Modelling, Planning and Control* 632 pp. (Springer-Verlag, London, 2010).
22. Faux, I. D. & Pratt, M. J. *Computational Geometry for Design and Manufacture* 329 pp. (1980).
23. De Berg, M., Cheong, O., van Kreveld, M. & Overmars, M. *Computational Geometry* Algorithms and Applications (Springer Berlin Heidelberg, 2008).
24. Boreskov, A. V. *Computer graphics programming. Modern OpenGL* Russian. 372 pp. (DMK Press, Moscow, 2019).
25. Manevich, V. A., Kotov, I. I. & Zengin, A. *Analytical geometry with image theory* Russian. 304 pp. (Vysshaya shkola, Moscow, 1969).
26. Dimentberg, F. M. *The Screw Calculus and Its Applications in Mechanics* 162 pp. (Foreign Technology Division translation, 1965).
27. Kotelnikov, A. P. *The helical number and some of its applications to geometry and mechanics* Russian. 222 pp. (ComKniga, Moscow, 2019).
28. Chelnokov, Y. Biquaternion solution of the kinematic control problem for the motion of a rigid body and its application to the solution of inverse problems of robot-manipulator kinematics. *Mechanics of Solids* **48,** 512. doi:10.3103/S0025654413010044 (Apr. 2013).
29. Hartley, R. & Zisserman, A. *Multiple View Geometry in Computer Vision* 2nd ed. 655 pp. (Cambridge University Press, Cambridge, 2004).
30. Lengyel, E. *Foundations of Game Engine Development.* 1: *Mathematics* 4 vols. 195 pp. (Terathon Software LLC, Lincoln, California, 2016).
31. Hammerlindl, A., Bowman, J. & Prince, T. *Asymptote. The Vector Graphics Language* https://asymptote.sourceforge.io/ (2023).

32. Kryachkov, Y. G. *Asymptote for beginners. Creating drawings in the language of vector graphics Asymptote* Russian. http://mif.vspu.ru/books/ASYfb.pdf (2020).
33. Volchenko, Y. M. *Scientific graphics in the Asymptote language* Russian. http://www.math.volchenko.com/AsyMan.pdf (2020).
34. Tedeev, V. A. *From painting to projective geometry* Russian. 231 pp. (Visha Shkola, Moscow, 1988).
35. Hartshorne, R. *Foundations of projective geometry* 161 pp. (W. A. Benjamin inc., New York, 1967).
36. Ilyin, V. A. & Poznyak, E. G. *Analytical geometry* Russian. 232 pp. (Nauka, Moscow, 1981).
37. Fedorchuk, V. V. *Analytical Geometry and Linear Algebra course* Russian. 328 pp. (MSU, Moscow, 1990).
38. Savelov, A. A. *Flat curves. Taxonomy, properties, and applications* Russian. 293 pp. (State Publishing House of Physico-mathematical Literature, Moscow, 1960).
39. Gevorkyan, M. N., Korolkova, A. V., Kulyabov, D. S. & Sevastyanov, L. A. Implementation of Analytic Projective Geometry for Computer Graphics. *Programming and Computer Software* **50,** 153–165. doi:10.1134/S0361768824020075 (Apr. 2024).

Information about the authors

Gevorkyan, Migran N. (Russian Federation)—Candidate of Sciences in Physics and Mathematics, Associate Professor of Department of Probability Theory and Cyber Security of Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University) (e-mail: gevorkyan-mn@rudn.ru, phone: +7 (495) 955-09-27, ORCID: 0000-0002-4834-4895, ResearcherID: E-9214-2016, Scopus Author ID: 57190004380)

Anna V. Korolkova—Docent, Candidate of Sciences in Physics and Mathematics, Associate Professor of Department of Probability Theory and Cyber Security of RUDN University (e-mail: korolkova-av@rudn.ru, phone: +7 (495) 952-02-50, ORCID: 0000-0001-7141-7610, ResearcherID: I-3191-2013, Scopus Author ID: 36968057600)

Kulyabov, Dmitry S.—Professor, Doctor of Sciences in Physics and Mathematics, Professor. of the Department of Probability Theory and Cyber Security of RUDN University; Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: kulyabov-ds@rudn.ru, phone: +7 (495) 952-02-50, ORCID: 0000-0002-0877-7063, ResearcherID: I-3183-2013, Scopus Author ID: 35194130800)

Leonid A. Sevastianov—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Computational Mathematics and Artificial Intelligence of RUDN University (e-mail: sevastianov-la@rudn.ru, phone: +7 (495) 955-07-83, ORCID: 0000-0002-1856-4643, ResearcherID: B-8497-2016, Scopus Author ID: 8783969400)

УДК 004.021:519.2:519.6

DOI: 10.22363/2658-4670-2025-33-1-74-102 EDN: AAAAHV

Аналитическая проективная геометрия для компьютерной графики

М. Н. Геворкян[1], А. В. Королькова[1], Д. С. Кулябов[1, 2], Л. А. Севастьянов[1, 2]

[1] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[2] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. Мотивом к написанию данной работы послужила разработка авторами курса по компьютерной геометрии для студентов физико-математических специальностей. Под термином «компьютерная геометрия» здесь и далее понимаются математические основы машинной графики. Важно отдельно подчеркнуть, что разрабатываемый курс должен быть рассчитан на студентов второго года обучения и, следовательно, от них можно требовать лишь предварительное знание стандартного курса алгебры и математического анализа. Это накладывает определённые ограничения на излагаемый материал. При изучении тематической литературы было выяснено, что стандартом де факто в современной компьютерной графике стало использование проективного пространства и однородных координат. Однако авторы столкнулись с проблемой методологического характера — практически полным отсутствием подходящей учебной литераторы как на русском, так и на английском языках. Для представления собранной авторами информации по данному вопросу и была написана данная работа.

Ключевые слова: проективная геометрия, система Asymptote, координаты Плюккера, собственные и несобственные точки, прямые и плоскости

Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 103–111
http://journals.rudn.ru/miph

Research article

UDC 519.7
PACS 07.05.Tp
DOI: 10.22363/2658-4670-2025-33-1-103-111

EDN: AFJUOE

On the methods of minimizing the risks of implementing artificial intelligence in the financial business of a company

Eugeny Yu. Shchetinin[1], Leonid A. Sevastianov[2, 3],
Anastasia V. Demidova[2], Tatyana R. Velieva[2]

[1] Financial University under the Government of the Russian Federation, 49 Leningradsky Prospect, Moscow, 125993, Russian Federation

[2] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[3] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: August 30, 2024; revised: September 20, 2024; accepted: September 25, 2024)

Abstract. Effective application of artificial intelligence (AI) models in various fields in the field of financial risks can increase the speed of data processing, deepen the degree of their analysis and reduce labor costs, thereby effectively improving the efficiency of financial risk control. The application of AI in the field of financial risk management puts forward new requirements for the system configuration and operation mode of financial supervision. With the rapid growth of computer and network technologies, the increase in the frequency of market transactions, the diversification of data sources, and the development and application of big data, this creates new problems for financial risk management based on big data. This paper analyzes the role of artificial intelligence in promoting the reform and growth of the financial industry, and proposes countermeasures for the rational use of AI in the field of financial risk management.

Key words and phrases: artificial intelligence, financial risks, big data

For citation: Shchetinin, E. Y., Sevastianov, L. A., Demidova, A. V., Velieva, T. R. On the methods of minimizing the risks of implementing artificial intelligence in the financial business of a company. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 103–111. doi: 10.22363/2658-4670-2025-33-1-103-111. edn: AFJUOE (2025).

1. Introduction

The annual economic growth rate in most developed countries could double in the near future due to the widespread implementation of artificial intelligence. However, the spread of innovative technologies also brings new challenges. Insurance companies need new risk management strategies to maximize the benefits of the implementation of artificial intelligence in society and business.

By 2035, AI-based technologies are expected to increase corporate productivity by an average of 38% across 16 industries in 12 countries, according to a 2018 Allianz survey of 1,911 risk experts worldwide. The proliferation of AI-based technologies, from chatbots to autonomous vehicles, is inexorably transforming industries and society [1].

Artificial intelligence is already being used to increase productivity through unique insights gained through data analysis and through automation of simple tasks. Expectations for AI-based technologies are growing, and private corporations have begun to invest more and more to be the first to take advantage of its benefits. Experts estimate the impact of AI and other innovative technologies on the economy to be higher than, say, the impact of political risks and climate change. At the same time, many of them note possible negative effects from the introduction of innovations.

For example, the penetration of artificial intelligence into manufacturing could make automated, autonomous, or self-learning machines more vulnerable to cyber threats, as well as the potential for large-scale disruptions and losses, especially when it comes to critical infrastructure. Artificial intelligence, for example, could reduce traffic accidents by up to 90%, but it also brings with it uncertainty about liability and ethics in the event of an accident. Uber Technologies recently halted testing of self-driving cars after one of them struck and killed a woman, marking the first fatal accident involving a pedestrian and self-driving cars. Immediately after the incident, Uber announced it was suspending all of its autonomous vehicle testing in Pittsburgh, San Francisco, Toronto, and much of Phoenix [2].

Healthcare is another sector of the economy where expectations for artificial intelligence are very high. There is a hypothesis that the use of advanced data analysis will help to overcome many diseases that are currently incurable, diagnose diseases that require detection and cross-checking through a large number of medical tests. At the same time, the problem of protecting patients' personal data is obvious, for example, with the widespread use of medical records by artificial intelligence to study new diseases. This problem has already drawn attention to the need to change the legislative regulation of data protection and patient rights [3].

The threat landscape in the digital security sphere is also likely to change. New technologies will reduce cyber risks by better detecting attacks, but will also increase their likelihood if, for example, hackers gain control. Artificial intelligence will pave the way for them to carry out more serious incidents, reducing the cost of organizing cyber attacks and allowing them to be carried out more targeted. Social issues will become acute. According to a study by the consulting company McKinsey, today more than 1.1 billion full-time jobs in the world are related to functions that can be automated, of which more than 100 million are in the United States and Europe [4].

In the financial sector, AI has also played a significant role, providing a number of benefits, but also causing certain risks. The prospects and risks of using artificial intelligence in the financial sector and the expected changes it will bring to this industry are the subject of research in this paper.

2. An overview of potential risks of implementing AI in financial business

Possibility of systematic errors. It is important to note that AI operates on the basis of algorithms, and its operation may become unstable if the data used is distorted or incorrect, which may lead to serious financial losses.

Threat of replacing human resources. The introduction of AI may lead to job losses in the financial sector, which will cause social and economic problems.

Insufficient regulation. It is necessary to develop appropriate legislative norms and rules that would regulate the use of AI in the financial sector to minimize risks and protect the interests of clients. The following are the most important risks in the implementation of AI:

1. Bias, unreliability of results due to inappropriate or unrepresentative data
2. Inability to interpret/explain the results of an AI model
3. Inappropriate use of data
4. Vulnerabilities to cyber-attacks to obtain and/or manipulate data

5. Social consequences of rapid transformation because of the transition to AI technologies.

The practical consequences of materializing these risks can take many forms: damage to reputation, reduction in organizational values, fines, legal costs. Organizations are afraid of risks, hence, more than half (56%) of respondents admitted that the implementation of AI technologies in their case is slow. However, this argument cannot remain dominant for long, because otherwise organizations risk losing their competitive opportunities. Rather than constantly putting on the brakes, it is better to use more thoughtful approaches to implementation, involving effective risk management [5].

Despite certain risks, the application of artificial intelligence in the financial sector is inevitable and is one of the key areas of development in this industry. It provides unique opportunities for growth and improvement of the quality of financial institutions, as well as for improving customer service. In any case, it is important to balance the prospects and risks to ensure safe and effective implementation of AI in the financial sector.

3. Basic methods for managing the risks of implementing AI in a company’s financial business

To successfully and safely implement AI in the financial sector, companies must take proactive measures to minimize the risks associated with it. The following strategies may be proposed for this purpose:

Improving the company’s cybersecurity. When using AI in the financial sector, there is a risk of leaking sensitive information, which can have serious consequences for customers and financial institutions. Financial institutions should invest in modern cybersecurity systems, including data encryption, multi-factor authentication, and regular security audits. It is also important to train employees in the basics of cybersecurity so that they can recognize potential threats.

Testing and validation of algorithms. Before deploying AI systems, algorithms must be thoroughly tested and validated. This includes using historical data to assess the accuracy of predictions and identify potential errors. Companies must also regularly update their models based on new data.

Transparency and explainability of AI algorithms and models. Developing “explainable” AI models will help increase trust with customers and regulators. Transparency in how decisions are made can reduce the risk of customer dissatisfaction and ensure regulatory compliance.

Ethical standards. With the rapid development of artificial intelligence (AI), businesses have incorporated generative AI into their operations and are enjoying numerous benefits such as workflow efficiency, cost reduction through task automation, reduction in human errors, rapid development of products and services, and improved data-driven decision making. Although numerous benefits can be achieved through the implementation of generative AI, the problem is that integrating AI into the business strategies of different parts of an organization comes with serious ethical challenges. These contradictions can disrupt competitive advantages and hinder customer engagement, and lead to decreased loyalty, trust, and brand building. Ethical dilemmas in AI integration include data bias that tends to distort results and is discriminatory, misuse of customer data obtained from AI, complex liability determination, and reliability and security risks during rapid deployment [6]. Financial firms should develop ethical guidelines for the use of AI, including mechanisms to prevent discrimination and ensure fairness. This could include the creation of ethics committees that monitor the use of AI and its impact on customers.

Combined approach to customer service. Although AI can automate many processes, it is important to maintain the human element in customer service. Combining AI with traditional customer service can provide a better customer experience and increase customer satisfaction.

Regulatory Compliance. Companies need to stay abreast of changes in legislation and regulation around AI and fintech. This includes compliance with data protection requirements, as well as reporting and transparency standards.

Personnel training and development. Investing in employee training will help them better understand AI technologies and their application in business. It will also help create a culture of innovation within the company and increase trust in new technologies.

Monitoring and auditing of AI systems. Regular monitoring and auditing of AI systems will help identify potential problems and deviations in the operation of algorithms. This includes analyzing their performance, checking for bias, and assessing the impact on business processes. Creating feedback mechanisms will allow for prompt responses to emerging problems [7–9].

Data management. Effective data management is a key aspect of successful AI implementation. Companies must ensure that data is of high quality, up-to-date, and secure. This includes developing strategies for collecting, storing, and processing data while respecting privacy regulations [10, 11].

Collaboration with external experts. Engaging with external consultants and AI experts can help companies avoid common pitfalls and adopt best practices. This may also include participating in joint projects with universities or research organizations.

Adaptation to market changes. The financial sector is constantly changing, and companies must be prepared to adapt to new conditions. This may include being flexible in the use of AI technologies, as well as being prepared for changes in legislation and consumer preferences.

Creating a culture of innovation. Creating a culture that encourages innovation will help employees feel comfortable adopting new technologies. This can be achieved by encouraging experimentation, openly discussing ideas, and implementing training and development initiatives.

Strategic planning. Companies should develop long-term strategies for the implementation of AI that take into account both current needs and future trends. This will help avoid impulsive decisions and ensure a consistent approach to technology integration.

Employee training and development. One of the key aspects of successful AI implementation is employee training. Companies should invest in upskilling programs to ensure employees are able to work effectively with new technologies. This includes both technical skills and an understanding of the ethical aspects of using AI.

Integrating AI into Business Processes. To achieve maximum effectiveness, AI should be integrated into existing business processes. This may include automating routine tasks, improving data analytics, and using predictive models to make more informed decisions [11, 12].

Change Management. The introduction of AI may cause resistance from employees, especially if they fear job losses. It is important to develop a change management strategy that includes open communication, support from management, and employee involvement in the transformation process.

Cybersecurity of company information systems. As the use of AI increases, so does the risk of cyber-attacks. Financial institutions must invest in cybersecurity to protect their data and systems from potential threats. This includes regular security audits and software updates [13].

Partnerships with technology companies. Collaboration with technology companies can accelerate the implementation of AI. Partnerships allow for the use of ready-made solutions and technologies, as well as the exchange of experience and knowledge.

Data analysis and decision making. One of the main advantages of AI is its ability to process and analyze huge amounts of data. Financial institutions can use AI to identify patterns and trends, which allows for more informed decisions. This may include assessing customer creditworthiness, analyzing investment risks, and optimizing portfolios [14–16].

Process automation. Automating routine tasks with AI frees up employees' time for strategic tasks. This may include automating loan application processing, account management, and other administrative processes, which improves the overall efficiency of the organization.

Risk Management. AI can significantly improve risk management in the financial sector. AI-based systems can predict potential financial crises, identify fraudulent transactions, and assess risks in real time, allowing for a quick response to threats.

Key AI Tools in Risk Management. Machine Learning and Deep Learning: Used to identify risk factors and create models that can predict the likelihood of adverse events.

Time Series Analysis: Used to predict market movements and identify trends.

NLP (Natural Language Processing): Used to analyze news, financial reports, and other text data that may impact market risks.

Examples of AI in Risk Management. Credit Risk. Example: Bank A uses machine learning models to assess credit risk. Algorithms analyze credit history, solvency, and other factors to predict the probability of loan default.

Tools: logistic regression, decision trees.

Market risks. Example: Trading company B uses deep learning algorithms to analyze market data and predict short-term price movements in stocks and currencies.

Tools: Recurrent neural networks, long short-term memory (LSTM).

Liquidity risk. Example: Financial institution C uses AI to predict liquidity risk by analyzing asset and liability turnover and market conditions.

Tools: Bayesian networks, time series analysis.

Operational risk. Example: Company D uses AI-based systems to monitor and manage operational risks, including fraud, cyber-attacks, and system failures.

Tools: Anomaly detection systems, data clustering.

The use of AI in risk management allows financial institutions to more effectively identify, assess, and mitigate various types of risks. These technologies provide deeper data analysis, which leads to more informed and accurate decisions. However, limitations and challenges related to the accuracy of AI data and models, as well as the ethical aspects of their use, must be taken into account.

Innovative products and services. With the introduction of AI, financial institutions can develop new innovative products and services that meet customer needs. This may involve the creation of new investment instruments, personal finance management applications, or lending platforms.

Customer Feedback. Using AI to analyze customer feedback helps to better understand their needs and expectations. Companies can use this data to improve their services and products, as well as to develop new offers based on customer preferences.

Investment in Research and Development. For the successful implementation of AI, it is important to invest in research and development. This allows you to stay at the forefront of technology and create competitive solutions that meet modern market requirements.

Monitoring and Adapting Strategies. Finally, it is important to constantly monitor the results of AI implementation and adapt strategies depending on the data received. Regularly analyzing the effectiveness of technologies and their impact on business will allow companies to remain competitive and develop successfully.

4. Research Results

The financial services sector worldwide is one of the leaders in the use and development of AI. However, AI poses numerous technical, ethical, and legal challenges that may undermine the data, cybersecurity, systemic risk, and ethics objectives of financial regulation, particularly regarding black box issues. As the research in this paper shows, traditional externally focused financial supervision is unlikely to be able to adequately address the risks posed by AI due to: (1) increased information asymmetry; (2) data dependence; and (3) interdependence. Accordingly, even if supervisors have exceptional resources and expertise, overseeing the use of artificial intelligence in the financial sector using traditional methods is extremely challenging. To address this shortcoming, it is necessary to strengthen the internal governance of financial institutions and introduce requirements for personal human accountability. This approach builds on the existing framework for executive accountability that was developed in the wake of the 2008 global financial crisis and the continuing stream of ethically questionable conduct around the world in finance. This framework should consider and be consistent with broader approaches to data privacy and human-in-the-loop outside finance.

From a financial oversight perspective, internal governance could be strengthened through an increased focus on the personal accountability of senior management (or key functional holders) for regulated areas and activities as defined for regulatory purposes. These rules for key functional holders — especially if complemented by specific requirements for AI due diligence and explainability — would help ensure that key personnel at financial firms are ensuring that any AI is operating in a manner consistent with the personal responsibilities of senior managers.

This direct personal responsibility encourages due diligence in studying new technologies, their use and impact, and demands fairness and explainability as part of any AI system, with correspondingly severe personal consequences for failure. For a financial services professional with direct responsibility, demonstrating due diligence and explainability will be key to personal protection in the event of regulatory claims.

5. Conclusion

Implementing AI in the financial sector is a complex but necessary process to ensure competitiveness in the modern world. Successful integration of technologies requires a comprehensive approach, including staff training, change management, compliance with ethical standards, and cybersecurity. Companies that can adapt to new conditions and effectively use the capabilities of AI will have a significant advantage in the market. It is important to remember that technology is only a tool; success depends on how it is applied to create value for customers and the business as a whole.

Implementing AI in the financial sector opens up new horizons for increasing efficiency, improving customer experience, and optimizing business processes. However, companies must be prepared for the challenges and risks associated with this technology. Applying risk mitigation strategies such as strengthening cybersecurity, ensuring algorithm transparency, and compliance with ethical standards will allow financial institutions to successfully integrate AI into their operations, while maintaining customer trust and regulatory compliance. Therefore, the approach to AI implementation must be comprehensive and balanced, which will ensure long-term sustainability and competitiveness in the market. Financial institutions that can effectively use AI will have a significant advantage in the market, providing their clients with better services and adapting to the constantly changing conditions. It is important to remember that technology is only a tool; success depends on how it is applied to create value for both the business and the clients.

Author Contributions: Conceptualization, Shchetinin E., Sevastianov, L.; methodology, Shchetinin E.; writing—review and editing Shchetinin E., Velieva T.; supervision, Demidova A., Sevastianov, L., Velieva T.; project administration, Sevastianov, L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hong, J. The Impact of Artificial Intelligence, Machine Learning, and Big Data on Finance Analysis/Jingqi Hong. *Advances in Economics Management and Political Sciences* **27,** 39–43. doi:10.54254/2754-1169/27/20231208 (2023).
2. Agarwal, A., Singhal, C. & Thomas, R. *AI-powered decision making for the bank of the future* 2021.
3. Guan, J. Artificial Intelligence in Healthcare and Medicine: Promises, Ethical Challenges and Governance. *Chinese Medical Sciences Journal* **34,** 76–83. doi:10.24920/003611 (2019).
4. Boukherouaa, E. B., Shabsigh, M. G., AlAjmi, K., Deodoro, J., Farias, A., Iskender, E. S. & Ravikumar, R. *Powering the digital economy: Opportunities and risks of artificial intelligence in finance* 34 pp. (International Monetary Fund, 2021).
5. Chan, L., Hogaboam, L. & Cao, R. *Applied artificial intelligence in business: Concepts and cases* 368 pp. doi:10.1007/978-3-031-05740-3 (Springer Cham, 2022).
6. Santosh, K. C. & Wall, C. *AI, Ethical Issues and Explainability–Applied Biometrics* doi:10.1007/978-981-19-3935-8 (Springer Singapore, 2022).
7. Charles, V., Rana, N. P. & Carter, L. Artificial Intelligence for data-driven decision-making and governance in public affairs. *Government Information Quarterly* **39,** 101742. doi:10.1016/j.giq.2022.101742 (2022).
8. Duft, G. & Durana, P. Artificial Intelligence-based Decision-Making Algorithms, Automated Production Systems, and Big Data-driven Innovation in Sustainable Industry 4.0. *Economics, Management, and Financial Markets* **15,** 9–18. doi:10.22381/EMFM15420201 (2020).
9. Lee, J. Access to finance for artificial intelligence regulation in the financial services industry. *European Business Organization Law Review* **21,** 731–757. doi:10.1007/s40804-020-00200-0 (2020).
10. Mogaji, E. & Nguyen, N. P. Managers' understanding of artificial intelligence in relation to marketing financial services: Insights from a cross-country study. *International Journal of Bank Marketing* **40,** 1272–1298. doi:10.1108/IJBM-09-2021-0440 (2021).
11. Truby, J., Brown, R. & Dahdal, A. Banking on AI: Mandating a proactive approach to AI regulation in the financial sector. *Law and Financial Markets Review* **14,** 110–120. doi:10.1080/17521440.2020.1760454 (2020).
12. Xie, M. Development of artificial intelligence and effects on financial system. *Journal of Physics: Conference Series* **1187,** 032084. doi:10.1088/1742-6596/1187/3/032084 (2019).
13. Camacho, J., Couce-Vieira A. andArroyo, D. & D., R. *A Cybersecurity Risk Analysis Framework for Systems with Artificial Intelligence Components* 2024.
14. Lee, J. Access to finance for artificial intelligence regulation in the financial services industry. *European Business Organization Law Review* **24,** 731–757. doi:10.1007/s40804-020-00200-0 (2020).
15. Rajagopal, N. K., Qureshi, N. I., Durga, S., Ramirez Asis, E. H., Huerta Soto, R. M., Gupta, S. K. & Deepak, S. Future of business culture: An artificial intelligence-driven digital framework for organization decisionmaking process. *Complexity,* 1–14. doi:10.1155/2022/7796507 (2022).
16. Daiya, H. AI-Driven Risk Management Strategies in Financial Technology. *Journal of Artificial Intelligence General science* **5,** 194–216. doi:10.60087/jaigs.v5i1.194 (2024).

Information about the authors

Eugeny Yu. Shchetinin—Doctor of Physical and Mathematical Sciences, Lecturer of Department of Mathematics (e-mail: riviera-molto@mail.ru, ORCID: 0000-0003-3651-7629, ResearcherID: O-8287-2017, Scopus Author ID: 16408533100)

Leonid A. Sevastianov—Professor, Doctor of Sciences in Physics and Mathematics, Professor at the Department of Computational Mathematics and Artificial Intelligence of RUDN University, Leading Researcher of Bogoliubov Laboratory of Theoretical Physics, Joint Institute for Nuclear Research (e-mail: sevastianov-la@rudn.ru, phone: +7 (495) 952-25-72, ORCID: 0000-0002-1856-4643)

Anastasia V. Demidova—Candidate of Physical and Mathematical Sciences, Associate Professor of Department of Probability Theory and Cyber Security of RUDN University (e-mail: demidova-av@rudn.ru, ORCID: 0000-0003-1000-9650)

Tatyana R. Velieva—Candidate of Physical and Mathematical Sciences, Assistant Professor of Department of Probability Theory and Cyber Security of RUDN University (e-mail: velieva-tr@rudn.ru, ORCID: 0000-0003-4466-8531)

УДК 519.7
PACS 07.05.Tp
DOI: 10.22363/2658-4670-2025-33-1-103-111 EDN: AFJUOE

О методах минимизации рисков внедрения искусственного интеллекта в финансовый бизнес компании

Е. Ю. Щетинин[1], Л. А. Севастьянов[2, 3], А. В. Демидова[2], Т. Р. Велиева[2]

[1] Финансовый университет при Правительстве Российской Федерации, Ленинградский проспект, д. 49, Москва, 125993, Российская Федерация

[2] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[3] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. Эффективное применение моделей искусственного интеллекта (ИИ) в различных областях в сфере финансовых рисков позволяет повысить скорость обработки данных, углубить степень их анализа и снизить трудозатраты, тем самым эффективно повышая эффективность контроля финансовых рисков. Применение ИИ в сфере управления финансовыми рисками выдвигает новые требования к конфигурации системы и режиму работы финансового надзора. В условиях быстрого роста компьютерных и сетевых технологий, увеличения частоты рыночных операций, диверсификации источников данных, а также развития и применения больших данных это создает новые проблемы для управления финансовыми рисками на основе больших данных. В данной статье анализируется роль искусственного интеллекта в содействии реформированию и росту финансовой отрасли, а также предлагаются контрмеры по рациональному использованию ИИ в сфере управления финансовыми рисками.

Ключевые слова: искусственный интеллект, финансовые риски, большие данные