

ТРУДЫ

**ИНСТИТУТА СИСТЕМНОГО
ПРОГРАММИРОВАНИЯ РАН**

**PROCEEDINGS OF THE INSTITUTE
FOR SYSTEM PROGRAMMING OF THE RAS**

ISSN Print 2079-8156
Том 35 Выпуск 2

ISSN Online 2220-6426
Volume 35 Issue 2

Институт системного
программирования
им. В.П. Иванникова РАН

Москва, 2023

ИСП **РАН**

Труды Института системного программирования РАН Proceedings of the Institute for System Programming of the RAS

Труды ИСП РАН – это издание с двойной анонимной системой рецензирования, публикующее научные статьи, относящиеся ко всем областям системного программирования, технологий программирования и вычислительной техники. Целью издания является формирование научно-информационной среды в этих областях путем публикации высококачественных статей в открытом доступе.

Издание предназначено для исследователей, студентов и аспирантов, а также практиков. Оно охватывает широкий спектр тем, включая, в частности, следующие:

- операционные системы;
- компиляторные технологии;
- базы данных и информационные системы;
- параллельные и распределенные системы;
- автоматизированная разработка программ;
- верификация, валидация и тестирование;
- статический и динамический анализ;
- защита и обеспечение безопасности ПО;
- компьютерные алгоритмы;
- искусственный интеллект.

Журнал издается по одному тому в год, шесть выпусков в каждом томе.

Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.

Труды ИСП РАН реферируются и/или индексируются в:

Proceedings of ISP RAS are a double-blind peer-reviewed journal publishing scientific articles in the areas of system programming, software engineering, and computer science. The journal's goal is to develop a respected network of knowledge in the mentioned above areas by publishing high quality articles on open access.

The journal is intended for researchers, students, and practitioners. It covers a wide variety of topics including (but not limited to):

- Operating Systems.
- Compiler Technology.
- Databases and Information Systems.
- Parallel and Distributed Systems.
- Software Engineering.
- Software Modeling and Design Tools.
- Verification, Validation, and Testing.
- Static and Dynamic Analysis.
- Software Safety and Security.
- Computer Algorithms.
- Artificial Intelligence.

The journal is published one volume per year, six issues in each volume.

Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:



Редколлегия

Главный редактор - [Аветисян Арутюн Ишханович](#), академик РАН, доктор физико-математических наук, профессор, ИСП РАН (Москва, Российская Федерация)

Заместитель главного редактора - [Кузнецов Сергей Дмитриевич](#), д.т.н., профессор, ИСП РАН (Москва, Российская Федерация)

Члены редколлегии

[Воронков Андрей Анатольевич](#), доктор физико-математических наук, профессор, Университет Манчестера (Манчестер, Великобритания)

[Вирбицкайте Ирина Бонавентуровна](#), профессор, доктор физико-математических наук, Институт систем информатики им. академика А.П. Ершова СО РАН (Новосибирск, Россия)

[Коннов Игорь Владимирович](#), кандидат физико-математических наук, Технический университет Вены (Вена, Австрия)

[Ластовецкий Алексей Леонидович](#), доктор физико-математических наук, профессор, Университет Дублина (Дублин, Ирландия)

[Ломазова Ирина Александровна](#), доктор физико-математических наук, профессор, Национальный исследовательский университет «Высшая школа экономики» (Москва, Российская Федерация)

[Новиков Борис Асенович](#), доктор физико-математических наук, профессор, Санкт-Петербургский государственный университет (Санкт-Петербург, Россия)

[Петренко Александр Федорович](#), доктор наук, Исследовательский институт Монреаля (Монреаль, Канада)

[Черных Андрей](#), доктор физико-математических наук, профессор, Научно-исследовательский центр CICESE (Энсенада, Баха Калифорния, Мексика)

[Шустер Ассаф](#), доктор физико-математических наук, профессор, Технион — Израильский технологический институт Technion (Хайфа, Израиль)

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.

Телефон: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Сайт: <http://www.ispras.ru/proceedings/>

Editorial Board

Editor-in-Chief - [Arutyun I. Avetisyan](#), Academician of RAS, Dr. Sci. (Phys.–Math.), Professor, Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Deputy Editor-in-Chief - [Sergey D. Kuznetsov](#), Dr. Sci. (Eng.), Professor, Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Editorial Members

[Igor Konnov](#), PhD (Phys.–Math.), Vienna University of Technology (Vienna, Austria)

[Alexev Lastovetsky](#), Dr. Sci. (Phys.–Math.), Professor, UCD School of Computer Science and Informatics (Dublin, Ireland)

[Irina A. Lomazova](#), Dr. Sci. (Phys.–Math.), Professor, National Research University Higher School of Economics (Moscow, Russian Federation)

[Boris A. Novikov](#), Dr. Sci. (Phys.–Math.), Professor, St. Petersburg University (St. Petersburg, Russian Federation)

[Alexandre F. Petrenko](#), PhD, Computer Research Institute of Montreal (Montreal, Canada)

[Assaf Schuster](#), Ph.D., Professor, Technion - Israel Institute of Technology (Haifa, Israel)

[Andrei Tchervnykh](#), Dr. Sci., Professor, CICESE Research Centre (Ensenada, Baja California, Mexico).

[Irina B. Virbitskaite](#), Dr. Sci. (Phys.–Math.), The A.P. Ershov Institute of Informatics Systems, Siberian Branch of the RAS (Novosibirsk, Russian Federation)

[Andrew Voronkov](#), Dr. Sci. (Phys.–Math.), Professor, University of Manchester (Manchester, United Kingdom)

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Tel: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Web: <http://www.ispras.ru/en/proceedings>

С о д е р ж а н и е

Реальное применение методов интерпретации свёрточных нейронных сетей: обнаружение и объяснение ошибок классификатора изображений документов. <i>Голодков, А.О., Беляева О.В., Перминов А.И.</i>	7
Генерация изображений рукописного текста на русском языке. <i>Богатенкова А.О., Беляева О.В., Перминов А.И.</i>	19
Исследование состязательных атак на нейронные сети распознавания образов. <i>Котляров Д.В., Дюдюн Г.Д., Ржевская Н.В., Лапина М.А., Бабенко М.Г.</i>	35
Разработка системы классификации нормативно-правовых документов на основе метода опорных векторов. <i>Насу Ю., Ланин В.В.</i>	49
Ферма данных: информационная система сбора, хранения и обработки неструктурированных данных из разнородных источников. <i>Левашкин С.П., Иванов К.Н., Кушуков С.В.</i>	57
Типизированные неизвестные значения: шаг к решению проблемы представления отсутствующей информации в реляционных базах данных. <i>Кузнецов С.Д.</i>	73
Программный комплекс SIO для работы со структурированными данными. <i>Игнатъев А.О., Мокшин С.Ю.</i>	101
Пути организации параллельного доступа к структурированным данным. <i>Игнатъев А.О., Мокшин С.Ю., Иванков Д.В., Бекетов Е.А.</i>	111
Разработка и реализация средства диагностики сетей Ethernet. <i>Охрименко Г.П., Мокшин С.Ю., Игнатъев А.О.</i>	127
Разработка подхода к сквозному тестированию приложения для управления конфигурацией виртуальной инфраструктуры предприятия. <i>Гераскин Е.В., Воинов Н.В.</i>	143
Параллельные вычисления в задачах восстановления искаженных изображений в пространственно-спектральной форме. <i>Костров Б.В., Григченко Н.Н., Вьюгина А.А., Баранова С.Н.</i>	157
Объектно-ориентированный подход к поддержке сценариев в системах оптического моделирования. <i>Копылов М.С., Дерябин Н.Б., Денисов Е.Ю.</i>	169
Об использовании открытых сторонних библиотек при программной реализации вихревых методов вычислительной гидродинамики. <i>Марчевский И.К., Измайлова Ю.А., Ерофеева М.А., Кобзарь Д.Ю.</i>	181

О моделировании плоских течений вязкой жидкости вихревыми методами в программном комплексе VM2D. <i>Коробова И.А., Рятина Е.П., Хорошева А.А.</i>	201
Особенности численного моделирования работы воздушно-тепловой завесы в OpenFOAM. <i>Григорьев С.В., Королева М.Р., Миценкова О.В.</i>	215
Параметрические исследования течений в микросоплах. <i>Митрюкова Е.А.</i>	225

Table of Contents

Real Application of CNN Interpretation Methods: Document Image Classification Model Errors' Idetection and Validation. <i>Golodkov A.O., Belyaeva O.V., Perminov A.I.</i>	7
Generation of images with handwritten text in Russian. <i>Bogatenkova A.O., Belyaeva O.V., Perminov A.I.</i>	19
Investigation of Adversarial Attacks on Pattern Recognition Neural Networks. <i>Kotlyarov D.V., Dyudyun G.D., Rzhetskaya N.V., Lapina M.A., Babenko M.G.</i>	35
Development of Legal Document Classification System Based on Support Vector Machine. <i>Nasu Iu., Lanin V.V.</i>	49
Data farm: Information system for collecting, storing and processing unstructured data from heterogeneous sources. <i>Levashkin S.P., Ivanov K.I., Kushukov S.V.</i>	57
Typed unknown values: a step towards solving the problem of representing missing information in relational databases. <i>Kuznetsov S.D.</i>	73
SIO program suite for structured data processing. <i>Ignatyev A.O., Mokshin S.Yu.</i>	101
Ways to organize parallel access to structured data. <i>Ignatyev A.O., Mokshin S.Yu., Ivankov D.V., Beketov E.A.</i>	111
Design and implementation of a diagnostic tool for Ethernet networks. <i>Okhrimenko G.P., Mokshin S.Yu., Ignatyev A.O.</i>	127
Approach to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure. <i>Geraskin E.V., Voinov N.V.</i>	143
Parallel computations in problems of reconstruction of distorted images in spatial-spectral form. <i>Kostrov B.V., Grinchenko N.N., Vyugina A.A., Baranova S.N.</i>	157
An object-oriented approach to scenario support in optics CAD systems. <i>Kopylov M.S., Deryabin N.B., Denisov E.Yu.</i>	169
On open third-party libraries usage in implementation of vortex particle methods of computational fluid dynamics. <i>Marchevsky I.K., Izmailova Yu.A., Erofeeva M.A., Kobzar D.Yu.</i>	181

On Problems in 2D Viscous Flows Simulation at Higher Values of the Reynolds Numbers by Vortex Methods Using the VM2D Code. <i>Korobova I.A., Ryatina E.P., Khorosheva A.A.</i>	201
Peculiarities of numerical simulation of air/heat curtain operation in OpenFOAM. <i>Grigorev S.V., Koroleva M.R., Mishchenkova O.V.</i>	215
Parametric studies of flows in microsoples. <i>Mitryukova E.A.</i>	225

DOI: 10.15514/ISPRAS-2023-35(2)-1



Real Application of CNN Interpretation Methods: Document Image Classification Model Errors' Detection and Validation

A.O. Golodkov, ORCID: 0000-0002-0417-2622 <golodkov@ispras.ru>

O.V. Belyaeva, ORCID: 0000-0002-6008-9671 <belyaeva @ispras.ru>

A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

*Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

Abstract. In this paper, we consider the case of applying convolutional neural networks interpretation methods to ResNet 18 model in order to identify and justify model errors. The model is used in the problem of classifying the orientation of text documents images. First, using interpretation methods, an assumption was made as to why the neural network shows low metrics on data that differs from training images. The alleged reason was the presence of artifacts on the generated training images, caused by the use of an image rotation function. Further, using the Vanilla Gradient, Guided Backpropagation, Integrated Gradients, GradCAM methods and the invented metric, we managed to accurately confirm the hypothesis put forward. The obtained results helped to significantly improve the accuracy of the model.

Keywords: CNN Interpretation; Document Image Classification; Document Orientation Detection.

For citation: Golodkov A.O., Belyaeva O.V., Perminov A.I. Real Application of CNN Interpretation Methods: Document Image Classification Model Errors' Idetection and Validation. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 7-18. DOI: 10.15514/ISPRAS-2023-35(2)-1

Реальное применение методов интерпретации свёрточных нейронных сетей: обнаружение и объяснение ошибок классификатора изображений документов

A.O. Голодков, ORCID: 0000-0002-0417-2622 <golodkov@ispras.ru>

О.В. Беляева, ORCID: 0000-0002-6008-9671 <belyaeva @ispras.ru>

А.И. Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

*Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25*

Аннотация. В данной статье рассматривается случай применения методов интерпретации свёрточных нейронных сетей к модели ResNet 18 с целью обнаружения и объяснения её ошибок. Сама модель используется для решения задачи определения ориентации изображений текстовых документов. Изначально с помощью методов интерпретации было выдвинуто предположение о причине низкого качества предсказаний модели на данных, отличных от примеров из обучающего набора. Предполагаемой причиной оказалось наличие артефактов на тренировочных данных, которые были сгенерированы с использованием функции поворота изображений. Далее, с помощью методов Vanilla Gradient, Guided Backpropagation, Integrated Gradients, GradCAM и предложенной метрики удалось точно обосновать выдвинутое предположение. Полученные результаты помогли значительно улучшить точность модели.

Ключевые слова: интерпретация свёрточных нейронных сетей; классификация изображений документов; определение ориентации документа

Для цитирования: Голодков, А.О., Беляева О.В., Перминов А.И. Реальное применение методов интерпретации свёрточных нейронных сетей: обнаружение и объяснение ошибок классификатора изображений документов. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 7-18. DOI: 10.15514/ISPRAS–2023–35(2)–1

1. Introduction

Convolutional neural networks are known for high metrics demonstrated in classification [1], segmentation [2], and object detection [3] tasks. This is due to the large number of improvements proposed to the original architecture [4] and the growing computing power. At the same time, the speed of neural networks due to the use of GPU also makes them an excellent replacement for the classical methods. ResNet 18, as an example of a convolutional neural network, shows high metrics in the classification task, while having a small number of parameters compared to other models, which causes its high speed. Also, this model has a simple architecture compared to later models, so its interpretation is quite simple. For these reasons, ResNet18 was chosen as the base solution.

Despite the advantages of convolutional neural networks, they still have one significant drawback - the difficulty of explaining model predictions. In other words, often neural networks are considered as a black box [5]. It can be critical when debugging a model.

At the moment, there are a large number of methods for interpreting neural networks, almost all of them are aimed at connecting parts of the input data with the prediction of the model [6]. For example, in image processing tasks, these methods help to highlight areas of the input image that led to a particular prediction.

In this paper, it is proposed to consider an example of applying existing methods for interpreting neural networks to identify the causes of systematic errors of a neural network in the problem of classifying the orientation of images of text documents. The very definition of the orientation of a text document is necessary for the correct recognition of text in the image. And often it is the image orientation correction module that is one of the first in the pipeline for text recognition. That is, the quality of text selection strongly depends on the work of this module. Therefore, fixing the errors of this module is necessary, but it may not be a trivial task at all.

In general, this paper considers several interpretation methods that allowed us to identify artifacts in the original training data. Also, the paper compares interpretation methods using several quality assessment metrics, one of which is proposed in this paper.

2. Related work

Consider interpretation methods for neural networks designed for image processing. According to work [7], existing methods for convolutional neural networks interpretation can be divided into two groups:

- Techniques to understand the decision-making process of a neural network by correlating the output of a neural network with areas of input data to see which parts of the input data most determine the output. Such methods can also be called local;
- Methods that allow us to consider in more detail the contents of the neural network and interpret how the internal layers' process data (not necessarily data in our subject area, but in the general case). Such methods could be called global.

The first group includes methods based on gradients as well as methods that allow you to build areas importance map of the input image. The main representatives of gradient methods are Vanilla Gradient [8], Guided Backpropagation [9], Integrated Gradients [10], GradCAM [11] and XRAI [12].

The second group includes methods for visualizing extracted features, the main of which is the Convolutional Features Visualization [13] method. In next sections global and local methods are described in more detail.

2.1 Vanilla Gradient

One of the first methods to explain neural networks is Vanilla Gradient [8]. For explaining the result of the classifier's work on the image, the method visualizes a matrix, which represents partial derivatives of the neural network output with respect to the input image. In the resulting matrix, the intensity of each pixel reflects the derivative of the output with respect to the corresponding image pixel. It is assumed that the greater the intensity of a pixel on a heat map, the more it affects the prediction of the classifier, i.e. the more this pixel is responsible for assigning it to a particular class.

2.2 Guided Backpropagation

A modification of the Vanilla Gradient method is the Guided Backpropagation method [9]. It differs from Vanilla Gradient in that all negative gradients are set to zero to build a feature map. It is assumed that positive gradients are positively correlated with the prediction of a particular class. In practice this makes the selected feature map less noisy. This method, like Vanilla Gradient, allows you to see which areas of the input image are most responsible for a particular neural network prediction.

2.3 Integrated Gradients

Another modification of the Vanilla Gradients method is the Integrated Gradients method [10]. This method differs from Vanilla Gradients and Guided Backpropagation in that it allows you to explore the neural network, as a function, not only at the point corresponding to the input image, but at some interval. It is assumed that the neural network, as some complex mathematical function, can fall into a local extremum when a certain input image is supplied. In this case, when calculating partial derivatives with respect to the input image, zero values can be obtained, which will not carry information about the importance of image areas for predicting a certain class. To consider a neural network on a certain interval, it is proposed to use the Integrated Gradients method on different combinations of inputs $x' + \alpha(x - x')$, where x is the original input image, x' is an auxiliary image that does not contain any information (e.g. white noise or a completely black image), and α is a coefficient from 0 to 1, indicating in what proportion the auxiliary image is added. That is, the neural network function is considered on the interval $[x', x]$ and integrated over α . The final formula for the value map looks like this:

$$\text{IntegratedGradients}_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x'_i + \alpha \times (x_i - x'_i))}{\partial x_i} d\alpha$$

Here $F(x)$ is a mathematical function representing a neural network.

This method allows to accumulate gradient values for different inputs, which gives a more stable map of values and allows you to more accurately see the most important areas of the image for prediction.

2.4 GradCAM

Method of interpreting neural networks most frequently encountered in modern literature is the GradCAM [11]. It also produces a heat map showing which parts of an image contribute to specific class prediction the most. The principle of operation of this method can be schematically represented in the form of the following list:

- A convolutional layer is selected. Usually this is the last convolutional layer of the neural network, since it highlights the most important information;
- Matrix of the selected layer activation is saved during forward propagation;

- Partial derivatives are taken from the resulting array relative to the input image and the matrix of derivatives is stored on the selected layer;
- Two saved matrices are multiplied to form a heat map.

3. Interpretation methods justification

Four methods were chosen for the experiments, namely Vanilla Gradient, Guided Backpropagation, Integrated Gradients and GradCAM. These methods are suitable for the specifics of the task and are suitable for working with images of text documents.

XRAI is not suitable for the specifics of the task, because it uses an image segmentation algorithm that separates objects by color and shape. Since the images we work with consist mostly of white backgrounds and black text, meaningful segmentation is not possible: there are not enough different colors and different shaped areas in the image. This method is more suitable, for example, for images of the ImageNet type, since there are objects of different shapes and colors on the images and the background is not monotonous.

Convolutional Features Visualization is also not suitable for interpretation in our case, because the feature images that this method renders themselves require interpretation and do not reveal any human-understandable features in the input images. Examples are shown in the Fig. 1.

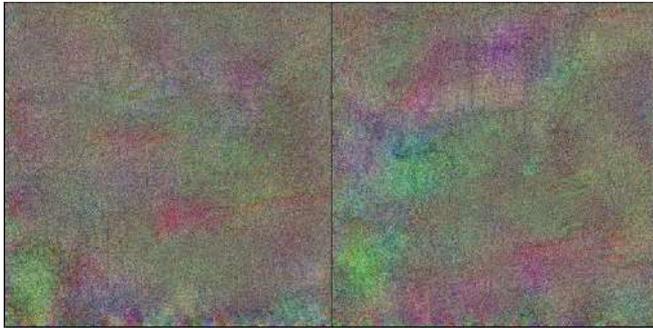


Fig. 1. Examples of visualized features for two convolutional filters of ResNet 18

4. Data set

To solve the problem of determining the document image orientation a data set was generated in order to train and test the selected model. The following is a detailed description of the data set.

2370 images of documents were taken as initial data, initially having a strictly vertical orientation (i.e. 0°). Among the documents have included in the data set, the following types can be distinguished: articles, terms of reference, statements, regulations, laws.

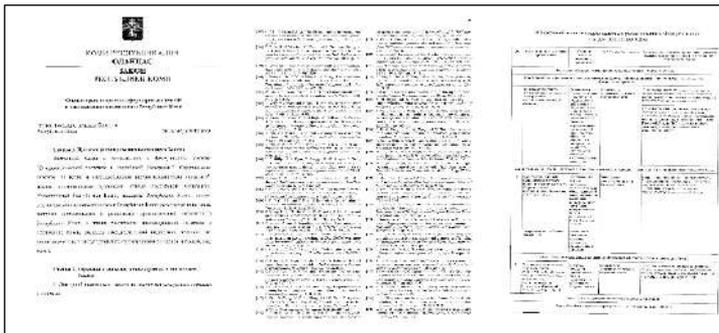


Fig. 2. Examples of documents used in the data set. Left – one column text and a small image. Middle – two column document. Right –table document

Document are characterized by black text on a white background, Manhattan document page layout, and can also contain tables and pictures. Text on different types of documents can consist of one of two columns. Examples are shown in the Fig. 2.

From the images described above, a data set was further compiled for training the classifier using the following augmentation method: each image was rotated by angles that are multiples of 90° and saved as a separate copy. That is, a document that initially had a vertical orientation (i.e. 0°) was presented in four possible orientations in the data set: 0°, 90°, 180° and 270°.

This image rotation was applied so that all document orientation directions occur the same number of times in the data set. Thus, out of 2370 images, a set containing 9480 images was obtained. The training set consisted of 7580 images, while the test set consisted of 1900. It is worth noting that the data set was partitioned by source images, meaning rotated versions of the same image only occur in one part of the data set.

In the process of forming the training data set, a rotation function based on affine transformations was initially used. That is, for the corresponding rotation angle, a transformation matrix was formed, which was multiplied by the image matrix.

5. Interpretation quality estimation

The main way to estimate the quality of interpretation heat maps was proposed in paper [14]. The main idea of this method is to evaluate how accurately heat maps reflect the most important areas of the image for prediction. Thus, leaving the 20% of the most intense pixels in the image, according to the heat map, it is assumed that the class prediction score will change slightly if the heat map well indicated the most important areas. If the class prediction score changes significantly, it can be concluded that the heat map did not accurately reflect the most important areas. The exact formula is given below.

$$ADS (\%) = \frac{1}{N} \sum_{i=1}^N \frac{\max(0, Y_i^c - O_i^c)}{Y_i^c} * 100.$$

Here i is the index of picture in data set, c is the index of class in the data set, Y_i^c is the class prediction score on the original image, O_i^c is the class prediction score on the image, where, according to the heat map, 20% of the most intense pixels are left, the remaining pixels are painted white, N is the number of images in the data set. The example of heat map overlay is shown on Fig. 3.

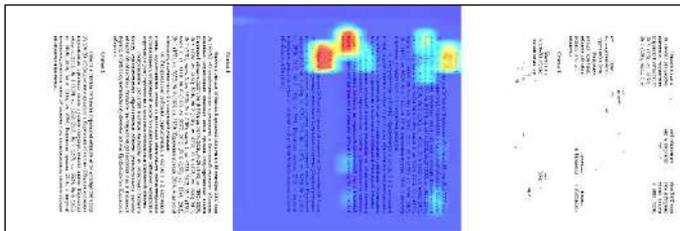


Fig. 3. Left - original input image. Middle - heat map obtained with interpretation method overlaid on image. Right - 20% most intense pixels left on original image according to heat map

6. Heat map result analysis

The first ResNet 18 interpretation experiment was aimed at identifying possible areas of images that lead to mispredictions. GradCAM was chosen as the interpretation method for the first experiment, and the last convolutional layer was chosen as the studied layer of the neural network, since it contains information about the most accurate features of the image. As the images on which the interpretation was carried out, images from the test part and images outside the data set were selected. In the resulting heat maps, red indicates the areas “more important” for prediction, and purple — “less important”.

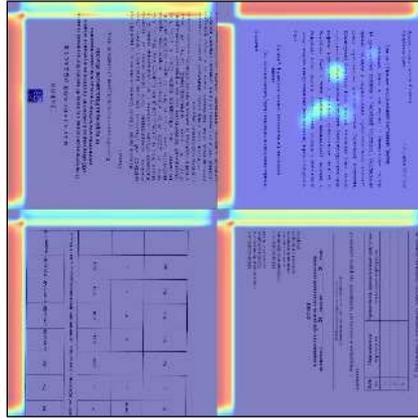


Fig. 4. Several examples of GradCAM heat maps obtained in the first interpretation experiment

Most of the received heat maps contained the same pattern – light stripes along the edges of the images. This means that the neural network “pays attention” to these corner areas. This result does not seem predictable, since, in theory, the neural network should highlight features from the entire image. In the following sections, this effect is discussed in more detail.

6.1 Alleged problem

In connection with the obtained heat maps, which indicated stripes along the edges as important for predicting image areas, an assumption arose about the presence of such artifacts in the images. At the same time, these artifacts, according to the assumption, should be contained either only in the training data set, or only outside it, since the pictures inside these groups are processed by the network in a similar way.

Using a difference visualization program, it was found that the same image, taken from the data set, and the same one rotated in a graphics editor, differ from each other by a shift of one pixel either vertically or horizontally, forming a black stripe at the edge of the image. In this case, this stripe is contained in the image from the data set.

By assumption, it was this artifact in the data set that influenced the fact that the neural network “pays attention” to the areas where these stripes are contained. The very reason for the occurrence of the artifact is discussed in more detail in the next part. Also, further experiments follow, allowing to check the put forward assumption.

6.2 Image rotation function

The *WarpAffine* rotation function of the OpenCV library, originally used in data set generation, is based on affine transformations. A rotation matrix is built from the rotation angle and applied to the matrix of the original image. For each pixel in the original image, the coordinates in the rotated image are calculated. But due to the rounding of floating-point numbers to integers, there is a shift by one pixel, as a result of which an artifact appears in the image in the form of a black stripe at the edge of the image. The location of the line also depends on the angle of rotation. Schematically, the location of the stripes is shown in the Fig. 5.

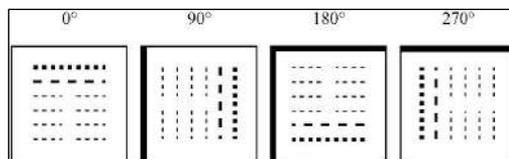


Fig. 5. Visualization of artifacts that appear when using the *WarpAffine* function

The rotation function, based on the transposition and reflection of the image matrix, does not create artifacts in the form of stripes, since in fact it simply rearranges the elements of the matrix.

6.3 Interpretation metrics introduction

The following metric was proposed for assessing the quality of interpretation of the neural network model to test the hypothesis put forward: the ratio of the average pixel intensity in the corner area (indicated in red) and the average pixel intensity in the central area (indicated in blue). The dimensions that define these regions vary in experiments. For *A*, these are 50 and 110 pixels; for *B*, these are 1150 and 1090 pixels, respectively. Let's call this metric corner to main ratio (CMR).

Recall that it is in the corner area that black stripes are contained that appear when using the rotation based on affine transformations. Thus, the metric clearly shows how much the pixels in the corner area are “more important” for the classifier than in the central area, that is, how much these bands are “more important” for the classifier than the rest of the image on a prediction step.

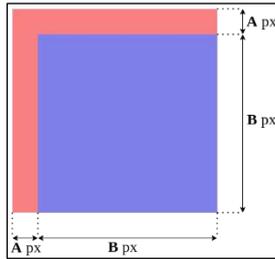


Fig. 6. Splitting the image into corner (marked in red) and central (marked in blue) areas

6.4 Fixed data set

After assuming the influence of artifacts of the rotation function on the work of the classifier, the data set was obtained from the original images in the same way, but using a different rotation function. Now, a combination of transpose and reflection of the image matrix was used as the rotation function. These images do not contain artifacts, since this rotation function uniquely rearranges the elements of the image matrix. The neural network was trained on a new data set for further experiments. In next experiments we will refer to these neural network weights as «fixed model» since this model does not have such unpredictable behavior. And we will refer to these pictures as “Trans+Ref” since such rotation function was used to generate them. We will also call the weights on which unexpected behavior of the neural network was detected an «error model», and pictures that were used to train it “Affine” since this rotation function was used.

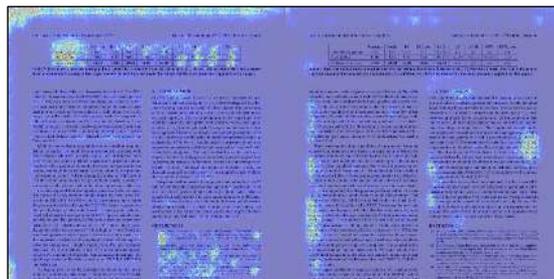


Fig. 7. The heat map obtained with Vanilla Gradients for Trans+Ref image. Left is a heat map for error model, right is a heat map for fixed model

7. Results of interpretation experiments

For experiments, 1000 images were selected from the data set, of which 500 were the results of the *WarpAffine* rotation function, and another 500 were the results of transposition and reflection. As a

result, the corner to main ratio metric was averaged over the entire data set. The calculation of this metric was carried out for two different values of A from Fig. 6: for 110 and 50 pixels. The higher the corner to main ratio, the more important the corner area is compared to the main part of the image.



Fig. 8. The heat map obtained with Guided Backpropagation for Trans+Ref image. Left is a heat map for error model, right is a heat map for fixed model

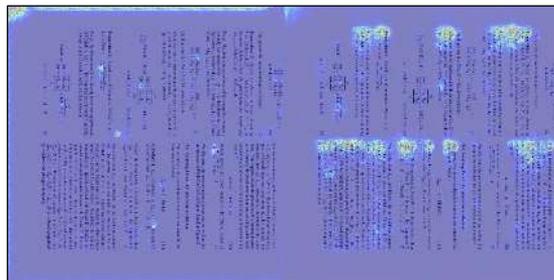


Fig. 9. The heat map obtained with Integrated Gradients for Trans+Ref image. Left is a heat map for error model, right is a heat map for fixed model



Fig. 10. The heat map obtained with GradCAM for the output layer of each of the four ResNet 18 blocks. Left is the conv_4, right is the conv_1. Bottom row contains heat maps for error model, Top row contains heat maps for fixed model. Trans+Ref image was used.

It is visible on Fig. 7, 8, 9, and 10 that the *error model* definitely "pays more attention" to the corner region of an input image however it does not contain artifacts since it is *Trans+Ref* image.

For the GradCAM method, CMR metric is calculated using heat maps obtained on the last convolutional layers of each of the four main blocks. In the Fig. 11, these blocks are named *conv_4*, *conv_3*, *conv_2* and *conv_1*, respectively.

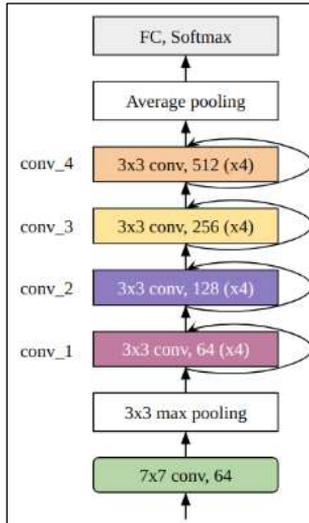


Fig. 11. Schematic representation of the ResNet 18 architecture

7.1 Artifact influence revealed by experiments

As can be seen from the Tables 1, 2, 3, and 4, CMR values are always greater for the *error model* regardless of rotation function used to generate an input image. The results of four chosen methods reveal the same trends, which strengthens the evidence.

Table 1. Corner to main ratio values for Vanilla Gradients method

Corner region width	Affine image		Trans+Ref image	
	error model	fixed model	error model	fixed model
110 px	3,3268	1,2885	2,9800	1,2986
50 px	6,0972	1,4316	5,5656	1,4549

Table 2. Corner to main ratio values for Guided Backpropagation method

Corner region width	Affine image		Trans+Ref image	
	error model	fixed model	error model	fixed model
110 px	1,1163	0,8657	1,2541	0,8860
50 px	1,8716	0,8876	2,1454	0,9268

Table 3. Corner to main ratio values for Integrated Gradients method

Corner region width	Affine image		Trans+Ref image	
	error model	fixed model	error model	fixed model
110 px	3,7876	1,2900	5,1387	1,3758
50 px	6,9799	1,4006	10,0370	1,5589

These results confirm our assumption about the effect of artifacts in the data set on the behavior of the classifier. It turns out that it were these artifacts that became the most important feature for classification, and when this feature did not appear on the image, this led to errors. The proof of this is the accuracy of the *error model* when processing *Affine* images. For example, this is reflected in Tables 5, 6 in the columns "Acc before heat map overlay". Namely, when processing *Trans+Ref* images, the accuracy is 0.5, while when processing *Affine* images, the accuracy is 1.0. At the same

time, for the *fixed model*, when processing *Trans+Ref* images, the accuracy is 0.998, and when processing *Affine* images, the accuracy is 1.0.

Table 4. Corner to main ratio values for GradCAM method

Corner region width		Affine images		Trans+Ref images	
		error model	fixed model	error model	fixed model
110 px	conv_4	2,3177	0,8230	2,3445	0,8263
	conv_3	1,5510	1,0686	1,7669	1,0744
	conv_2	1,0826	0,8793	1,0429	0,8916
	conv_1	0,9085	0,7238	0,9157	0,7645
50 px	conv_4	2,1987	0,6608	2,2196	0,6625
	conv_3	2,2600	1,1567	2,4967	1,1735
	conv_2	1,0512	0,8998	1,0259	0,9017
	conv_1	0,9354	0,7511	0,9516	0,7855

Table 5. Results of the experiment for interpretation quality estimation for the methods Vanilla Gradient, Guided Backpropagation, Integrated Gradients. Bold indicates ADS less than 10 percent

	Trans+Ref image				Affine image			
	Acc before heat map overlay	Acc after heat map overlay	Acc diff	ADS, %	Acc before heat map overlay	Acc after heat map overlay	Acc diff	ADS, %
Vanilla Gradient								
error model	0,500	0,5020	0,002	19,32	1,000	0,9980	0,002	0,58
fixed model	0,998	0,8880	0,110	13,24	1,000	0,8920	0,108	12,56
Guided Backpropagation								
error model	0,500	0,5100	0,010	9,79	1,000	0,8240	0,176	19,88
fixed model	0,998	0,9960	0,002	0,34	1,000	1,0000	0,000	0,15
Integrated Gradients								
error model	0,500	0,4700	0,030	34,28	1,000	0,8760	0,124	14,69
fixed model	0,998	0,7500	0,248	27,91	1,000	0,7580	0,242	26,97

Table 6. Results of the experiment for interpretation quality estimation of GradCAM heat map for each of four ResNet 18 blocks. Bold indicates ADS less than 10 percent

		Trans+Ref image				Affine image			
		Acc before heat map overlay	Acc after heat map overlay	Acc diff	ADS, %	Acc before heat map overlay	Acc after heat map overlay	Acc diff	ADS, %
error model	conv_1	0,500	0,4580	0,042	24,46	1,000	0,6520	0,348	36,39
	conv_2	0,500	0,3940	0,106	34,71	1,000	0,5820	0,418	44,66
	conv_3	0,500	0,3800	0,120	39,18	1,000	0,7740	0,226	24,12
	conv_4	0,500	0,4220	0,078	34,91	1,000	0,7680	0,232	25,82
fixed model	conv_1	0,998	0,7800	0,218	26,65	1,000	0,8100	0,190	23,36
	conv_2	0,998	0,8440	0,154	19,77	1,000	0,8200	0,180	20,97
	conv_3	0,998	0,7960	0,202	23,13	1,000	0,8180	0,182	21,66
	conv_4	0,998	0,8840	0,114	16,21	1,000	0,8740	0,126	16,65

7.2 Quality estimation

In order to assess how accurately the interpretation heat maps were obtained, it is proposed to evaluate the ADS metric described in section 5. This metric will also allow us to understand which of the interpretation methods used turned out to be more accurate.

From the results of the experiment in Tables 5 and 6 we can conclude that the most accurate heat maps were obtained by the Guided Backpropagation method, since The ADS of this method turned out to be the smallest. Moreover, difference in accuracy after heat map overlay is also smaller for Guided Backpropagation. This means that the heat maps of this method most accurately highlight important areas in the image, and therefore masking the image while leaving these most important areas does not lead to a noticeable decrease in the quality of the model.

8. Conclusion

In this work, we have considered how interpretation methods can be applied to a neural network in case of unexpected behavior. We have shown that interpretation can help identify the causes of erroneous behavior and further help improve model accuracy. At the beginning, thanks to the interpretation methods, an assumption was made about the influence of artifacts in the training data set on the behavior of the model. It was found that these artifacts in the form of black stripes with a width of one pixel occur at the edges of images due to the use of a rotation function based on affine transformations. A metric was introduced to test the proposed assumption. With the help of four interpretation methods and this metric, it was possible to accurately prove the hypothesis. All methods showed similar results. Using the average drop in score metric, we managed to choose the best of the four methods.

References

- [1] Wang J., Yang Y. et al. CNN-RNN: A Unified Framework for Multi-label Image Classification. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2285-2294.
- [2] Milletari F., Navab N., Ahmadi S.A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proc. of the Fourth International Conference on 3D Vision (3DV), 2016, pp. 565-571.
- [3] Xie X., Cheng G. et al. Oriented R-CNN for Object Detection. In Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3500-3509.
- [4] He F., Liu T., Tao D. Why ResNet Works? Residuals Generalize. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, issue 12, 2020, pp. 5349-5362.
- [5] Buhmester V., Münch D., Arens M. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *Machine Learning and Knowledge Extraction*, vol. 3, issue 4, 2021, pp. 966-989.
- [6] Li G., Yu Y. Visual Saliency Detection Based on Multiscale Deep CNN Features. *IEEE Transactions on Image Processing*, vol. 25, issue 11, 2016, pp. 5012-5024.
- [7] Barredo-Arrieta A., Díaz-Rodríguez N. et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, vol. 58, 2020, pp. 82-115.
- [8] Simonyan K., Vedaldi A., Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv preprint arXiv:1312.6034, 2013, 8 p.
- [9] Springenberg J.T., Dosovitskiy A. et al. Striving for Simplicity: The All Convolutional Net. arXiv preprint arXiv:1412.6806, 2014, 14 p.
- [10] Sundararajan M., Taly A., Yan Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3319-3328.
- [11] Selvaraju R.R., Cogswell M. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proc. of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618-626.
- [12] Kapishnikov A., Bolukbasi T. et al. XRAI: Better Attributions Through Regions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4948-4957.
- [13] Olah C., Mordvintsev A., Schubert L. Feature Visualization, 2017. Available at: <https://distill.pub/2017/feature-visualization/?ref=hackernoon.com>, accessed May 18, 2023.
- [14] Desai S., Ramaswamy H.G. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 972-980.

Information about authors / Информация об авторах

Alexander Olegovich GOLODKOV – Graduate of the Moscow Institute of Physics and Technology, senior laboratory assistant. Research interests: optical character recognition, computer vision.

Александр Олегович ГОЛОДКОВ – выпускник Московского физико-технического института, старший лаборант. Сфера научных интересов: оптическое распознавание символов, компьютерное зрение.

Oksana Vladimirovna BELYAEVA is a PhD student, researcher. Research interests: document layout analysis, document structure analysis, digital image processing, neural network data processing, image pattern recognition, face recognition.

Оксана Владимировна БЕЛЯЕВА является аспирантом, стажером-исследователем. Научные интересы: анализ шаблонов документов, анализ структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов компьютерного зрения, распознавание лиц.

Andrey Igorevich PERMINOV is a PhD student, researcher. His research interests include digital signal processing, neural network data processing, generation of artificial data.

Андрей Игоревич ПЕРМИНОВ является аспирантом, стажером-исследователем. Его научные интересы включают цифровую обработку сигналов, нейросетевую обработку данных, создание искусственных данных.

DOI: 10.15514/ISPRAS-2023-35(2)-2



Генерация изображений рукописного текста на русском языке

¹А.О. Богатенкова, ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>

²О.В. Беляева, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>

²А.И. Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

¹Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1

²Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

Аннотация. Задача автоматического распознавания рукописного текста является важной составляющей в процессе анализа электронных документов, однако её решение все еще далеко от идеала. Одной из основных причин сложности распознавания рукописного текста на русском языке является недостаточное количество данных, используемых для обучения моделей распознавания. При этом, для русского языка проблема встаёт более остро и усугубляется большим разнообразием сложных почерков. В данной работе исследуется влияние различных методов генерации дополнительных обучающих наборов данных на качество моделей распознавания: метод на основе рукописных шрифтов, метод склейки слов из символов StackMix, метод на основе генеративно-состязательной сети. В рамках данной работы был разработан новый метод создания изображений рукописного текста на русском языке на основе шрифтов. Кроме того, предлагается алгоритм формирования нового кириллического рукописного шрифта на основе имеющихся изображений рукописных символов. Эффективность разработанного метода проверялась с помощью экспериментов, которые проводились на двух общедоступных кириллических наборах данных с помощью двух различных моделей распознавания. Результаты экспериментов показали, что разработанный метод генерации изображений позволил повысить точность распознавания рукописного текста в среднем на 6%, что сравнимо с результатами других, более сложных методов. Исходный код экспериментов, предложенного метода, а также сгенерированные в процессе экспериментов наборы данных выложены в открытый доступ и готовы для скачивания.

Ключевые слова: распознавание рукописного текста; генерация рукописного текста; глубокое обучение; компьютерное зрение

Для цитирования: Богатенкова А.О., Беляева О.В., Перминов А.И. Генерация изображений рукописного текста на русском языке. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 19-34. DOI: 10.15514/ISPRAS-2023-35(2)-2

Generation of images with handwritten text in Russian

¹A.O. Bogatenkova, ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>

²O.V. Belyaeva, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>

²A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

¹Lomonosov Moscow State University,

GSP-1, Leninskie Gory, Moscow, 119991, Russia

²Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

Abstract. Automatic handwriting recognition is an important component in the process of electronic documents analysis, but its solution is still far from ideal. One of the main reasons for the complexity of Russian handwriting recognition is the insufficient amount of data used to train recognition models. Moreover, for the Russian language the problem is more acute and is exacerbated by a large variety of complex handwriting. This paper explores the impact of various methods of generating additional training datasets on the quality of recognition models: the method based on handwritten fonts, the StackMix method of gluing words from symbols, and the use of a generative adversarial network. A font-based method for creating images of handwritten text in Russian has been developed and described in this work. In addition, an algorithm for the formation of a new Cyrillic handwritten font based on the existing images of handwritten characters is proposed. The effectiveness of the developed method was tested using experiments that were carried out on two publicly available Cyrillic datasets using two different recognition models. The results of the experiments showed that the developed method for generating images made it possible to increase the accuracy of handwriting recognition by an average of 6%, which is comparable to the results of other more complex methods. The source code of the experiments, the proposed method, as well as the datasets generated during the experiments are posted in the public domain and are ready for download.

Keywords: handwritten text recognition; handwritten text generation; deep learning; computer vision

For citation: Bogatenkova A.O., Belyaeva O.V., Perminov A.I. Generation of images with handwritten text in Russian. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 19-34 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-2

1. Введение

Рукописные записи повсеместно используются в нашей повседневной жизни, как правило, в заметках, списках или других коротких текстах. До изобретения печатного станка в XV веке, рукописи были единственным способом передачи и сохранения информации различного рода. Поэтому огромное количество информации содержится в рукописном виде в исторических документах. Кроме того, рукописный текст систематически используется в других областях, например, в написании конспектов на академических занятиях, на деловых встречах или при заполнении различных бланков и заявлений.

Несмотря на распространенность и удобство использования рукописных записей различного рода, этот способ в настоящее время не является предпочтительным. Основной недостаток рукописей связан с исключительной трудностью их цифровизации с целью более удобного хранения, структуризации и распространения информации. В современном мире большая часть процессов работы с данными автоматизирована, с ними работают компьютеры. Однако компьютер не умеет работать с аналоговыми данными, такими как изображения рукописного текста, эти данные должны быть представлены в понятном для машины виде. В этом контексте способность распознавать и оцифровывать содержимое рукописного текста необходима для извлечения из него необходимой информации.

Автоматическое распознавание рукописного текста (handwritten text recognition, HTR) – задача, которая решается в течение уже довольно продолжительного времени. Она состоит в автоматическом переводе изображений, содержащих рукописный текст, в символическое представление. Эту задачу можно сформулировать следующим образом: пусть $x^{m \times n \times c}$ –

входное изображение шириной m , высотой n и числом каналов c ; $y^t = (y_1 \dots y_t)$, $y_i \in A$, $i = 1, \dots, t$ – выходная последовательность символов из алфавита A ; $X = \{x^{m \times n \times c}, m, n > 0, c \in \{1, 3\}\}$, $Y = \{y^t, 0 < t \leq T\}$ – множества входных изображений и выходных последовательностей соответственно. Задача распознавания рукописного текста состоит в определении отображения: $X \rightarrow Y$, задающего для каждого изображения рукописного текста его цифровое представление в виде последовательности символов. В настоящее время задача в общем виде ещё не решена и активно исследуется.

Отдельно следует сказать о распознавании рукописного текста на русском языке. Большинство исследований в области распознавания ведется для текста на английском языке, либо на языке, основой которого являются латинские символы. Вследствие этого намного проще найти и наборы данных, и методы решения задачи для текста на таких языках.

При этом существует лишь несколько работ [1, 2], посвящённых распознаванию текста на кириллице, равно как и небольшое количество эталонных наборов данных, используемых для сравнения результатов с другими методами. Всё это добавляет дополнительные сложности на пути решения задачи к существующим многочисленным проблемам.

Как правило, для решения задачи распознавания рукописного текста применяются методы машинного и глубокого обучения, требующие большого объема разнообразных обучающих данных. Так, одной из основных причин отсутствия хорошей универсальной модели распознавания рукописного текста является то, что размер обучающих данных недостаточно велик. В этом случае к существующим данным применяют методы аугментации данных, в рамках рассматриваемой задачи мы остановимся на рассмотрении подвида аугментации – генерации дополнительных синтетических данных.

Как правило, расширение обучающего набора данных осуществляется путем генерации изображений рукописного текста с помощью рукописных шрифтов. Эта методика характерна для англоязычных текстов, для которых создано большое количество разнообразных шрифтов. Несмотря на относительную простоту реализации этого метода, в литературе не упоминается его использование в контексте русского языка. Это может быть связано с меньшим разнообразием доступных шрифтов, а также слабой изученностью данной темы в принципе. Помимо метода генерации данных с помощью шрифтов, применяют также и другие техники, зачастую требующие обучения специализированных моделей машинного обучения [2, 3]. Изучение эффективности данных методов может позволить улучшить качество обучаемых моделей распознавания рукописного текста, в частности, на русском языке.

Статья организована следующим образом: разд. 2 содержит информацию об общеизвестных эталонных наборах данных, используемых в рамках задачи распознавания рукописного текста на русском языке, а также описание стандартных метрик оценки качества распознавания и существующие методы генерации дополнительных наборов данных для обучения моделей. В разд. 3 описывается предлагаемый метод генерации синтетических изображений рукописного текста на основе шрифтов, а также полуавтоматический метод создания нового шрифта. Разд. 4 содержит детали экспериментальной проверки предложенного метода и его сравнения с существующими, а разд. 5 – описание полученных результатов. Наконец, в разд. 6 представлены краткие выводы о проделанной работе и предлагаются возможные варианты дальнейших исследований.

2. Обзор существующих методов

В этом разделе описываются существующие открытые наборы данных, используемые в рамках решения задачи распознавания рукописного текста на русском языке. Кроме того, приводятся общепринятые метрики оценки качества моделей распознавания, а также методы, применяемые для расширения обучающих наборов данных.

2.1 Наборы данных на русском языке

Существует несколько общедоступных наборов данных на русском языке для обучения моделей и сравнения результатов. В настоящее время известны два набора данных со словами и предложениями на кириллице:

- Cyrillic Handwriting Dataset [4];
- HKR [5].

Основная информация о наборах данных представлена в табл. 1.

Табл. 1. Описание наборов данных с кириллицей

Table. 1. Cyrillic datasets description

	Cyrillic Handwriting Dataset	HKR
Описание	Набор русских текстов длиной не больше 40 символов, собранный из различных Интернет-ресурсов	Набор из русских (95%) и казахских (5%) слов и предложений: ключевые слова, поэмы и алфавит
Размер	train=72286, test=1544	train=45470, val=9359, test1=5057, test2=5057
Уникальные слова	37519	2808
Уникальные почерки	880	–
Фон	Разнообразный фон, встречаются пятна, линии, соседний текст и т.д.	Однообразный светлый фон

Набор данных **Cyrillic Handwriting Dataset** был опубликован в 2022 году, поэтому еще нет работ, содержащих результаты его обработки. Этот набор очень интересен с точки зрения разнообразия данных: в него входят студенческие конспекты, заполненные формы, электронные рукописные документы. Изображения содержат различного рода шумы и неоднородный фон, в некоторых случаях на изображение одного слова попадают части другого или разлиновка листа. Согласно табл. 1, набор данных состоит из более 70 тысяч обучающих примеров – изображений слов и предложений, что также является его существенным достоинством. Кроме того, он является открытым как для научных исследований, так и для коммерческого использования. Примеры изображений, встречающиеся в наборе данных, представлены на рис. 1.



Рис. 1. Примеры изображений набора Cyrillic Handwriting Dataset

Fig. 1. Examples of the Cyrillic Handwriting Dataset dataset images

Наиболее популярным набором данных на русском языке, упоминаемым в научной литературе, является набор казахских и русских слов и предложений **HKR**. Всего в нем содержится более 60 тысяч изображений слов и предложений, написанных примерно 200 различными почерками. Он создавался путем заполнения однотипных форм, поэтому имеет одну особенность – в нем содержится большое количество копий одного и того же текста,

написанного разными почерками. Отдельно авторами [5] дается разбиение набора данных на тренировочный, валидационный и два тестовых набора. Первый тестовый набор содержит слова, которых нет в тренировочном наборе, но написанные почерками, присутствующими в тренировочном наборе. Напротив, второй тестовый набор содержит слова, которые есть в тренировочном наборе, но написанные “новыми” почерками. Эта особенность позволяет провести анализ того, на что в большей степени обращает внимание обучаемая модель: новые способы написания символов или новые сочетания.

Несмотря на значительный объем набора, изображения в нем достаточно хорошего качества и относительно однообразны. Таким образом, произвольное изображение рукописного текста из “реального мира” совершенно не похоже на то, что содержится в описанном наборе данных. В добавление к этому, использование данного набора ограничивается научными исследованиями, для его использования в коммерческих целях необходимо обратиться к его авторам. Примеры изображений, встречающиеся в наборе данных, представлены на рис. 2.

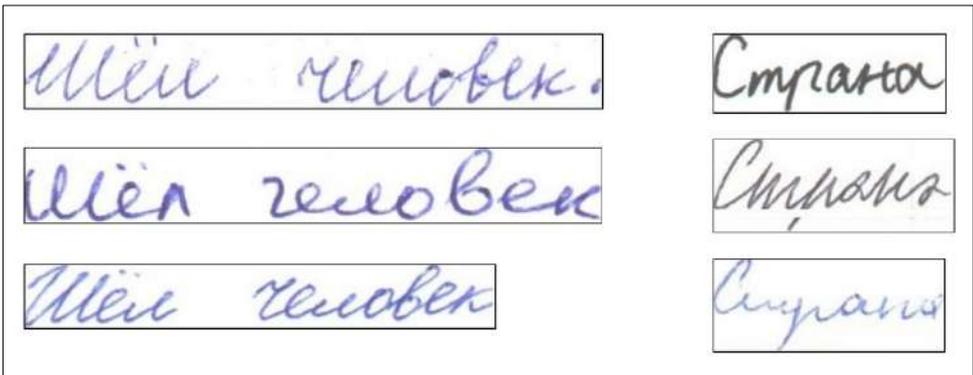


Рис. 2. Примеры изображений набора HKR
Fig. 2. Examples of the HKR dataset images

Таким образом, суммарный размер вышеописанных наборов не превышает 140 тысяч изображений рукописных слов и предложений, а уникальных текстов – не более 40 тысяч, что сигнализирует о недостатке обучающих данных, которые необходимы в большем количестве в силу сложности задачи. Особое внимание следует уделить разнообразию текстов, которое можно увеличить с помощью методов генерации синтетических изображений рукописного текста.

2.2 Метрики оценки качества

Двумя основными метриками, обычно используемыми для оценки моделей распознавания рукописного текста на уровне слов и строк, являются **частота ошибок символов** (Character Error Rate, CER) и **частота ошибок слов** (Word Error Rate, WER).

CER измеряет расстояние Левенштейна [6] между предсказанной и реальной последовательностью символов слова. Расстояние Левенштейна, также иногда называемое расстоянием редактирования, представляет собой метрику для измерения разницы между двумя последовательностями символов. Неформально, расстояние Левенштейна между двумя словами (предсказание модели и реальное слово) – это минимальное количество вставок, удалений или замен, необходимых для преобразования предсказания в правильное слово, делённое на длину правильного слова, как показано в формуле (1):

$$CER(prediction, real) = \frac{substitutions + insertions + deletions}{len(real)}. \quad (1)$$

Частота ошибок в словах (WER) определяется аналогично CER путем вычисления минимального количества вставок, замен и удалений слов, необходимых для перехода от текстовой строки, предсказанной моделью, к реальной текстовой строке.

В некоторых работах наравне с частотой ошибок символов и частотой ошибок слов используется **точность** (accuracy). Данная метрика используется для любых текстовых строк, содержащих как слова, так и предложения.

Точность описывается формулой (2), в которой под равенством подразумевается полное совпадение двух строк:

$$accuracy(prediction, real) = \frac{\sum_{i=1}^N pred_i = real_i}{N}. \quad (2)$$

Данная метрика позволяет оценить качество модели более грубо, так как ошибка в одном символе сильно понижает результирующее значение.

2.3 Методы генерации обучающих наборов данных

Как отмечалось в разд. 1, генерация дополнительного набора данных для обучения моделей распознавания является одной из подзадач, которые возникают в рамках задачи распознавания рукописного текста. Метод генерации новых слов и стилей написания в рукописных текстах вряд ли позволит получить по-настоящему реалистичные и совершенно новые изображения, однако он может дать существенный прирост в качестве результатов обучаемых моделей.

Один из интересных способов генерации новых слов и предложений в стиле уже имеющихся слов в наборе данных, предложили авторы статьи [2], которые также предложили метод аугментации путём рисования пятен с помощью кривых Безье. Свой метод они назвали StackMix и состоит он в разбиении слов на символы или подслова, а затем составлении из этих кусочков новых слов. Пример работы данного метода представлен на рис. 3. Такой метод не решает проблему соединения символов и создания новых стилей, зато он может помочь в создании рукописных слов, которых нет в обучающем наборе данных. Тем не менее, метод Stackmix позволил повысить точность распознавания предлагаемой авторами модели с 71% до 80% на наборе HKR.

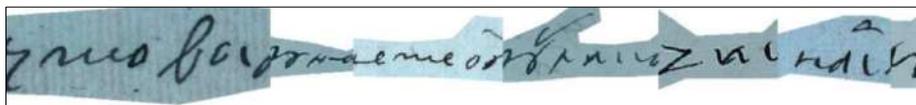


Рис. 3. Пример работы алгоритма Stackmix
Fig. 3. The Stackmix algorithm work example

Следующий метод [7] состоит в генерации изображений рукописного текста с использованием типографских шрифтов. Авторы отобрали 90 тысяч уникальных английских слов и сгенерировали 90 миллионов изображений слов с помощью 750 рукописных шрифтов. Подобный набор слов можно использовать для предобучения моделей распознавания рукописного текста, однако в силу его однообразности потребуются дообучение на реальных данных, чтобы предотвратить переобучение сети. Пример генерации изображения рукописного текста с помощью шрифтов приведен на рис. 4.

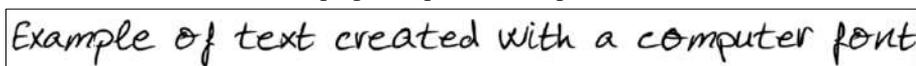


Рис. 4. Пример рукописного текста, полученного с помощью шрифта
Fig. 4. An example of the handwritten text made using a font

В настоящее время активно исследуется целый класс методов генерации рукописного текста, основанных на использовании генеративно-состязательных сетей (GAN) [8]. Такие модели в своей основе содержат две нейронных сети: одна из них генерирует примеры (генеративная модель), а другая пытается отличить подлинные образцы от сгенерированных первой сетью

(дискриминативная модель). Наиболее свежими примерами генеративно-состязательных сетей в области генерации изображений рукописного текста являются ScrabbleGAN [3], GANwriting [9] и TextStyleBrush [10]. Пример результата работы ScrabbleGAN приведен на рис. 5.

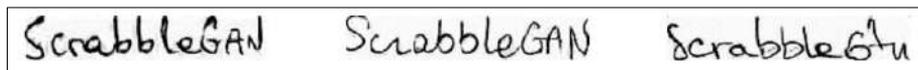


Рис. 5. Пример рукописного текста, полученного с помощью ScrabbleGAN

Fig. 5. An example of the handwritten text made using ScrabbleGAN

Опишем более подробно перечисленные архитектуры генеративно-состязательных сетей для генерации изображений рукописного текста.

- ScrabbleGAN [3] – сеть, состоящая из генератора, дискриминатора и распознавателя символов (OCR). В данной архитектуре дискриминатор влияет на качество изображения, а распознаватель на читаемость текста на изображении. Для обучения сеть использует картинки с текстом и сам текст, стили написания текста меняются с помощью вектора шума, на который домножается входной вектор закодированного текста.
- GANwriting [9] обладает более сложной архитектурой, в которой присутствуют генераторы текста и стиля, соответственно используются дискриминатор, классификатор стиля и распознаватель текста. Вместо одной картинки с текстом, на вход подается группа картинок с текстом, которая формирует конкретный стиль. Поэтому сеть позволяет сгенерировать текст в разных стилях, которые можно смешивать друг с другом.
- TextStyleBrush [10] – наиболее свежая модель, состоящая из 7 нейронных сетей и принимающая на вход картинку со стилем и текст. Здесь также есть генераторы текста и стиля, дискриминатор, классификатор стиля и OCR-модель. Дополняется этот набор сетью, преобразующей выход генератора стиля в набор векторов, а также генератором стилизованного текста, который использует и объединяет результаты работы предыдущих генераторов. Данная модель очень сложна по своей структуре и способна генерировать изображения только тех стилей, на которых была обучена.

Описанные нейросетевые модели генерации обучались и тестировались для английского языка. Соответственно, помимо недостатка, связанного с искусственностью получаемых изображений, имеется и другой – трудоемкость задачи переобучения сети для другого языка. Более того, далеко не все авторы делятся исходным кодом описанных ими моделей.

Таким образом, в научной литературе рассматривается ряд методов, позволяющих увеличить размер обучающего набора данных путём генерации синтетических изображений рукописного текста. При этом лишь один из них [2] был изучен в контексте русского языка, а наиболее простой в реализации и ресурсоёмкости метод на основе шрифтов, насколько нам известно, не применялся с русскими текстами. Поэтому актуально более детальное исследование методов генерации данных применительно к задаче распознавания рукописного текста на русском языке.

3. Описание метода

Согласно разд. 2, метод генерации изображений рукописного текста на основе шрифтов может быть полезен при обучении моделей распознавания, однако он не был использован для русского языка по разным причинам: предположительно, из-за слабой изученности темы, недостатка общедоступных рукописных шрифтов и однообразия получаемых на выходе данных.

Генерация изображений текста с помощью рукописных шрифтов – наиболее простой способ создания дополнительного обучающего набора данных. Этот метод не требует обучения дополнительных моделей, не зависит от какого-либо фиксированного набора данных, прост в реализации и позволяет генерировать данные с большой скоростью. Несмотря на

многочисленные достоинства, генерация текста с помощью шрифтов обладает рядом недостатков. Типографские шрифты, даже рукописные, не обладают достаточным разнообразием и выглядят искусственно не только для нейронной сети, но и для человека.

Как правило, в шрифтах содержится ограниченное количество доступных для написания символов, что влечет за собой зависимость метода от того языка, для которого генерируются дополнительные данные.

Ещё одним существенным недостатком является сложность добавления нового стиля написания, т. е. непосредственно шрифта, так как, как правило, для этого требуется много времени и умение работать со специализированным программным обеспечением. Этот недостаток сильно заметен по отношению к кириллическим шрифтам – существует не так много открытых рукописных шрифтов, содержащих русские символы.

Поэтому имеет смысл каким-то образом автоматизировать процесс создания нового шрифта. В этом может помочь приложение *calligraphr* [11], позволяющее из изображений символов получить шрифт в формате ttf. Изображения рукописных символов можно взять из базы рукописных символов [12], в которой для каждого рукописного кириллического символа существует более 1500 вариантов написания.

Приложение *calligraphr* требует заполнения шаблона с заранее выбранным набором символов. Шаблон представляет собой изображение, содержащее таблицу, в которой в определённых ячейках должны помещаться изображения конкретных символов.

Примеры пустого и заполненного шаблона представлены на рис. 6.

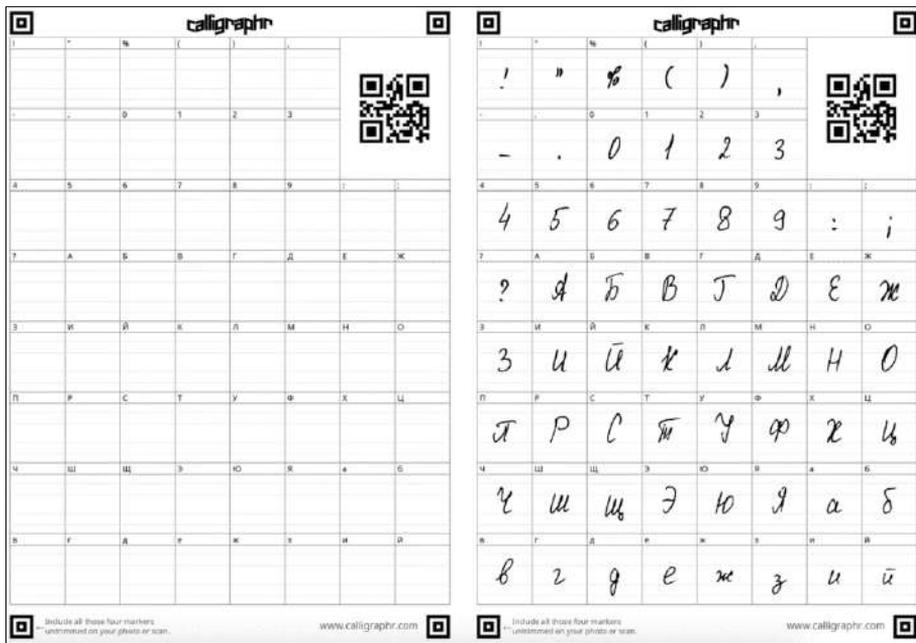


Рис. 6. Примеры пустого и заполненного шаблонов для приложения *calligraphr*

Fig. 6. Examples of the empty and filled templates for the *calligraphr* application

Заполненный шаблон далее анализируется приложением, которое извлекает глифы – векторные изображения символов. У полученных глифов можно вручную поменять расположение и размер, добавляя таким образом реалистичность к результату. Затем из глифов формируется шрифт необходимого формата. Таким образом, создание шрифта можно определенным образом автоматизировать.

Для создания нового кириллического шрифта предлагается следующая последовательность действий:

- 1) заполнение шаблона для *calligraphr* изображениями кириллических символов;
- 2) ручная коррекция полученных с помощью *calligraphr* глифов при необходимости;
- 3) итоговая сборка шрифта.

Последние два пункта алгоритма выполняются непосредственно с помощью приложения. Первый пункт требует анализа границ шаблона для поиска координат вставки изображений, а также нахождения базовой линии символа для того, чтобы вставить его в соответствии с ней. Границы можно найти вручную один раз и зафиксировать для конкретного шаблона. Нахождение базовой линии символа можно с некоторыми оговорками реализовать аналогично нахождению базовой линии слова, для этого существует метод на основе устойчивой регрессии, описанный в работе [13]. В силу того, что для символов базовую линию находить сложнее, требуется дополнительная коррекция на втором шаге алгоритма. Пример одного из шрифтов, полученных с помощью описанного выше алгоритма, представлен на рис. 7.

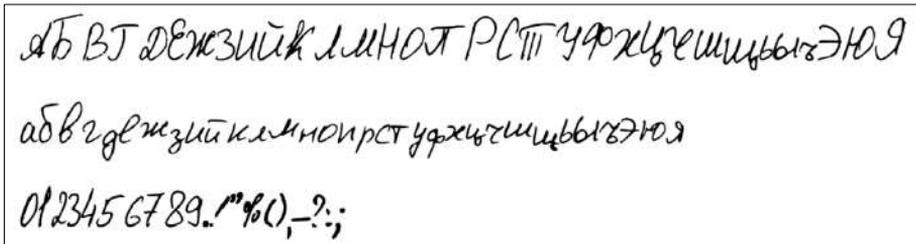


Рис. 7. Пример шрифта, сгенерированного с помощью приложения *calligraphr*
Fig. 7. An example of the font made using the *calligraphr* application

Помимо создания новых шрифтов, можно использовать немногочисленные доступные кириллические рукописные шрифты. Среди общедоступных кириллических шрифтов можно выделить следующие: Abram, Anselmo, Benvolio, Capuletty, Gogol, Lorencо, Pushkin, Voronov и другие, доступные на сайтах <https://localfonts.eu/freefonts/handwritten-cyrillic-free-fonts>, <https://www.rufonts.ru/fonts/rukopisnyj>, <https://allbestfonts.com/category/russian-handwritten>, <https://fontesk.com/font/handwritten/?tag=cyrillic>, <https://www.abstractfonts.com/language/9>, <https://myskotom.ru/shrifoteka>. При наличии нескольких десятков шрифтов можно реализовать генератор рукописного текста с помощью многочисленных библиотек отрисовки текста на изображении (например, библиотека Pillow для языка программирования Python), и далее создавать достаточно разнообразные синтетические изображения рукописного текста. Более того, к имеющимся возможностям можно добавить некоторую рандомизацию, напоминающую аугментацию изображений. При создании изображений можно использовать следующие аугментации:

- изменение размера шрифта, сжатие изображения;
- шумы различных видов (Гауссовский, ISO, мультипликативный);
- размытие различных видов (движения, медианное);
- морфологические операции (эрозия, дилатация) для изменения толщины символов;
- изменение наклона шрифта согласно алгоритму из работы [sueiras2021continuous];
- искажение перспективы изображения;
- небольшие обрезка и повороты изображения;
- имитация рукописных зачеркиваний, описанная в работе [shonenkov2021stackmix];
- вставка обрезанных символов по краям изображения (имитация соседних строк);
- добавление случайных (светлых, темных) пятен на изображение;
- изменение яркости, контрастности, насыщенности изображения.

Таким образом, разработан метод генерации кириллических шрифтов, а также изображений рукописного текста на основе шрифтов. Примеры результатов работы генератора рукописного текста представлены на рис. 8.

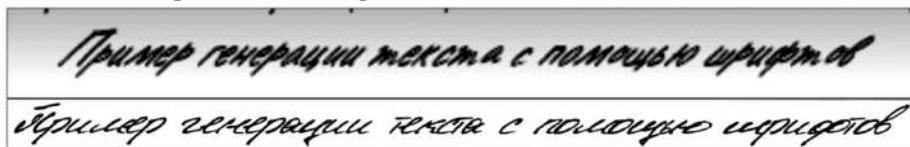


Рис. 8. Примеры результатов работы генератора рукописного текста
Fig. 8. Examples of the handwriting generator's results

4. Описание экспериментов

Для проведения экспериментов зафиксируем модели распознавания рукописного текста, используемые для тестирования методов генерации дополнительных обучающих наборов.

- AttentionHTR [14] – модель архитектуры seq2seq [15], состоящая из сверточного модуля извлечения признаков ResNet [16], рекуррентного модуля разметки последовательности BiLSTM [17] в качестве энкодера, и модуля внимания [18] в качестве декодера. В работе [14] модель была обучена для текстов на английском языке и показала одни из лучших результатов на наборе данных IAM [19].
- Модель с архитектурой трансформер [20] – изначально обучалась на данных английского и русского языка, состоит из свёрточного модуля извлечения признаков ConvNext [21], трансформер-энкодера и двух декодеров: CTC [22] и трансформер [23]. Авторы модели не публиковали подробных результатов на конкретных наборах данных, однако архитектура модели интересна в силу того, что трансформеры являются многообещающим способом решения задачи.

Так как для вышеперечисленных моделей были определены специальные методы стандартизации размера изображений, а также параметры обучения, необходимые для получения наилучшего результата, было решено работать с ними без изменений. При обучении использовались следующие гиперпараметры:

- для AttentionHTR: размер входного изображения 32×100 ; размер выходного слоя ResNet 512; размер скрытого слоя BiLSTM 256; количество эпох 50; оптимизатор AdaDelta со скоростью обучения $\nu = 1$ и константой затухания $\rho = 0.95$; кросс-энтропия в качестве лосс-функции;
- для трансформера: размер входного изображения 96×384 ; размер выходного слоя ConvNext 768; количество эпох 100; оптимизатор MADGRAD со скоростью обучения $1 \cdot 10^{-6}$, настраиваемой с помощью CosineAnnealingLR; сумма кросс-энтропии и CTC-лосса в качестве лосс-функции.

Выбранные модели обучались в течение указанного количества эпох, однако обучение могло остановиться ранее, если в течение 10 эпох не уменьшается значение лосс-функции или метрики CER на валидационном наборе данных. Кроме того, процесс обучения организован так, что на каждой итерации берется одинаковое количество данных из каждого набора для предотвращения несбалансированности обучающих данных.

Изображения, используемые для обучения, подавались на вход моделям без проведения предобработки, так как она может потенциально привести к ухудшению их качества. Также была зафиксирована аугментация данных (искажение существующих изображений):

- адаптивное выравнивание гистограммы с ограниченным контрастом (CLAHE);
- небольшие повороты изображения;
- удаление небольших регионов изображения (Cutout);

- искажение сетки (Grid Distortion);
- размытие изображения;
- сжатие JPEG.

Также зафиксируем методы генерации синтетических данных для их сравнения между собой.

- С помощью рукописных шрифтов (разд. 3). В результате поисков готовых шрифтов и создания новых с помощью базы символов было получено 65 различных рукописных шрифтов. Данные, дополняющие наборы НКР и Cyrillic, создавались похожим образом с отличием в количестве преобразований, применяемых к отрисованным изображениям. Для набора, дополняющего НКР, не использовались преобразования, связанные с добавлением пятен на фон, так как изображения набора отличаются белым фоном без шумов. Для набора, дополняющего Cyrillic Handwriting Dataset, использовались всевозможные преобразования, описанные в главе 3, в силу большого стилового разнообразия изображений, входящих в набор.
- С помощью метода Stackmix [2]. Для использования метода на каждом из наборов НКР и Cyrillic была обучена нейронная сеть, предложенная авторами алгоритма, с рекомендуемыми ими параметрами обучения в течение 100 эпох.
- С помощью генеративно-сопоставительной сети ScrabbleGAN [3]. На каждом наборе была обучена сеть ScrabbleGAN с параметрами, рекомендуемыми авторами исходного кода, в течение 100 эпох. После этого обученным моделям были переданы корпуса текстовых строк, для которых были сгенерированы синтетические данные, расширяющие наборы НКР и Cyrillic Handwriting Dataset.

Каждый из данных методов применялся к фиксированному набору из 300,000 слов и наборов слов, взятых из случайных статей Википедии [24]. Выбранные тексты статей были очищены от символов, не встречающихся в имеющихся наборах данных, а также были оставлены только уникальные элементы. Для наборов данных НКР и Cyrillic Handwriting Dataset были выбраны различные корпуса текстов в силу того, что наборы символов в них сильно варьировались.

Таким образом, выбраны два кириллических набора данных, три метода расширения этих наборов, а также две модели распознавания рукописного текста. Данные модели были обучены на каждом из наборов без добавления каких-либо данных, а также на наборах, дополненных тремя перечисленными выше способами.

5. Результаты

В табл. 2, 3, 4 и 5 представлены результаты экспериментов, описанных в главе 4. Табл. 2 и 3 содержат результаты обучения модели AttentionHTR, а табл. 4 и 5 – модели трансформер. Табл. 3 и 5 отражают более детальные результаты для набора данных НКР, в котором присутствуют два тестовых набора, описанных в разд. 2.

Табл. 2. Результаты обучения нейронной сети AttentionHTR
Table. 2. The AttentionHTR results

	НКР all			Cyrillic Handwriting Dataset		
	ACC	CER	WER	ACC	CER	WER
Без генерации	63.67	22.55	37.64	37.95	18.53	60.63
Шрифты	64.69	13.55	31.78	49.09	13.12	49.23
StackMix	68.96	8.45	25.04	49.9	12.44	46.03
GAN	64.69	12.09	30.28	37.69	16.68	59.1

Табл. 3. Результаты обучения нейронной сети AttentionHTR на наборе HKR

Table. 3. The AttentionHTR results on the HKR dataset

	HKR test1			HKR test2		
	ACC	CER	WER	ACC	CER	WER
Без генерации	39.12	42.1	67.41	87.86	3.31	8.33
Шрифты	42.32	24.08	55.23	86.73	3.2	8.69
StackMix	53.08	13.64	40.13	84.61	3.34	10.2
GAN	43.11	21.08	51.65	85.96	3.24	9.24

Рассмотрим результаты обучения модели AttentionHTR. Исходя из значений метрик, показанных в табл. 2 и 3, можно сделать вывод, что лучшие результаты по части повышения качества обучения модели AttentionHTR показывает алгоритм StackMix, позволяя повысить точность распознавания (accuracy) на 5% и 12% для наборов HKR и Cyrillic Handwriting Dataset соответственно. При этом, если рассматривать набор Cyrillic Handwriting Dataset, то можно заметить, что результаты обучения на наборе, дополненном изображениями, сгенерированными с помощью шрифтов, не сильно хуже результатов для Stackmix – здесь также наблюдается прирост точности распознавания на 12%.

В дополнение к этому, для набора HKR результаты расширения данных с помощью генеративно-сопоставительной сети ScrabbleGAN практически аналогичны результатам расширения данных с помощью шрифтов, и способ на основе шрифтов явно выигрывает у ScrabbleGAN на наборе Cyrillic Handwriting Dataset. Следует заметить, что использование набора данных, созданного с помощью генеративно-сопоставительной сети, очень слабо повлияло на улучшение качества распознавания для обоих рассматриваемых наборов данных.

Табл. 4. Результаты обучения нейронной сети трансформер

Table. 4. The transformer's results

	HKR all			Cyrillic Handwriting Dataset		
	ACC	CER	WER	ACC	CER	WER
Без генерации	66.68	10.56	29.08	48.57	11.71	44.87
Шрифты	73.94	6.29	21.55	52.59	9.58	42.69
StackMix	76.12	4.91	18.32	59.00	8.35	37.27
GAN	72.18	7.49	23.51	49.09	12.26	47.75

Табл. 5. Результаты обучения нейронной сети трансформер на наборе HKR

Table. 5. The transformer's results on the HKR dataset

	HKR test1			HKR test2		
	ACC	CER	WER	ACC	CER	WER
Без генерации	45.08	19.24	53.33	87.96	2.02	5.22
Шрифты	56.18	10.85	37.85	91.43	1.8	5.5
StackMix	61.47	7.87	30.82	90.56	2.01	6.02
GAN	53.00	13.05	41.68	91.07	2.03	5.62

Рассмотрим результаты обучения модели с архитектурой трансформер. Результаты, показанные в табл. 4 и 5, содержат более высокие показатели, в некотором роде схожие с результатами для AttentionHTR. Здесь также стабильно лидирует алгоритм Stackmix, с помощью которого оказалось возможным поднять точность распознавания (accuracy) примерно на 10%. Кроме того, данные таблиц позволяют сделать вывод о том, что метод генерации синтетических изображений с помощью шрифтов занимает второе место, позволяя поднять точность распознавания на 4–7%. Хуже всех с задачей справился алгоритм

генерации изображений с помощью генеративно-состязательной сети; для набора данных Cyrillic Handwriting Dataset практически не наблюдается каких-либо улучшений в качестве распознавания модели.

В процессе анализа таблиц с результатами обучения моделей (табл. 3 и 5) всплывает еще одна особенность, связанная с набором данных НКР. Как говорилось ранее, тестовый набор НКР делится на две части, обладающие определенными особенностями, которые оказывают сильное влияние на результаты, представленные в таблицах. Тестовая часть test1 содержит новые слова и почерки, на которых модель обучалась, а часть test2 содержит новые почерки, но старые слова, встречающиеся в тренировочной части набора. По полученным значениям метрик можно сделать вывод о том, что выбранные модели сильно переобучаются на фиксированном наборе слов, входящем в тренировочную часть набора НКР. Такой вывод можно сделать благодаря существенному подъему качества распознавания на наборе test1, а также небольшому снижению показателей качества на наборе test2.

По этим результатам видно, что влияние новых почерков на качество распознавания минимально, если еще учитывать тот факт, что все изображения набора имеют похожий стиль. При этом добавление дополнительных обучающих данных с новыми словами позволяет снизить эффект переобучения на фиксированном корпусе и поднять средние значения метрик качества обученной модели на обоих тестовых наборах. Эффект переобучения на словах также может быть связан с тем, что набор данных НКР содержит большое количество изображений с одинаковым текстом: при количестве элементов свыше 45 тысяч, уникальных текстовых единиц всего лишь 2808 (см. табл. 1), некоторые из них незначительно отличаются друг от друга. При этом валидационная часть набора состоит преимущественно из слов, содержащихся в тренировочной части (9133 слова из 9375). Таким образом, обучение только на данных из тренировочной части набора НКР недостаточно оправданно, необходимы дополнительные данные с изображениями новых слов.

По итогам анализа результатов можно сделать вывод о том, что метод генерации дополнительного синтетического набора данных с помощью рукописных шрифтов не сильно уступает другим методам в плане повышения качества обучения моделей распознавания. В среднем, он позволяет повысить точность распознавания (ассурасу) обученной модели на 6%, по сравнению с обучением модели на наборе без использования дополнительно сгенерированных данных. Этот метод имеет одно несомненное преимущество перед другими рассмотренными методами – он не требует обучения тяжеловесных моделей и затачивания под конкретный набор данных.

Код с реализацией экспериментов, а также реализация генератора рукописного текста находятся в общем доступе: <https://github.com/NastyBoget/hrtr> (эксперименты), <https://github.com/NastyBoget/HandwritingGeneration> (генератор). Кроме того, в открытый доступ выложены наборы данных, сгенерированные тремя способами для наборов данных НКР и Cyrillic Handwriting Dataset:

- synthetic_hkr (https://huggingface.co/datasets/nastyboget/synthetic_hkr);
- stackmix_hkr (https://huggingface.co/datasets/nastyboget/stackmix_hkr);
- gan_hkr (https://huggingface.co/datasets/nastyboget/gan_hkr);
- synthetic_cyrillic (https://huggingface.co/datasets/nastyboget/synthetic_cyrillic);
- stackmix_cyrillic (https://huggingface.co/datasets/nastyboget/stackmix_cyrillic);
- gan_cyrillic (https://huggingface.co/datasets/nastyboget/gan_cyrillic).

6. Заключение

В данной статье описывается разработанный авторами метод создания кириллических рукописных шрифтов, а также метод генерации изображений рукописного текста на их основе. Проведено сравнение предложенного метода с существующими решениями путем

обучения двух моделей распознавания рукописного текста. Модели обучались на двух различных общедоступных наборах данных, дополненных изображениями, сгенерированными различными способами. По результатам исследований можно сделать вывод о том, что разработанный метод сравним по результатам повышения качества распознавания моделей с другими методами. Он позволяет повысить точность распознавания обученной модели в среднем на 6%, при этом не требует обучения дополнительных моделей. Код разработанного метода, а также проведённых экспериментов выложен в открытый доступ. Сгенерированные в рамках исследования наборы данных доступны для скачивания на платформе huggingface.co.

Дальнейшие исследования могут быть направлены на повышение качества распознавания выбранных моделей: можно убрать верхний порог количества эпох обучения и более тонко скорректировать гиперпараметры, сгенерировать большее количество обучающих данных, а также одновременно использовать дополнительные наборы данных, сгенерированные тремя вышеописанными методами.

Список литературы / References

- [1] Abdallah A., Hamada M., Nurseitov D. Attention-Based Fully Gated CNN-BGRU for Russian Handwritten Text. *Journal of Imaging*, vol. 6, issue 12, 2020, article no. 141, 23 p.
- [2] Shonenkov A., Karachev D. et al. StackMix and Blot Augmentations for Handwritten Text Recognition. arXiv preprint arXiv:2108.11667, 2021, 10 p.
- [3] Fogel S., Averbuch-Elor H. et al. ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4324-4333.
- [4] Cyrillic Handwriting Dataset. Available at: <https://www.kaggle.com/datasets/constantinwerner/cyrillic-handwriting-dataset>, accessed 02.05.2023.
- [5] Nurseitov D., Bostanbekov K. et al. Handwritten Kazakh and Russian (HKR) database for text recognition. *Multimedia Tools and Applications*, vol. 80, issue 21-23, 2021, pp. 33075 - 33097.
- [6] Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов. Доклады Академии наук СССР, том 163, ном. 4, 1965, стр. 845-848 / Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, vol. 10, no. 8, 1966, pp. 707-710.
- [7] Krishnan P., Jawahar C.V. Generating Synthetic Data for Text Recognition. arXiv preprint arXiv:1608.04224, 2016, 5p.
- [8] Goodfellow I., Pouget-Abadie J. et al. Generative adversarial networks. *Communications of the ACM*, vol. 63, issue 11, 2020, pp. 139-144.
- [9] Kang L., Riba P. et al. GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images. *Lecture Notes in Computer Science*, vol. 12368, 2020, pp. 273-289.
- [10] Krishnan P., Kovvuri R. et al. TextStyleBrush: Transfer of Text Aesthetics from a Single Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2023, 12 p.
- [11] Calligraphr. Available at: <https://www.calligraphr.com>, accessed 02.05.2023.
- [12] База сегментированных рукописных символов / Segmented Handwriting Character Base. Available at: <https://drive.google.com/folderview?id=0B0EQuc5HmgcGS0l2RDlKenlpNnc&usp=sharing>, accessed 02.05.2023 (in Russian).
- [13] Sueiras J. Continuous Offline Handwriting Recognition using Deep Learning Models. arXiv preprint arXiv:2112.13328, 2021, 210 p.
- [14] Kass D. Vats E. AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. *Lecture Notes in Computer Science*, vol. 13237, 2022, pp. 507-522.
- [15] Sutskever I., Vinyals O., Le Q.V. Sequence to sequence learning with neural networks. In Proc. of the 27th International Conference on Neural Information Processing Systems, vol. 2, 2014, pp. 3104-3112.
- [16] He K., Zhang X. et al. Deep Residual Learning for Image Recognition. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
- [17] Hochreiter S., Long Short-term Memory, *Neural computation*, vol. 9, issue. 8, 1997, pp. 1735-1780.
- [18] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473, 2014, 15 p.

- [19] Marti U.-V., Bunke H. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, vol. 5, issue 1, 2002, Pp. 39–46.
- [20] Timakin V., Afanasyev M. A modern approach to the end-to-end bilingual handwriting text recognition on the example of Russian school notebooks. Available at: <https://github.com/t0efL/end2end-HKR-research>, accessed 02.05.2023.
- [21] Liu Z., Mao H. et al. A Convnet for the 2020s. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976–11986.
- [22] Graves A., Fernández S. et al. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proc. of the 23rd International Conference on Machine Learning*, 2006, pp. 369-376.
- [23] Vaswani A., Shazeer N. et al. Attention is all you need. In *Proc. of the 31st Conference on Neural Information Processing System*, 2017, pp. 5998-6008.
- [24] Википедия / Wikipedia. Available at: <https://ru.wikipedia.org>, accessed 02.05.2023 (in Russian).

Информация об авторах / Information about authors

Анастасия Олеговна БОГАТЕНКОВА является студентом магистратуры кафедры мистемного программирования. Научные интересы: распознавание структуры документов, цифровая обработка изображений, создание искусственных данных.

Anastasiya Olegovna BOGATENKOVA is a master's student of the Department of System Programming. Research interests: document layout analysis, digital image processing, generation of artificial data.

Оксана Владимировна БЕЛЯЕВА является аспирантом, стажером-исследователем. Научные интересы: анализ шаблонов документов, анализ структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов компьютерного зрения, распознавание лиц.

Oksana Vladimirovna BELYAEVA is a PhD student, researcher. Research interests: document layout analysis, document structure analysis, digital image processing, neural network data processing, image pattern recognition, face recognition.

Андрей Игоревич ПЕРМИНОВ является аспирантом, стажером-исследователем. Его научные интересы включают цифровую обработку сигналов, нейросетевую обработку данных, создание искусственных данных.

Andrey Igorevich PERMINOV is a PhD student, researcher. His research interests include digital signal processing, neural network data processing, generation of artificial data.

DOI: 10.15514/ISPRAS-2023-35(2)-3



Investigation of Adversarial Attacks on Pattern Recognition Neural Networks

D.V. Kotlyarov, ORCID: 0009-0003-2530-4043 <den14kotlyarov@yandex.ru>
G.D. Dyudyun, ORCID: 0009-0008-1256-0204 <gleb.dudun@gmail.com>
N.V. Rzhetskaya, ORCID: 0009-0002-1285-4196 <natalia070901@gmail.com>
M.A. Lapina, ORCID: 0000-0001-8117-9142 <mlapina@ncfu.ru>
M.G. Babenko, ORCID: 0000-0001-7066-0061 <mgbabenko@ncfu.ru>

*North-Caucasus Federal University,
1, Pushkin Street, Stavropol. 355017, Russia*

Abstract. This article discusses the algorithm for creating a neural network based on pattern recognition. Several types of attacks on neural networks are considered, the main features of such attacks are described. An analysis of the Adversarial attack was carried out. The results of experimental testing of the proposed attack are presented. Confirmation of the hypothesis about the decrease in the accuracy of recognition of the neural network during the implementation of the attack by an attacker was obtained.

Keywords: neural network; machine learning; pattern recognition; artificial intelligence; attack algorithm; information security; Adversarial attack, malicious machine learning

For citation: Kotlyarov D.V., Dyudyun G.D., Rzhetskaya N.V., Lapina M.A., Babenko M.G. Investigation of Adversarial Attacks on Pattern Recognition Neural Networks. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 35-48. DOI: 10.15514/ISPRAS-2023-35(2)-3

Acknowledgements. This work was supported by the Russian Science Foundation 19-71-10033, <https://rscf.ru/project/19-71-10033/>

Исследование состязательных атак на нейронные сети распознавания образов

Д. В. Котляров, ORCID: 0009-0003-2530-4043 <den14kotlyarov@yandex.ru>
Г. Д. Дюдюн, ORCID: 0009-0008-1256-0204 <gleb.dudun@gmail.com>
Н.В. Ржевская, ORCID: 0009-0002-1285-4196 <natalia070901@gmail.com>
М.А. Лапина, ORCID: 0000-0001-8117-9142 <mlapina@ncfu.ru>
М.Г. Бабенко, ORCID: 0000-0001-7066-0061 <mgbabenko@ncfu.ru>

*Северо-Кавказский федеральный университет,
Россия, 355017, г. Ставрополь, ул. Пушкина, 1*

Аннотация. В данной статье рассмотрен алгоритм создания нейросети, базирующую на распознавание образов. Рассмотрены несколько видов атак на нейронные сети, описаны основные особенности таких атак. Проведен анализ Adversarial attack. Приводятся результаты экспериментальной апробации предложенной атаки. Получено подтверждение гипотезы о снижении точности распознавания нейросети при реализации атаки злоумышленником.

Ключевые слова: нейронная сеть; машинное обучение; распознавание образов; искусственный интеллект; алгоритм атак; информационная безопасность; состязательная атака; вредоносное машинное обучение

Для цитирования: Котляров Д.В., Дюдюн Г.Д., Ржевская Н.В., Лапина М.А., Бабенко М.Г. Исследование состязательных атак на нейронные сети распознавания образов. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 35-48. DOI: 10.15514/ISPRAS-2023-35(2)-3

Благодарности. Работа выполнена при поддержке РФФИ 19-71-10033, <https://rscf.ru/project/19-71-10033/>

1. Introduction

In modern times, such concepts as “neural network”, “artificial intelligence” and other end-to-end technologies are used in various fields, such technologies are tightly integrated into our lives and are often used everywhere. Now nobody is surprised using search algorithms from various advanced companies. Even the older generation now often uses voice assistants, without thinking that these are yet another “brainchild” of a neural network. In fact, a neural network can now do quite a lot: generate scientific reports, write poetry or a song, draw a picture that is not much different from the real one, the main thing is to train it correctly.

But such a network can serve both good and evil, depending on the purpose of the developer. So, for example, when recognizing images, an attacker can purposefully introduce errors into the recognition process, trying to force the system to incorrectly recognize the image being processed [1]. As a result, so-called spoofing attacks appear. Often, such attacks can be used in cases where an attacker seeks to disguise himself as another person and thereby commit illegal actions.

2. State-of-the-art

At present, it is difficult to unambiguously define neural networks. After analyzing the study of several authors, we can say that a neural network or Artificial Neural Networks (ANN) is a learning system, which is a certain mathematical model built on the principle of human neurons, as well as its software implementation [2, 3]. Ivanyuk V.A. claims that artificial neural networks can be used to create intelligent decision-making systems, simulation modeling, expert systems [2].

A neural network is a mathematical model made up of interconnected nodes that work together to solve a problem. The nodes are arranged in layers, and each node performs a simple mathematical operation on the input to produce the output [4].

Kachagina K.S. in her research, gives a range of applications of a neural network in everyday life. So, in this study, we can highlight that the ANN is already used in security organizations, law enforcement agencies, at various factories and much more [3]. Therefore, it is very important to organize the security of these systems, since further the scope of neural systems will only grow.

It should be noted that the main tasks of neural networks are reduced to:

- Classification, that is, the separation of a certain object with a certain attribute from others.
- Prediction, this task often serves the interests of the financial world.
- Recognition, which will help to simplify the work, for example, for law enforcement agencies.
- Solving problems without a teacher.

In recent years, there has been an introduction into information and telecommunication systems as a means of identification, and often authentication of users [1]. According to experts, the introduction of such technologies often brings with it massive discontent from the outside. The reason for this was that the ANN is imperfect. Such a system has several vulnerabilities that will be used to disable it [3].

Since each person is unique, by spoofing biometric data, attackers can describe such data mathematically and use it as input to machine learning algorithms to automate the recognition process, and then use such ANN to replace their identity.

There are many attacks on ANN that prevent the system from working properly. An attacker can carry out large-scale attacks without being noticed. For example, in biometric systems, an attacker

can intentionally introduce errors into the process of recognizing biometric data. Ensuring the security of such systems is an important issue.

Article [5] describes how adversarial attacks work by exploiting vulnerabilities in neural networks that can be easily fooled by small noises or modifications to the input data that are imperceptible to humans but can cause the network to misclassify the input data.

Consider some types of attacks on biometric systems that disrupt the recognition process [1]:

- Fast Gradient Sign Method – an attack with noise overlay on the image with each new iteration. This attack is quite effective when constantly analyzing the image. This type of attack is practically unrealizable in the absence of direct access to data.
- Using Infrared LEDs to Change Human Facial Features.
- Overlaying black or white stickers on the image for incorrect recognition.
- The use of devices that allow you to identify a person for another.
- Hostile attacks are a growing concern in the field of artificial intelligence because they can be used to trick neural networks into misclassifying inputs.

One of the first studies on Adversarial Attacks was carried out by Goodfellow et al. [6], who showed that neural networks can be fooled by small noise inputs. Since then, a large amount of research has been done on this topic, including the development of new attack methods and defense strategies.

One of the most common types of contention attacks is the Fast Gradient Sign Method (FGSM), which was introduced in [6]. This method involves calculating the gradient of the loss function with respect to the input data and then modifying the data in the direction of the gradient to maximize the loss. Many subsequent studies have relied on this method, including the iterative FGSM (IFGSM) attack presented by Kurakin et al [7].

Article [5] also explains various types of hostile attacks, such as targeted and non-targeted attacks, and provides examples of real-life applications of hostile attacks, such as manipulating the systems of unmanned vehicles.

Other types of attacks that have been developed include the Jacobian-based saliency map attack [8], which uses the Jacobian matrix to determine the most sensitive input features, and the deep fool attack [9], which generates small perturbations, which minimize the distance between the original input and the misclassified output.

Various strategies have been proposed to protect against these attacks. Adversarial learning involves augmenting the training data with adversarial examples to make the neural network more robust [10], while defensive distillation involves training a separate network to detect adversarial examples [11]. Other protection strategies include randomization, input transformation, and gradient masking [12].

Despite these defense strategies, adversary attacks remain a major threat to machine learning systems. As noted by Akhtar and Mian [13], attacks by the adversary can have serious consequences in the real world, such as causing self-driving cars to misinterpret road signs or medical systems to misdiagnose diseases.

In addition, article [5] discusses some of the techniques that have been developed to defend against adversary attacks, including adversary training and defensive distillation.

In recent years, researchers have also studied the impact of adversarial attacks on object detection systems [14], semantic segmentation [15], and generative models [16]. These studies have shown that enemy attacks are effective against these systems and have proposed new defense methods to improve their reliability.

Despite significant progress in the development of adversarial attacks and defenses, there are still open issues that require further research. One of these tasks is the development of effective and reliable methods of protection. Another challenge is understanding the vulnerabilities of deep learning models to attack by malicious actors and finding ways to fix them.

Several studies [6, 14, 15] have examined the effectiveness of adversarial attacks on various types of Machine Learning (ML) models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and autoencoders. For example, Xu et al. [17] have shown that adversarial attacks can be effective against RNN-based text classifiers even if the attacks are generated using a different language model.

To protect against enemy attacks, researchers have proposed various defense strategies. One common defense strategy is adversarial learning, which involves training a model on adversarial examples in addition to regular training data [10]. Other security strategies include gradient masking, which involves hiding gradients from an attacker [11], and feature compression, which involves preprocessing input data to remove redundant features [18].

Several recent studies have also explored the use of generative models such as Generative Adversarial Networks (GANs) to generate adversarial examples [14]. These models can be used to create more realistic examples of competitive actions that are harder to detect and more difficult to defend against.

Biometric authentication systems, which use physiological or behavioral characteristics to verify people's identities, have become increasingly popular in recent years. However, these systems are not immune from attacks, and neural networks are used to carry out attacks on biometric data [18].

One common type of attack on biometrics is presentation attacks, also known as spoofing attacks, where an attacker uses a fake or artificial biometric to impersonate a legitimate user. Neural networks have been used to create attacks with a realistic representation of various biometric modalities, including fingerprints, face recognition, and voice recognition.

For example, Wang et al. [19] used a convolutional neural network (CNN) to generate realistic fingerprints that can be used to fake fingerprint recognition systems. Similarly, Nguyen et al. [20] used a generative adversarial network (GAN) to generate synthetic facial images that can be used to fake facial recognition systems.

Other research has focused on using neural networks to launch attacks on biometric data by exploiting vulnerabilities in the biometric system. For example, Li et al. [21] proposed a method for generating adversarial examples for fingerprint recognition systems by distorting the input fingerprint image using a gradient-based optimization method. The resulting fingerprint of the attacker can be used to avoid detection by the biometric system.

To protect against attacks on biometric data, researchers have proposed various defense strategies, including the use of liveness detection techniques to detect attacks on presentations and the use of deep neural networks to increase the resilience of biometric systems to attacks. For example, Tan et al. [22] proposed a deep neural network liveness detection method for detecting presentational attacks in face recognition systems.

3. Materials and research methods

Using the MNIST dataset as an example, we can consider the principle of building a certain neural network, and then explore its vulnerability. Based on the research of V.A. Ivanyuk, any neural network is mathematically a superposition of regression functions that describe the relationship between the values of the inputs and outputs of the network [2].

The input layer receives data in the form of features or input variables, which are then passed to the first hidden layer. Each node in the hidden layer performs a linear transformation of the input using weights and biases and then applies a non-linear activation function to produce a non-linear output. This output is then fed as input to the next level, and the process is repeated until the final output is obtained.

Starting the study, it is worth saying that the basic element of such systems is the so-called neuron. It is necessary to create a programming model. A neuron in an ANN is an artificial analogue of a real neuron, only represented as a simple mathematical function that determines the rules for

generating an output signal based on input data [2]. In the work of K.S. Kachagina stated that a neuron is an imaginary black object with several input and one output hole [3].

A neural network can be represented mathematically as a function $f(x; \theta)$, where x is the input, θ are the network parameters (weights and biases), and $f(x; \theta)$ is the output of the network.

The weights and biases in the neural network are adjusted during training to optimize network performance. This is done by minimizing the cost function, which measures the difference between projected output and actual output. The backpropagation algorithm is used to update the weights and biases in such a way as to reduce the loss function.

A neural network consists of layers of interconnected nodes, and the calculation of the output of each node can be represented mathematically as (1):

$$z = w \cdot x + b \quad (1)$$

where z – weighted sum of input parameters x , w – scale vector, b – displacement vector.

The output of a node is calculated by applying a non-linear activation function to a weighted sum, which can be represented mathematically according to (2):

$$a = g(z) \quad (2)$$

where $g(x)$ is the activation function.

The node layer output is calculated as follows (3):

$$a^1 = g(z^1) \quad (3)$$

where a^1 – exit of the first layer, z^1 – weighted sum of first layer inputs, g is the activation function.

The output of the last layer is the output of the neural network, which can be used for prediction or classification.

During training, the weights and biases of the neural network are adjusted to minimize the cost function, which measures the difference between the predicted output and the actual output. This is done using gradient descent, where the gradient of the cost function with respect to weights and biases is computed, and the weights and biases are updated accordingly.

The backpropagation algorithm is used to efficiently compute the gradient of the cost function with respect to the weights and biases of the network.

Thus, the mathematical logic of a neural network includes calculating the weighted sum of the input data for each node, applying a non-linear activation function, and propagating the output through the layers of the network to obtain the final output. The network weights and biases are optimized during training to minimize the cost function using gradient descent and backpropagation.

It is worth deciding on the neural network training algorithm:

- 1) Import libraries
- 2) Data checking
 - a) Checking for Lost Values
 - b) Data normalization
- 3) Modeling
- 4) Getting a result

The neural network analyzes the biometric data by learning patterns and features from the input data during the training process. It can then use these learned features to predict new biometrics. For example, a facial recognition neural network can learn to recognize certain facial features and patterns, such as eye position or mouth shape, and use that information to identify people in new images. Similarly, a fingerprint recognition neural network can learn to recognize the unique ridges and patterns on a fingerprint and use that information to verify a person's identity.

Let's analyze this algorithm on a specific example (Fig. 1).

It is necessary to create and train a neural network based on the MNIST database that will recognize handwritten numbers.

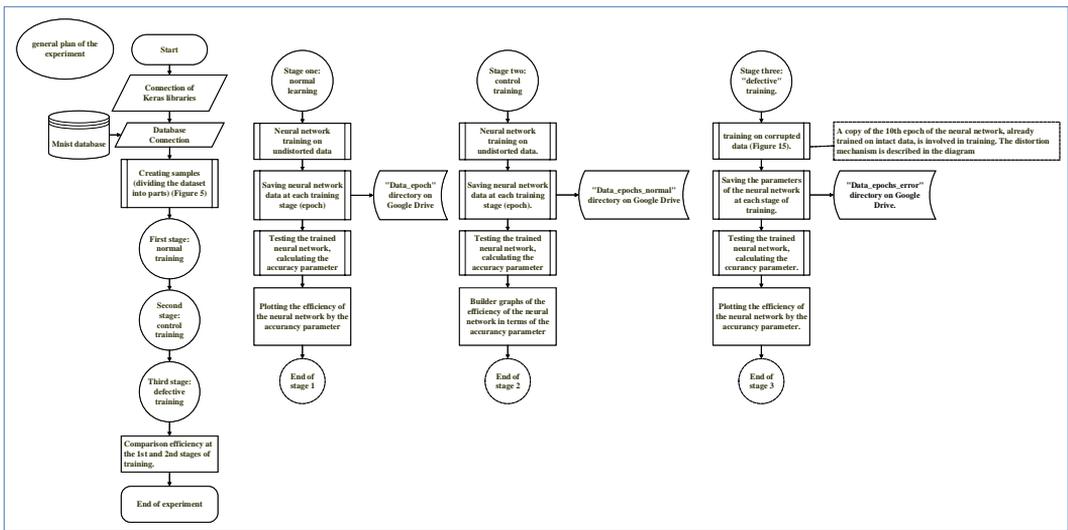


Fig 1. General structure of the experiment algorithm

It is necessary to create and train a neural network based on the MNIST database that will distinguish between handwritten numbers.

The following libraries and modules have been used:

Numpy is a library for working with multidimensional arrays and matrices [23].

Keras is a framework for building and training neural networks. It is part of the Tensorflow library and allows you to create neural network models using a set of high-level abstractions.

Import the NumPy library and give it the alias "np". We import the MNIST dataset from the Keras library. MNIST contains images of numbers from 0 to 9, which will be used to train the neural network [25].

Sequential is a neural network model in which layers are added sequentially one after the other. Dense and Flatten layers from the Keras library. Adam is an optimizer from the Keras library. Optimizers are used to adjust the neural network weights during training. Sparse Categorical Crossentropy is a loss function from the Keras library. The loss function is a metric that evaluates how well a neural network performs on a classification task.

Matplotlib is used for data visualization. ModelCheckpoint is a class that allows you to save model weights during training. The load_model function from the Keras library is used to load a saved model from a file [25].

Next, we connect Google Drive to Google Colab. Google Drive is used to save the model and other files.

The shuttle module in Python provides a high-level interface for working with files and directories. It contains functions for copying, moving, renaming, and deleting files and directories.

```
import numpy as np
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import SparseCategoricalCrossentropy
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import ModelCheckpoint
from keras.models import load_model
from google.colab import drive
drive.mount('/content/gdrive')
import shutil
```


For clarity, we derive one of the elements x_{train} and y_{train} in the form in which they are stored in tuples. Take element number 277 (Fig. 5).

Display the image of the element $x_{train}[277]$ in black and white (white is 0 and black is 255) (Fig. 4).



Fig 5. Image of element 277

Let's normalize the pixel values to bring the values from 0 to 255 to the range from 0 to 1. This is done by dividing each pixel value by 255. Pixel normalization improves the performance of the model, as it facilitates training and reduces the time required for processing data.

At the next stage, we split the training and test datasets into two equal parts. The first part will be used for the first training of the neural network, and the second part will be distorted and used in the second training. Define the architecture of the model (See Fig. 6).

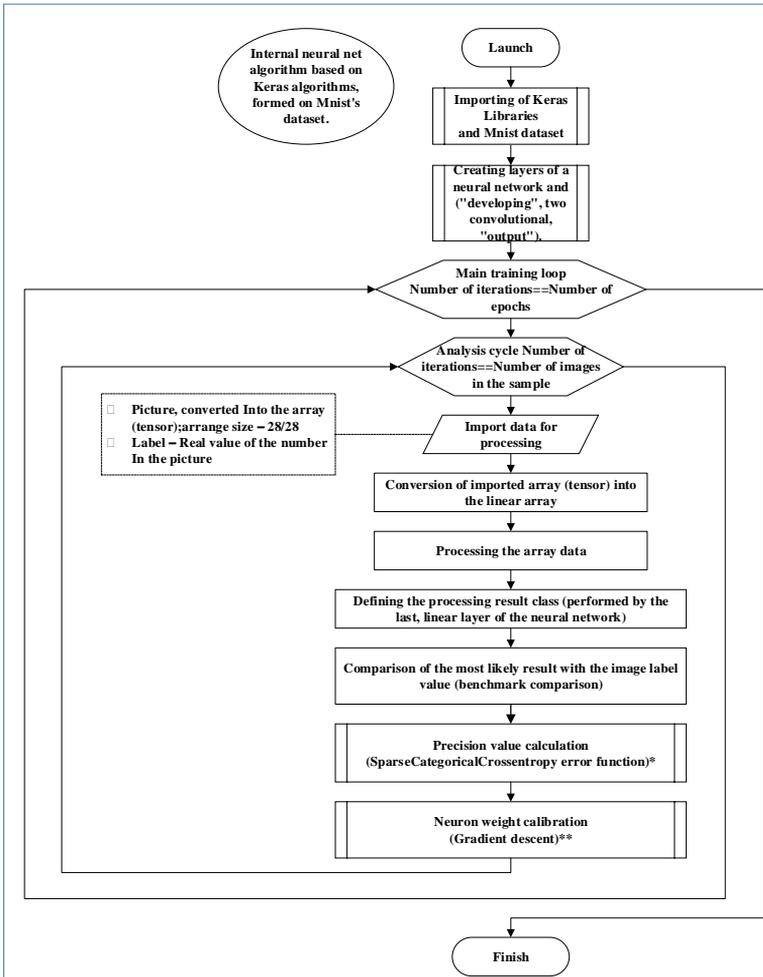


Fig 6. The structure of the algorithm of the internal mechanism of the neural network

Using the method of experiment, the optimal structure of the neural network and the parameters of its training were selected.

The first layer – Flatten converts a two-dimensional array (28, 28) into a one-dimensional array (dimension 784) so that it can be fed to the input of the neural network.

The second layer – Dense with 100 neurons and the ReLU (Rectified Linear Unit) activation function uses a linear operation followed by a non-linear activation function. The ReLU activation function returns 0 for negative values and the value itself for positive ones.

The third layer is Dense with 50 neurons and the ReLU activation function.

The first optimizer parameter defines the optimization method that will be used to train the model.

In this case, the optimizer Adam is used at the rate of learning (`learning_rate`) equal to 0.01.

Second parameter `loss` defines the loss function to be used during model training. Here we use categorical cross entropy (`SparseCategoricalCrossentropy`).

The fourth layer is Dense with 10 neurons and softmax activation function. softmax converts neuron values into probabilities summing up to 1.0 and is used for multi-class classification [29].

We compile the model with the necessary parameters.

Third parameter `metrics` defines the metrics that will be used to evaluate the model. In this case, we will use only the accuracy metric (`accuracy`).

```
model.compile(optimizer=Adam(learning_rate=0.01),  
              loss=SparseCategoricalCrossentropy(),  
              metrics=['accuracy'])
```

Create a checkpoint object that is used to save the state of the model as a file in a folder `data_epochs` after each learning epoch.

```
checkpoint = ModelCheckpoint('/content/gdrive/My Drive/neyro/data_epochs/  
epoch_{epoch:02d}.h5')
```

Train the model on training data `Xtrue_train` with appropriate labels `Ytrue_train`. Specify the number of epochs 10. For one training iteration, we take the batch size 100 (Fig. 7).

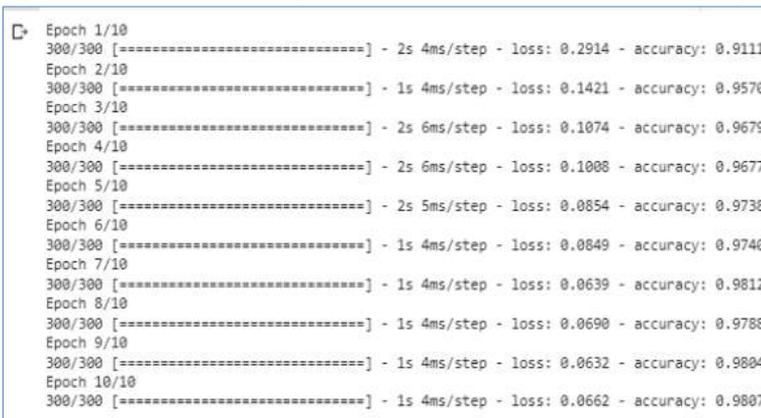


Fig 7. Data output after the first training

`shuffle=True` specifies that the training dataset will be shuffled before each epoch to avoid the possibility that the model might remember the order of the training examples.

`callbacks=[checkpoint]` indicates that the object `checkpoint` will be used as a callback to save the state of the model after each epoch.

Learning outcomes are saved to an object `history`. After training the model, `history` will contain information about the change in the loss function and accuracy metrics during model training.

```
epochs = 10
```

```
history = model.fit(Xtrue_train, Ytrue_train, epochs=epochs,  
                    batch_size=100, shuffle=True, callbacks=[checkpoint])
```

Let's evaluate the model on test data. After the first training, we got an accuracy of 0.9542 on test data.

```
test_loss, test_acc = model.evaluate(Xtrue_test, Ytrue_test)  
print('Test accuracy:', test_acc)  
  
157/157 [=====] - 0s 2ms/step - loss: 0.2230 - accuracy: 0.9542  
Test accuracy: 0.954200029373169
```

Fig 8. Accuracy of training on test data

Derive the graphs of the first training.

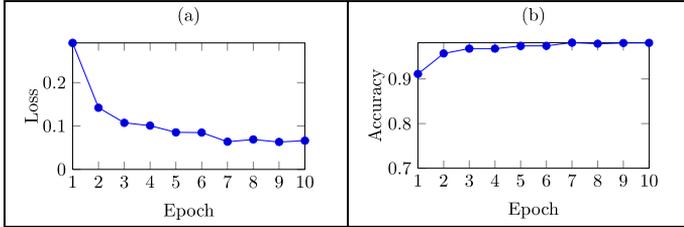


Fig 9. a) Graph of losses, b) Graph of accuracy

Thus, the neural network was trained to recognize handwritten numbers on the MNIST dataset.

The next stage of work consists in distorting the dataset, that is, simulating the intervention of an attacker, and then analyzing the results after retraining the neural network.

The task is to find a way to calculate malicious interference in the operation of the neural network.

Let's put forward a hypothesis that one of the signs of interference in the neural network may be the difference in performance and accuracy to the trained neural network from the original one.

Let us test this hypothesis on a practical problem.

Before training the model on a distorted dataset, we will make a copy of the 10th epoch file and train the model loaded from it in the same way, but on the second half of the dataset (Xerror_train и Yerror_train), but without changing anything in it.

```
source_file = '/content/gdrive/My Drive/neyro/data_epochs/epoch_10.h5'  
destination_file = '/content/gdrive/My Drive/neyro/data_epochs/  
                    epoch_10_copy1.h5'  
  
# Copying a file to a new path  
shutil.copyfile(source_file, destination_file)  
model = load_model('/content/gdrive/My Drive/neyro/data_epochs/  
                    epoch_10_copy1.h5')  
  
#Model compilation with Adam optimizer,  
SparseCategoricalCrossentropy loss function and accuracy metric.  
history = model.compile(optimizer=Adam(learning_rate=0.01),  
                        loss=SparseCategoricalCrossentropy(),  
                        metrics=['accuracy'])  
  
# Creating a folder to save training data after each epoch  
import os  
if not os.path.exists('/content/gdrive/My Drive/neyro/  
                        data_epochs_continue'):  
    os.makedirs('/content/gdrive/My Drive/neyro/data_epochs_continue')  
# Saving the model after each epoch to folder data_epochs_error  
checkpoint = ModelCheckpoint('/content/gdrive/My Drive/  
                             neyro/data_epochs_continue/epoch_{epoch:02d}.h5')  
# Training the model on the second half (not yet distorted) of the  
dataset  
epochs = 10  
history = model.fit(Xerror_train, Yerror_train, epochs=epochs,  
                    batch_size=100, shuffle=True, callbacks=[checkpoint])
```

```

Epoch 1/10
300/300 [=====] - 3s 6ms/step - loss: 0.1726 - accuracy: 0.9577
Epoch 2/10
300/300 [=====] - 2s 6ms/step - loss: 0.0972 - accuracy: 0.9724
Epoch 3/10
300/300 [=====] - 2s 6ms/step - loss: 0.0901 - accuracy: 0.9740
Epoch 4/10
300/300 [=====] - 2s 6ms/step - loss: 0.0867 - accuracy: 0.9758
Epoch 5/10
300/300 [=====] - 2s 8ms/step - loss: 0.0787 - accuracy: 0.9773
Epoch 6/10
300/300 [=====] - 2s 8ms/step - loss: 0.0646 - accuracy: 0.9807
Epoch 7/10
300/300 [=====] - 2s 6ms/step - loss: 0.0633 - accuracy: 0.9824
Epoch 8/10
300/300 [=====] - 2s 6ms/step - loss: 0.0571 - accuracy: 0.9832
Epoch 9/10
300/300 [=====] - 2s 6ms/step - loss: 0.0468 - accuracy: 0.9857
Epoch 10/10
300/300 [=====] - 2s 7ms/step - loss: 0.0622 - accuracy: 0.9829
    
```

Fig 10. Output of training results

Accuracy after retraining increased from 0.9542 to 0.9762 (Fig. 10).

Let's create a copy of the epoch 10 file so that it can be used in the second training of the neural network and compare the received data.

Let's compile the model using the same parameters as the first time.

```

history = model.compile(optimizer=Adam(learning_rate=0.01),
                        loss=SparseCategoricalCrossentropy(),
                        metrics=['accuracy'])
    
```

Training and testing of the ANN is carried out on the second half of the MNIST dataset (with the prefix error), which was created at the beginning of the code.

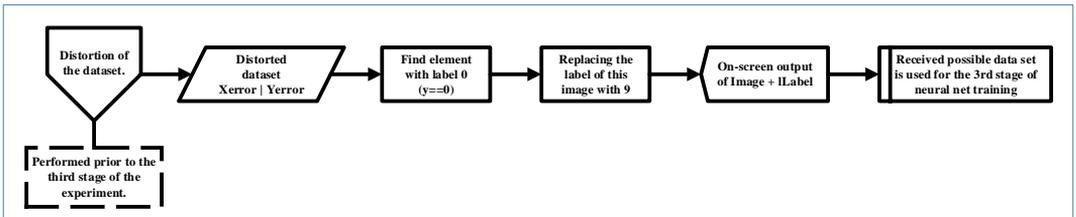


Fig 11. The structure of the algorithm for direct distortion of the sample "Error"

Let's find an element from the data set whose value is 0 and display its image and its corresponding label. `plt.imshow(Xerror_train[9], cmap='binary')` (Fig. 12).

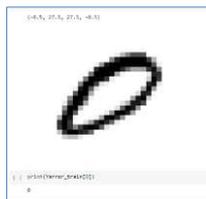


Fig 12. Image of dataset element 0 and its label

Replace all label values in the training dataset `Yerror_train` with values 0 to 9 (Fig. 11).

This stage is an imitation of an attacker's action. This example can be projected onto real face recognition neural networks, that is, different dataset numbers are different faces of people.

One of the reasons for the emergence of the Adversarial attack is that machine learning methods were originally developed for stationary and secure environments, where the training and test samples were generated from the same statistical distribution. However, in practice, attackers can covertly manipulate input data to exploit vulnerabilities in ML algorithms and compromise the security of the entire machine learning system.

The type of attack in which an attacker seeks to gain access to the data and the learning process of an ML model to “poison” it (train it incorrectly) for subsequent inadequate work is called poisoning. It can be seen as a malicious infection of the training data. Thus, the “white box” strategy is used here, when the attacker has information about the victim – “malicious knowledge” (Adversarial Knowledge, AK): how the data for training are prepared and from what sources and what they are, what are the main functions of the attacked system, what algorithms it uses, what are the results, etc. Poison attacks involve insider information about the ML system and a high level of attacker’s competence in Data Science [23].

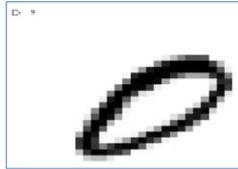


Fig 13. Image of dataset element 0 and its label after replacement

As a result, all images 0 correspond to labels with a value of 9 (Fig. 13).

Retrained the neural network on data sets `Xerror_train` and `Yerror_train`, using the same parameters as the first time.

`epochs = 10`

`history = model.fit(Xerror_train, Yerror_train, epochs=epochs, batch_size=100, shuffle=True, callbacks=[checkpoint])`

```
Epoch 1/10
300/300 [=====] - 2s 4ms/step - loss: 0.2091 - accuracy: 0.9531
Epoch 2/10
300/300 [=====] - 1s 4ms/step - loss: 0.1028 - accuracy: 0.9696
Epoch 3/10
300/300 [=====] - 1s 4ms/step - loss: 0.0860 - accuracy: 0.9740
Epoch 4/10
300/300 [=====] - 1s 4ms/step - loss: 0.0765 - accuracy: 0.9768
Epoch 5/10
300/300 [=====] - 1s 5ms/step - loss: 0.0734 - accuracy: 0.9779
Epoch 6/10
300/300 [=====] - 1s 4ms/step - loss: 0.0674 - accuracy: 0.9806
Epoch 7/10
300/300 [=====] - 1s 5ms/step - loss: 0.0600 - accuracy: 0.9815
Epoch 8/10
300/300 [=====] - 2s 6ms/step - loss: 0.0592 - accuracy: 0.9829
Epoch 9/10
300/300 [=====] - 2s 7ms/step - loss: 0.0486 - accuracy: 0.9858
Epoch 10/10
300/300 [=====] - 1s 4ms/step - loss: 0.0531 - accuracy: 0.9849
```

Fig 14. Data output after training

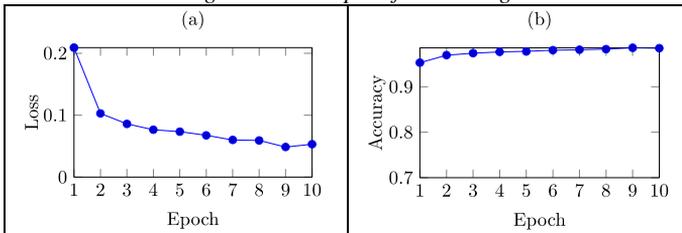


Fig 15. a) Graph of losses b) Graph of accuracy

We got an accuracy on the test data of 0.8758, which is much less than the result of 0.9762 obtained after the model was trained on an undistorted dataset, and less than after the first training, where the result on the test data was 0.9542 (Fig. 14, 15, Table 1).

Table. 1. Simulation results

	ANN training		
	First	Undistorted dataset	Distorted dataset
Accuracy	0,9542	0,9762	0,8758

This confirms the hypothesis. Thus, when retraining a neural network on a distorted dataset, the accuracy on test data drops, and when retraining on an undistorted dataset, it increases.

This is since the information received by the neural network about handwritten numbers from the MNIST dataset during initial training comes into conflict with the information received during retraining on the dataset with changed labels.

Therefore, a sharp decrease in the accuracy of the neural network during additional training is one of the signs of an attacker's intervention.

The presented method is one of the simplest methods for detecting malicious interference, and in further research it is planned to rely on the gradient descent method, which can be more accurate for this task. This is exactly what scientists from Cornell University did for the article "Interpreting Deep Neural Networks with SVCCA" [30].

4. Conclusion

In conclusion, adversarial attacks are of particular interest in the field of artificial intelligence, and a large amount of research has been done in this direction. While defense strategies have been proposed and the arms race between attackers and defenders continues, new methods of attack are constantly being developed. Thus, it is important for researchers to study adversarial attacks and develop new protection strategies to ensure the security of machine learning systems, the formation of a secure model of trusted artificial intelligence.

References / Список литературы

- [1] Marshalko G. Attacks on biometric systems. Information Security (in Russian) / Маршалко Г. Атаки на биометрические системы. Information Security. Available at: <https://www.itsec.ru/articles/ataka-na-biometricheskie-sistemy>, accessed 30.03.2023.
- [2] Ivanyuk V. Neural Networks and Their Analysis. Chronoeconomics, 2021, issue 4, pp. 58-61 / Иванюк В.А. Нейронные сети и их анализ. Хроноэкономика, вып. 4, 2021 г., стр. 58-61.
- [3] Kachagina K. S., Safarova A. D. Neron Networks - Development Prospects. E-Scio, issue 2, 2021, 10 p. (in Russian) / Качагина К.С., Сафарова А.Д. Нейронные сети - перспективы развития. E-Scio, вып. 2, 2021 г., 10 стр.
- [4] Akhtar Z., Foresti G. L. Face spoof attack recognition using discriminative image patches. Journal of Electrical and Computer Engineering, 2016, article id 4721849, 15 p.
- [5] Chernobrov A. How to cheat a neural network or what is an Adversarial attack (in Russian) / Чернобров А. Как обмануть нейросеть или что такое Adversarial attack. Available at: <https://chernobrovov.ru/articles/kak-obmanut-nejroset-ili-cto-takoe-adversarial-attack.html>, accessed: 02.04.2023.
- [6] Goodfellow I.J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014, 11 p.
- [7] Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.2016, 14 p.
- [8] Wiyatno R., Xu A. Maximal Jacobian-based Saliency Map Attack. arXiv preprint arXiv:1808.07945, 2018, 5 p
- [9] Moosavi-Dezfool S.-M., Fawzi A., Frossard P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574-2582
- [10] Madry A., Makelov A. et al. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017, 28 p.
- [11] Papernot N., McDaniel P. et al. The limitations of deep learning in adversarial settings. In Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P), 2016, pp. 372-387.
- [12] Carlini N., Wagner D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proc. of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 3-14.
- [13] Akhtar N., Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, vol. 6, 2018, pp. 14410-14430.
- [14] Xiao C., Deng R. et al. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. Lecture Notes in Computer Science, vol. 11214, 2018, pp. 220-237.

- [15] Xie C., Wang J. et al. Adversarial examples for semantic segmentation and object detection. In Proc. of the IEEE International Conference on Computer Vision, 2017, pp. 1369-1378.
- [16] Samangouei P., Kabkab M., Chellappa R. Defense-gan: protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1795.06605, 2018, 18 p.
- [17] Xu H., Ma Y. et al. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing*, vol. 17, issue 2, 2020, pp. 151-178.
- [18] Xu W., Evans D., Qi Y. Feature squeezing: Detecting adversarial examples in deep neural networks. In Proc. of the Network and Distributed Systems Security (NDSS) Symposium, 2018, 15 p
- [19] Wang Y., Kang L. et al. Fingerprint presentation attack detection using convolutional neural network with transfer learning. *IEEE Access*, vol. 7, 2019, pp. 131443-131451.
- [20] Nguyen T.M., Kim K.H. et al. Generative adversarial network-based face presentation attack detection using partial convolution and multi-domain learning. *IEEE Transactions on Information Forensics and Security*, vol. 14, issue 10, 2019, pp. 2764-2779.
- [21] Li X., Chen T., Yang J. Adversarial fingerprint attacks and defenses. *IEEE Transactions on Information Forensics and Security*, vol. 14, issue 1, 2019, pp. 66-80.
- [22] Tan H., Li H., et al. Deep learning based liveness detection: A survey. *ACM Computing Surveys*, vol. 52, issue 3, (2019), pp. 1-27.
- [23] The official website of the NumPy Library. Available at: Available: <https://numpy.org>, accessed 30.03.2023.
- [24] Examples of neural networks' implementation (in Russian) / Примеры реализации нейронных сетей. Available at: Available: <https://webtort.ru>, accessed 30.03.2023.

Information about authors / Информация об авторах

Denis Vladimirovich KOTLYAROV is a student. His research interests include artificial neural networks, cryptography.

Денис Владимирович КОТЛЯРОВ – студент. Его научные интересы включают искусственные нейронные сети, криптография.

Gleb Dmitrievich DYUDYUN – student. His research interests include artificial neural networks, cryptography.

Глеб Дмитриевич ДЮДЮН – студент. Его научные интересы включают искусственные нейронные сети, криптография.

Natalya Vitalievna RZHEVSKAYA is a student. Her research interests include artificial neural networks, cryptography.

Наталья Витальевна РЖЕВСКАЯ – студентка. Ее научные интересы включают искусственные нейронные сети, криптография.

Maria Anatolyevna LAPINA – Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Information Security of Automated Systems. Her research interests include artificial neural networks, cryptography, numerical methods, differential calculus.

Мария Анатольевна ЛАПИНА – кандидат физико-математических наук, доцент кафедры информационной безопасности автоматизированных систем. Ее научные интересы включают искусственные нейронные сети, криптография, численные методы, дифференциальные исчисления.

Mikhail BABENKO – Doctor of Physical and Mathematical Sciences, Head of the Department of Computational Mathematics and Cybernetics. His research interests include artificial neural networks, cryptography, homomorphic encryption, secret sharing schemes, modular arithmetic, cloud computing.

Михаил Григорьевич БАБЕНКО – доктор физико-математических наук, заведующий кафедрой вычислительной математики и кибернетики. Его научные интересы включают искусственные нейронные сети, криптография, гомоморфное шифрования, схемы разделения секрета, модулярная арифметика, облачные вычисления.

DOI: 10.15514/ISPRAS-2023-35(2)-4



Development of Legal Document Classification System Based on Support Vector Machine

*Iu. Nasu, ORCID: 0009-0005-3852-0022 <yunas2002@iCloud.com>
V.V. Lanin, ORCID: 0000-0002-0650-2314 <vlanin@hse.ru>*

*HSE University,
38, Studencheskaya street, Perm, Perm Krai, 614070, Russia*

Abstract. This paper was prepared while developing text classification system for legal documents, especially those that issued by Legislative Assembly of Perm Krai. The problem in question is a lack of solutions that meet regional requirements, the main of which is the classification used in region. The research that evaluates applications of Natural Language Processing models is conveyed. The primary result of the study is the actual applicability of Support Vector Machine (SVM) to preprocessed legal document categorization. There were a server-side API constructed to perform the task, and a server-side models pre-trained of which SVM is favored.

Keywords: FastAPI; Legal Tech; Supervised Learning; Support Vector Machine

For citation: Nasu Iu., Lanin V.V. Development of Legal Document Classification System Based on Support Vector Machine. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 49-56. DOI: 10.15514/ISPRAS-2023-35(2)-xx

Разработка системы классификации нормативно-правовых документов на основе метода опорных векторов

*Ю. Насу, ORCID: 0009-0005-3852-0022 <yunas2002@iCloud.com>
В.В. Ланин, ORCID: 0000-0002-0650-2314 <vlanin@hse.ru>*

*Национальный исследовательский университет «Высшая школа экономики»,
614070, Россия, Пермский край, г. Пермь, ул. Студенческая, 38*

Аннотация. Статья подготовлена в ходе разработки системы классификации текстов юридических документов, особенно тех, которые издаются Законодательным собранием Пермского края. Рассматриваемая проблема заключается в отсутствии решений, отвечающих региональным требованиям, основным из которых является применяемая в регионе классификация. Представлено исследование, оценивающее применение моделей обработки естественного языка. Основным результатом исследования является фактическая применимость машины опорных векторов (SVM) к предварительно обработанной категоризации юридических документов. Для выполнения этой задачи был создан API на стороне сервера, а также модели на стороне сервера, предварительно обученные тому, какой SVM предпочтительнее.

Ключевые слова: FastAPI; юридические информационно-аналитические системы; контролируемое обучение; метод опорных векторов

Для цитирования: Насу Ю., Ланин В.В. Разработка системы классификации нормативно-правовых документов на основе метода опорных векторов. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 49-56. DOI: 10.15514/ISPRAS-2023-35(2)-4

1. Introduction

The number of normative legal acts in force only at the federal level in Russia is just below 10 000[1]. That quantity of documents is clearly impossible for any lawyer to memorize or access swiftly, not taking into account other professions who also base their work on legislation (e.g., bookkeepers, accountants, engineers, etc.).

The changeability of Law in this situation only aggravates it. New Legal Acts enter into force every month. The volume of the legislative text increases, hence may be a factor that hinders its perception and, thus, execution. Suffice it to point that the rapid tracking of novel legal acts has become possible only due to the development of the Internet.

A partial solution to the problem in question can be a construction of an IT system for classification of legal acts by categories (i.e., classes) and its further implementation to Legal Knowledge Bases. This approach will certainly not reduce the number of legal documents (this requires cultural measures, not technological ones), but it will facilitate the work with legal information.

The system designed and this paper focuses mainly on solving the problem of classification of legal acts. Thus, the process of classification of legal acts, now carried out by experts, is automated.

2. State of Art

Legal Tech systems might be divided in two categories: the ones which automate or facilitate routine organizational business process within legal firm, departments and even courts (e.g., online dispute resolution, chatbots, online consulting, etc.) and ones which work with legal data and automate different legal processes (e.g., automated document analysis, document builders, legal libraries, etc.).

This work will focus on the second type of systems. Several of them should be noted before future exploration. There are *Consultant Plus* assistant system, *Garant* and the *Ministry of Justice*' web portal. They allow user to search Legal Acts, Court Decisions. *Consultant* and *Garant* also provide document builders and several other services such as calculators, online consultations, etc. Each of those systems has its variety of the paid Professional Versions.

The only reason those solutions could not be used is the customer's requirement to use **regional classification** (see Appendix 1) for processing.

3. The Task

The main purpose of the system is its application for building catalogues of documents based on their topic/topics. It will make search easier for legislators and lawyers.

Hence, the task in question is to build document classification system that could help to organize files. The automized process could be described as mapping document (its text) to a class. Thus, the system takes in a file and yields a string name of a class.

The mathematical definition of the task is finding an approximation of the function that maps the set of documents to the set of categories. Each of the documents is a set of lexemes, thus the task could be described as finding a function that maps sets of sets of lexemes to set of classes.

As the words may have different semantic weight, those that contain less meaning (so called stop-words) might be omitted during the analysis enhancing the speed of processing.

4. Requirments

The vital part of information system development is gathering customer's requirement. Thus, the chances to meet their needs increases. During the interview with the customer the list of requirements and their proper definitions of done were formulated. They are listed in Table 1.

Table 1. Requirements

Requirement	Definition of Done
The System is trustworthy	The System's accuracy metric is no less than 80% on both Orders and Laws
The System accepts 'docx' and 'docm' files	The System takes in 'docx' and 'docm' file and yields correct result. On attempt to put another file the System returns an error message and continues the work.
On damaged document returns respective message	On damaged document a Server returns respective message
Client-Server Architecture	Client and Server is separated
The System is easy to understand	The System returns the code and the name of the category
The system is easy to modify	There must a brief documentation

5. Technologies

As is known text classification is a supervised learning task. Hence, we need a test sample of already classified texts, which was kindly provided by the customer.

Considering the most efficient stack of technologies, we should consider the predominant use of Python programming language for solving that kind of tasks (first of all, due to the quantity of libraries written for this language). The libraries often used are *pandas*, *matplotlib* (for data analysis), *word2vec* and *sklearn* libraries (for machine learning models). Use of any other language would lead to solving many already decided problems (from preprocessing to building the logic of the classification model itself) from scratch.

Now that the choice of the main programming language is resolved, the Python Web framework must be chosen. Using FastApi is in the context of this context proves as the most effective solution, primarily because of its speed [2].

6. Exploring Data

As is known, Legal Documents have the structure which could be used to identify and parse them. Documents usually contain a Title, an Issue Number, Dates of Registration and Adoption (that could easily be confused while parsing), an Authority that signs the document. It should be noted that several documents may have the very same Title or Number. Thus, it requires several parameters to identify the document.

In our research, we focus on two types of Legal Acts: Laws and Orders. They severely differ in their structure. Laws are usually voluminous documents which have much less references to another laws (usually 1-2) per page than Orders, which are based on Laws and often are one-page long. Classification system may malfunction on such a little amount of unique data as present in Orders. It is worth to use references to already processed documents to classify them properly (especially, if those Orders are 0.5-1 page long).

The other problem which is met is the data provided. The regional document collection, which is 3000 items long, is obviously unbalanced. The distribution of documents over the classes is shown on Fig. 1.

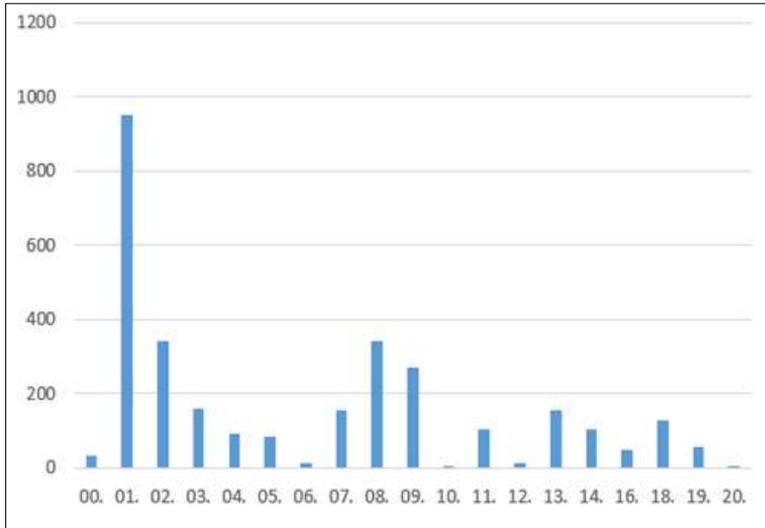


Fig. 1. Distribution of Documents

Classes 06. “Labor and Employment”, 10. “Interregional Economic Affairs”, 12. “Information and Informatization”, 20. “International Affairs” are represented by less than twenty documents each (they may also contain subclasses within them). To solve this issue, two solutions might be utilized:

- 1) Adding Legal Documents from other regions (or the ones accepted by the Federation);
- 2) Using models which sustain the imbalance.

This research focuses mainly on the second way, but it proves that the first is efficient as well.

7. Classification model

There were three classification models used [3]:

- 3) Naive Bayes Classifier,
- 4) Support vector machine,
- 5) Logistic regression.

Tokenization, stop-words removal, and lemmatization had been performed before passing the training data to TF-IDF vectorizer. Then the data was passed to classifiers.

To select a model, each of them was built and trained on the very same sets. Thus, we might choose the best one. The metrics shall be listed in Table 2.

Table 2. Models and Metrics

	Naive Bayes	SVM	Logistic Classification
Accuracy. 600 items	0.17	0.48	0.23
F1-Score ¹ 600 items	0.18	0.50	0.25
Accuracy. 750 items	0.3	0.6	0.37
F1-Score 750 items	0.3	0.61	0.39
Accuracy. 900 items	0.34	0.7	0.6

¹ Macro-averaged.

	Naive Bayes	SVM	Logistic Classification
F1-Score 900 items	0.36	0.71	0.66
Accuracy. 1500 items	-	0.76	-
F1-Score 1500 items	-	0.78	-

As is seen, Support Vector Machine proves to be more accurate than its counterparts (due to the ability to describe even the smallest classes), therefore it is chosen to be used in production. The metrics listed prove that quality might be improved via using the vaster dataset. The only drawback is difficulty of interpretation (find more detailed works on this topic in [4-10]).

To increase F1-score there were 600 new documents introduced (achieving the total quantity of 1500 items). Those items were collected to represent the classes that contained less than 100 training documents (Fig. 1).

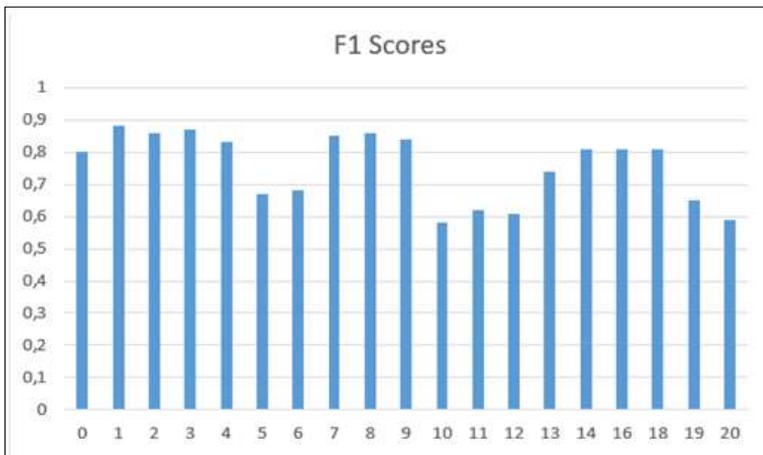


Fig. 2. F1 measures for each class (1500 items)

8. Ways to Improve

As is said the main problem is unbalanced data which worsens results (fig. 2) as the least represented classes usually yield the worst result. Thus, focusing on less presented classes is the way to improve the whole picture. Thus, by adding training documents of deficient classes helps to improve classifier (see last two rows of Table 2).

The very first addition made is filtering non-Legal Documents. It could be done two ways: adding a filtering model (Legal/Non-Legal Document Classifier), using the structure of the document that is provided by the user.

Those two ways could be combined in different ways: structure-oriented and content-oriented. The first approach mainly focuses on similarity between the document structures (i.e., document provided by the user is compared with “typical” document which we use to measure incoming), the second evaluates the content inside the document.

Let us compare the advantages and disadvantages of both these approaches. The structure-oriented approach is easier to perform, there is little if any machine learning required. Its main flaw is a subsequence of simplicity – the “typical” structures of Legal Acts allow little flexibility. Thus, it would not pass Legal Documents that does not follow the structure but those Non-Legal Documents that do.

The second approach is a little more complex. It requires to gather different kinds of Non-Legal Documents and build a binary classifier. Suffice it to say the classifier should be adjusted to detect all the Legal Documents.

Thus, the model should bolster recall over precision due to false positives (i.e., Non-Legal Documents passed) be not such a problem as inability to classify Legal Document. Nevertheless, the “force-check” option could be provided to user, allowing to check any documents they wish.

9. Web API

The Web API in question was made using FastAPI Python Framework. The system support client-server architecture (fig. 3).

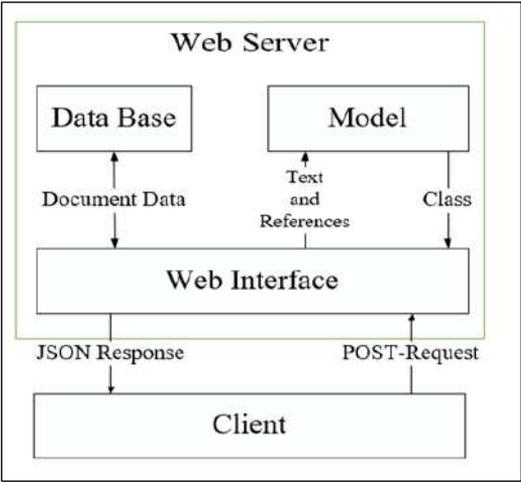


Fig. 3. Scheme of Architecture

It follows the REST pattern by saving all the necessary information to Microsoft SQL Server Database, which now contains four data tables.

The one called “Document” contains fields for its name and for date it is received.

The “Text” table encapsulates texts present in the incoming documents. Those texts could be used to retrain the model.

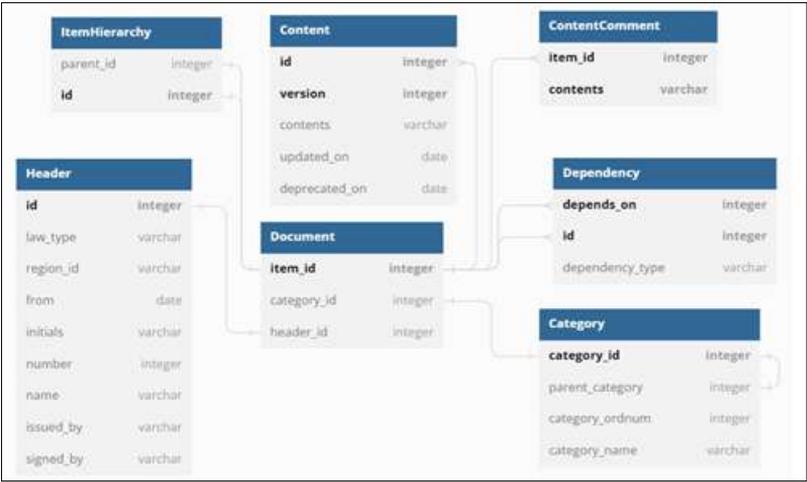


Fig. 4. Data Base Scheme

Along with those tables there are “Document Classifier” (which contains all the possible classes) and “Classified” (that saves the result of processing) tables.

The database is in the Third Normal Form since tables contain only complete primary key and lack non-trivial dependencies (fig. 4).

The API has three endpoints:

- /classify,
- /help;
- /.

The first of them is used to perform classification. It utilizes POST-request method and requires a “.docx” or “.docm” files. It returns JSON response (1):

(1) {“id”: <class id>, “name”: <class name>}

The other two endpoints use GET-request method and provide information about the system. Thus, “/help” method returns the regional classifier and “/” returns information about the maker and the resources on which the API is tested.

10. Future Updates

As a result, the only requirement that is not fully fulfilled is the quality of prediction. Thus, future development will involve application of the deep leaning method and use of N-Grams to the task in question.

A table named “References” would be present in the next update, it will contain all the links to other Legal Acts present in the document. It will be a “many-to-many” table linking two IDs. This table will be a groundwork of a classification approach based on references to already classified documents.

By enlarging the training base there were better results produced. Thus, there will be more documents used for classification in next updates.

References

- [1] Search Results in Russian Legislation. Available at: <http://pravo.gov.ru/proxy/ips/?searchres=&bpas=cd00000&a3=102000505&a3type=1&a3value=%D4%E5%E4%E5%F0%E0%EB%FC%ED%FB%E9+%E7%E0%EA%EE%ED&a6=&a6type=1&a6value=&a15=&a15type=1&a15value=&a7type=1&a7from=&a7to=&a7date=&a8=&a8type=1&a1=&a0=&a16=&a16type=1&a16value=&a17=&a17type=1&a17value=&a4=102000037%3B102000038&a4type=1&a4value=&a23=&a23type=1&a23value=&textpres=&sort=7&x=65&y=10>, accessed 06 Apr. 2023.
- [2] Shaji A. Serving a Machine Learning Model with FastAPI and Streamlit. Available at: <https://testdriven.io/blog/fastapi-streamlit/>, accessed 01 Apr. 2023.
- [3] Li S. Multi-Class Text Classification Model Comparison and Selection. Available at: <https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568>, accessed on 01 Apr. 2023.
- [4] Joachims T. Transductive inference for text classification using support vector machines. In Proc. of the Sixteenth International Conference on Machine Learning, 1999, pp. 200-209.
- [5] Tong S., Koller D. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, vol. 2, 2001, pp. 45-66.
- [6] Fernandez-Delgado M., Cernadas E. et al. Do we need hundreds of classifiers to solve real world classification problems. Journal of Machine Learning Research, vol. 15, issue 1, 2014, pp. 3133-3181.
- [7] Leslie C.S., Eskin E., Noble W.S. The spectrum kernel: A string kernel for SVM protein classification. In Proc. of the Pacific Symposium on Biocomputing, 2002, pp. 566-575.
- [8] Eskin E., Weston J. et al. Mismatch string kernels for SVM protein classification. In Proc. of the 15th International Conference on Neural Information Processing Systems (NIPS), 2002, pp. 1441-1448.
- [9] McCallum A., Nigam K. A comparison of event models for naive bayes text classification. In Proc. of the AAAI-98 Workshop on Learning for Text Categorization, 1998, pp. 41-48.
- [10] Kim S.B., Han K.S. et al. Some effective techniques for naive bayes text classification. IEEE transactions on knowledge and data engineering, vol. 18, issue 11, 2006, pp. 1457-1466.

Appendix 1. Classifier

- 1) Constitutional system;
- 2) Basics of governing;
- 3) Civil law;
- 4) Family law;
- 5) Housing law;
- 6) Labor and employment;
- 7) Social welfare and insurance;
- 8) Finances;
- 9) Economic activities;
- 10) Interregional Economic Affairs;
- 11) Natural resources and environmental control;
- 12) Information and Informatization;
- 13) Education. Science. Culture;
- 14) Public health. Sports. Tourism;
- 15) —;
- 16) Public order and security;
- 17) —;
- 18) Justice;
- 19) Public prosecutor. Legal authorities. Advocacy;
- 20) International affairs;

Information about authors / Информация об авторах

Юри НАСУ – студент 3 курса направления «Программная инженерия» факультета социально-экономических и компьютерных наук НИУ ВШЭ, Пермь. Научные интересы: Модели анализа текстовых данных и их применение в гуманитарных областях.

Yuri NASU – 3rd year student of the direction «Software Engineering» of the Faculty of Socio-Economic and Computer Science, HSE University, Perm. Research interests: Text data analysis models and their application in the humanities.

Вячеслав Владимирович ЛАНИН – старший преподаватель кафедры информационных технологий в бизнесе факультета социально-экономических и компьютерных наук НИУ ВШЭ, Пермь. Научные интересы: промышленная разработка программного обеспечения, семантические технологии, технологии анализа данных в Интернете.

Vyacheslav Vladimirovich LANIN – Senior Lecturer, Department of Information Technologies in Business, Faculty of Socio-Economic and Computer Science, HSE University, Perm. Research interests: industrial software development, semantic technologies, Internet data analysis technologies.

DOI: 10.15514/ISPRAS-2023-35(2)-5



Ферма данных: информационная система сбора, хранения и обработки неструктурированных данных из разнородных источников

С.П. Левашкин, ORCID: 0000-0001-8687-2704 <s.levashkin@psuti.ru>

К.Н. Иванов, ORCID: 0000-0002-6248-9564 <k.ivanov@psuti.ru>

С.В. Кушуков, ORCID 0009-0004-0270-5853 <s.kushukov@psuti.ru>

*Поволжский государственный университет телекоммуникаций и информатики,
443086, Россия, г. Самара, Московское ш., 77*

Аннотация. Представлена оригинальная информационная система «ферма данных». Сегодня успешное применение алгоритмов искусственного интеллекта, прежде всего глубокого обучения на основе нейронных сетей, практически полностью зависит от наличия данных. И чем больше объем этих данных, тем лучше результаты работы алгоритмов. Хорошо известны примеры таких алгоритмов от Facebook, Google, Microsoft, Yandex и др. Данные должны содержать как обучающую выборку, так и тестируемую. Причем, данные должны быть хорошего качества и обладать определенной структурой, в идеале быть размеченными, чтобы алгоритмы обучения работали адекватно. Это представляет серьезную проблему, требующую огромных вычислительных и человеческих ресурсов. Именно решению этой проблемы посвящена данная статья. На сегодня ферма данных представляет из себя довольно сложную информационную систему, построенную по модульному принципу, схожую с известным конструктором «Лего». Отдельными модулями системы являются различные современные алгоритмы, технологии и целые библиотеки искусственного интеллекта, а все вместе они призваны автоматизировать процесс получения и структурирования качественных больших данных в различных предметных областях. Система была протестирована на данных по COVID-19 в регионах России и странах мира. Кроме того, был разработан удобный интерфейс визуализации данных, собранных и обработанных на ферме. Это дает возможность проводить наглядные численные эксперименты компьютерного моделирования и сравнивать их с реальными данными, превращая ферму в интеллектуальную информационную систему поддержки принятия решений.

Ключевые слова: интеллектуальная информационная система; ферма данных; большие данные; обработка данных; визуализация данных; компьютерное моделирование

Для цитирования: Левашкин С.П., Иванов К.Н., Кушуков С.В. Ферма данных: информационная система сбора, хранения и обработки неструктурированных данных из разнородных источников. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 57-72. DOI: 10.15514/ISPRAS–2023–35(2)–5

Благодарности: Ферма данных была разработана в рамках мегагранта Минобрнауки. Авторы благодарят коллег РФЯЦ – ВНИИТФ (Снежинск) и ИПМ им. М.В. Келдыша (Москва) за сотрудничество.

Data farm: Information system for collecting, storing and processing unstructured data from heterogeneous sources

S.P. Levashkin, ORCID: 0000-0001-8687-2704 <s.levashkin@psuti.ru>

K.I. Ivanov, ORCID: 0000-0002-6248-9564 <k.ivanov@psuti.ru>

S.V. Kushukov, ORCID: 0009-0004-0270-5853 <s.kushukov@psuti.ru>

*Povolzhskiy State University of Telecommunications and Informatics,
77, Moskovskoe sh., Samara, 443086, Russia*

Abstract. The original information system «data farm» is presented. Today, the successful application of artificial intelligence algorithms, primarily deep learning based on artificial neural networks, almost completely depends on the availability of data. And the larger the amount of these data (big data), the better are the results of the algorithms execution. There are well-known examples of such algorithms from Facebook, Google, Microsoft, Yandex, etc. The data must contain both the training sample and the test one. Moreover, the data must be of good quality and have a certain structure, ideally, be labeled in order for the learning algorithms to work adequately. This is a serious problem requiring huge computational and human resources. This paper is dedicated to solve this problem. Today data farm is a rather complex information system built on a modular basis, similar to the well-known Lego constructor. Separate modules of the system are various modern algorithms, technologies and entire libraries of artificial intelligence, and all together they are designed to automate the process of obtaining and structuring high-quality big data in various subject domains. The system has been tested on data of COVID-19 in regions of Russia and countries around the world. In addition, a user-friendly interface for visualizing collected and processed on the farm data was developed. This makes it possible to conduct visual numerical experiments of computer simulation and compare them with real data, turning the farm into an intelligent decision support information system.

Keywords: intelligent information system; data farm; big data; data processing; data visualization; computer modeling

For citation: Levashkin S.P., Ivanov K.I., Kushukov S.V. Data farm: Information system for collecting, storing and processing unstructured data from heterogeneous sources. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 57-72 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-5

Acknowledgements: The data farm was developed as part of a mega-grant from the Ministry of Science and Education. The authors are grateful to the colleagues of VNIITF of Rosatom (Snezhinsk) and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences (Moscow) for collaboration.

1. Введение

Автоматизированный сбор, хранение и структурирование разнородных данных является одной самых сложных проблем в так называемом «слабом искусственном интеллекте» (СИИ), то есть интеллекте, основанном на данных. Ее решение представляет из себя серьезную проблему, требующую огромных вычислительных и человеческих ресурсов. До сих пор для структурирования и разметки данных применяется ручной или интерактивный режим. Между тем, успешное применение алгоритмов машинного обучения, которые являются основными в СИИ, невозможно без процесса предварительной подготовки данных [1]. Этой задачей занимаются специалисты в науке о данных (data scientists) [2]. Цель, которую преследует представленная в этой статье ферма данных, максимально автоматизировать данный процесс.

Хорошо известна концепция «фермерство данных» (data farming) и его различные определения. Например: *Фермерство данных — это процесс использования разработанных вычислительных экспериментов для «выращивания» данных, которые затем можно анализировать с использованием статистических методов и методов визуализации, чтобы получить представление о сложных системах. Эти методы могут быть применены к любой вычислительной модели* [3] или *Фермерство данных — это метод, в котором используется междисциплинарный подход, включающий высокопроизводительные вычисления,*

моделирование и симуляцию, а также статистический анализ, позволяющий глубоко изучить множество вопросов с многочисленными альтернативами. Это инновационный метод тщательного изучения неопределенных событий с множеством потенциальных результатов. Проводя многочисленные эксперименты, фермерство данных позволяет нам понять как очевидные, так и неясные результаты, тем самым предоставляя содержательные ответы лицам, принимающим решения [4].

Наша концепция фермы данных заключается в том, чтобы на ферме выращивание данных, их хранение, анализ и визуализация были сведены в единую автоматизированную систему, построенную по модульному принципу конструктора «Лего». Причем, следуя принципу «Лего», мы можем как собрать эти модули, так и разобрать, переделав их, если результаты работы системы не будут удовлетворительными. А именно:

- M1) модуль выращивания (больших) данных заключается в сборе данных в какой-то предметной области из гетерогенных источников (в основном из сети Интернет) в режиме реального времени 24/7;
- M2) модуль хранения данных – это хранилище данных в NoSQL базе MongoDB;
- M3) модуль анализа данных – это предварительная обработка данных, их разметка, агрегация и выдача в формате удобном для применения машинного обучения или иных алгоритмов искусственного интеллекта;
- M4) модуль визуализации данных, собранных и обработанных на ферме – это дашборд, который дает возможность проводить наглядные численные эксперименты компьютерного моделирования и сравнивать их с реальными данными, что позволяет максимально точно вычислить или оценить параметры той или иной модели какой-либо предметной области.

Эффективность данного подхода была протестирована на данных по COVID-19 и модели, разработанной РФЯЦ – ВНИИТФ Росатома в мегапроекте «Моделирование эпидемий вирусных инфекций». Стоит также отметить, что предложенная архитектура фермы данных универсальна и легко может быть адаптирована к любой предметной области.

Проделанный анализ открытых источников информации по методам и программам не позволил выявить программное средство, удовлетворяющее всем основным требованиям к разрабатываемому в данном научном исследовании проекту «ферма данных», в котором осуществляется сбор информации из открытых источников, а также прогнозирование развития событий на основе собранных данных при помощи математической модели.

Наибольший интерес из рассмотренных систем представляет программный комплекс автоматического интеллектуального сбора данных из различных интернет источников [5] (Экспериментальный образец программного комплекса «Автоматическая интеллектуальная система сбора данных из различных интернет источников» (ЭО ПК)), разработанный в Национальном центре когнитивных разработок (НЦКР) — центре компетенции Национальной технологической инициативы (НТИ) на базе Университета ИТМО. Он обеспечивает функции создания задач сбора данных, тестирования задач, создания сценариев сбора, запуск сценариев локально и в распределенном режиме. Он может быть использован для создания различных систем мониторинга. ЭО ПК предназначен для решения следующих задач:

- сбор данных из различных источников;
- обеспечение мониторинга в сети интернет.

Сильной частью этого комплекса является удобный интерфейс и мультиканальное использование со встроенными механизмами управления и квотирования (различные интерфейсы доступа). Взаимодействие оператора с ЭО ПК осуществляется посредством графического пользовательского интерфейса, загружаемого в приложение Telegram путем манипуляций отображаемыми графическими элементами. С помощью этих манипуляций

пользователь может осуществлять работу по созданию и запуску сценариев сбора данных, в рамках которых обеспечивается функциональность ЭО ПК.

К недостаткам рассмотренного пакета программ можно отнести:

- строго интерактивный режим работы системы, который предполагает высокий уровень ручной работы оператора и требует его соответствующей квалификации;
- весьма ограниченное использование современных информационных технологий и искусственного интеллекта: непосредственно интерфейс системы представляет собой telegram-приложение (по сути это telegram-бот, ведущий «диалог» с оператором);
- отсутствие каких-либо примеров собранных системой данных;
- отсутствие средств обработки (агрегация, разметка) собранных данных;
- отсутствие возможности применения средств для прогнозирования на основе собранных данных.

Заметим, что все эти недостатки отсутствуют в ЭО ПК «ферма данных».

Среди зарубежных разработок стоит отметить BrandWatch – это платформа мониторинга социальных сетей, которая просматривает в Интернете отзывы, статьи и обсуждения. Компания задействует анализ настроений на основе правил [6]. В общем, подход компании – это сбор и обработка данных, их фильтрация, обработка и визуализация для конечного пользователя.

Упомянутые выше компании не раскрывают информацию о том, какие программные комплексы они используют, если речь идет о применении машинного обучения для анализа данных. В своем большинстве, используются опросы как через специальные формы, так и с помощью телефонных звонков. Наиболее близкой концепцией в плане сбора данных обладает отечественный университет ИТМО, а по своей общей концепции наша система имеет больше общего с зарубежной BrandWatch.

Кроме того, нам неизвестны и другие аналоги разработанной системы, что подтверждается проведенным нами патентным поиском [7] и отзывами экспертов на наши публикации [8-10]. Другие разделы статьи устроены следующим образом. Разд. 2 содержит технические требования к реализации программного комплекса. В разд. 3 описаны архитектура фермы данных, ее модули и функционалы. Статью завершают выводы, сделанные на основании исследования, список литературы и информация об авторах.

2. Требования к реализации программного комплекса «ферма данных»

При реализации программного комплекса ферма данных (ПКФД) в качестве основных ключевых условий были сформулированы следующие:

- разрабатываемый ПКФД должен позволять осуществлять поиск, извлечение и обновление требующейся информации в задаваемых форматах и с задаваемой периодичностью;
- разрабатываемый ПКФД необходимо реализовать на принципах максимального использования открытого программного обеспечения;
- хранилище данных создаваемого ПКФД должно обеспечиваться программной средой системы управления базой данных (СУБД);
- ПКФД должен быть масштабируемым, иметь модульную схему, может быть примененным в любых предметных областях.

В следующем разделе статьи представлены описание ПКФД, его компоненты и применяемые методы, позволяющие собирать, обрабатывать и хранить данные из открытых источников в автоматическом режиме. А также результирующая визуализация средствами интерактивной аналитической панели.

3. Архитектура фермы данных

Цель разработки программно-инструментальной системы «ферма данных» направлена на создание технических решений (см. раздел 2) для формирования динамических баз данных. Система осуществляет поиск и извлечение информации из структурированных и неструктурированных источников, используя технологии искусственного интеллекта и обработки больших данных. Это позволяет получить достаточный объем релевантной информации для поддержки моделей, создаваемых в рамках комплексного проекта в определенной предметной области, например, в сфере COVID-19 [9]. Одно из главных преимуществ разработанной системы является возможность быстрого развертывания сбора, хранения и обработки данных на основе Open-Source компонентов, работающих в режиме реального времени 24/7. Помимо этого, система позволяет получать оповещения о работе через мессенджеры. Она является модульной и позволяет легко заменять компоненты. Благодаря этому, спектр применения системы очень широк, так как ее можно адаптировать для структурирования больших данных в любой предметной области.

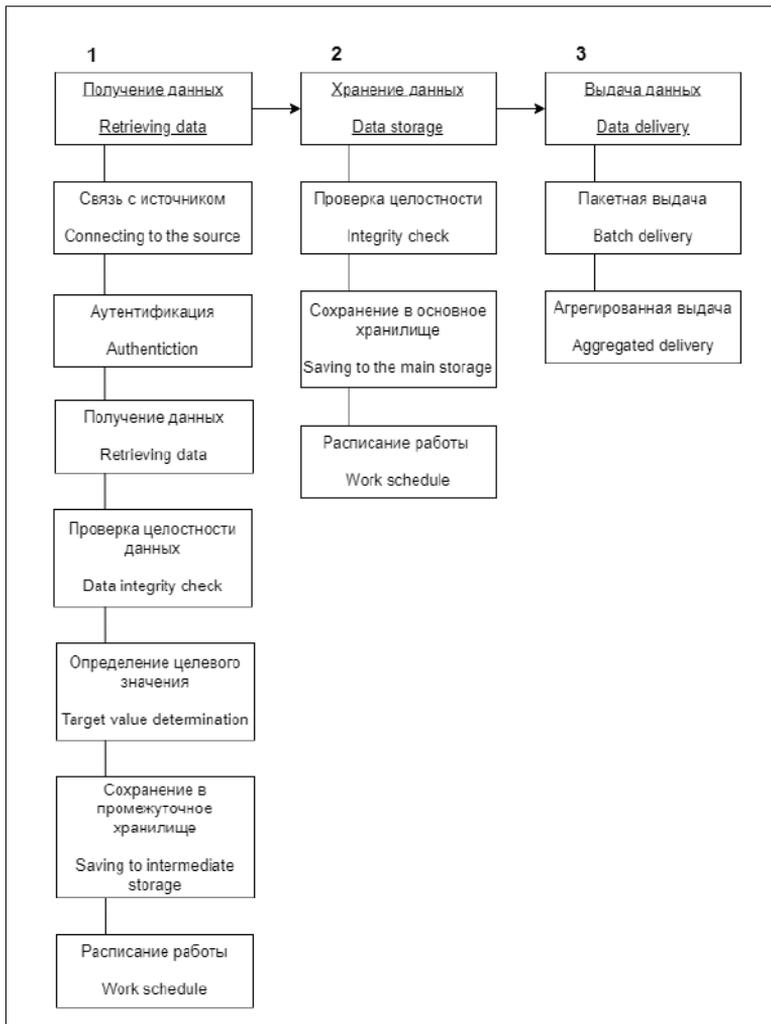


Рис. 1: Принципиальная схема работы программного комплекса ферма данных (ПКФД)
Fig. 1: Operational scheme of the data farm software package (DFSP)

В основу ПКФД легли следующие программные компоненты:

- Ubuntu server 20.4.01 LTS;
- MongoDB 4.4.1;
- Conda 4.9.0;
- Jupyter-notebook 6.0.3;
- Python 3.9.5;
- PyTorch;
- TensorFlow;
- NLTK;
- и другие [8-10].

Отметим некоторые преимущества MongoDB, которые обусловили выбор в пользу этого NoSQL хранилища данных:

- легкая масштабируемость;
- гибкая схема добавления документа;
- хранение данных в формате JSON;
- возможность глубоких запросов.

Главные процессы, представленные на рис. 1:

- подключение к ресурсу – обеспечивает функционал подключения к ресурсу данных по различным протоколам как в интернете, так и подключение к другим базам данных;
- аутентификация – обеспечивает механизмы безопасности при подключении, такие как хранение и предоставление паролей и ключей;
- получение данных – механизм опроса для синхронных и асинхронных источников данных, пакетная обработка;
- проверка данных на целостность – обеспечивает качество полученных данных.

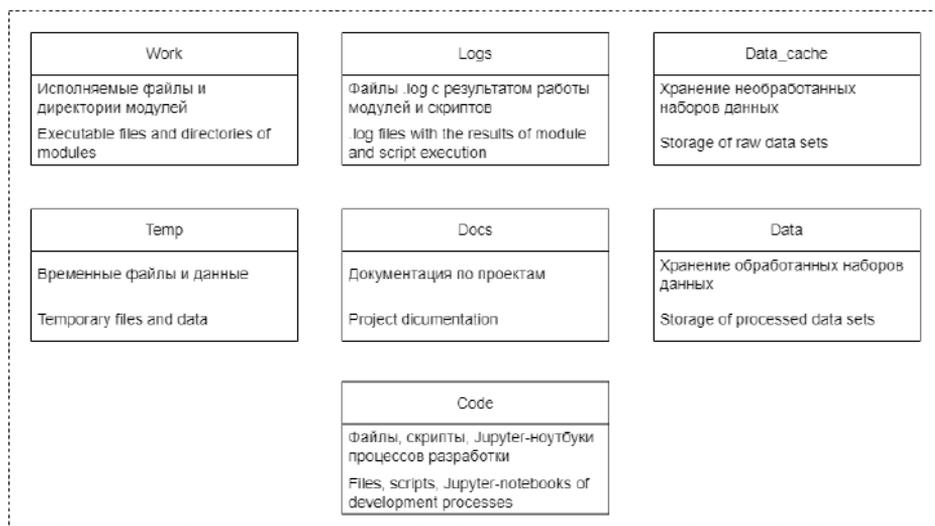


Рис. 2: Структура директорий программного комплекса ферма данных (ПКФД)
Fig. 2: Directory structure of the data farm software package (DFSP)

Кроме того, в работе ПКФД учитываются:

- информация в комментариях о ресурсе данных;
- импорт библиотек Python;
- специальные преобразования данных (очистка, понижение размерности, восстановление и т.п.);
- сохранение данных в директорию временного размещения;
- вызов функций и указание параметров для обработки данных перед тем, как поместить их в основное хранилище MongoDB.

Структура директорий ПКФД на удаленном сервере, представлена на рис. 2:

На рис. 3 показана детальная модульная структура ПКФД. Принципы организации моделей и их функциональное назначение описываются в следующих подразделах статьи.

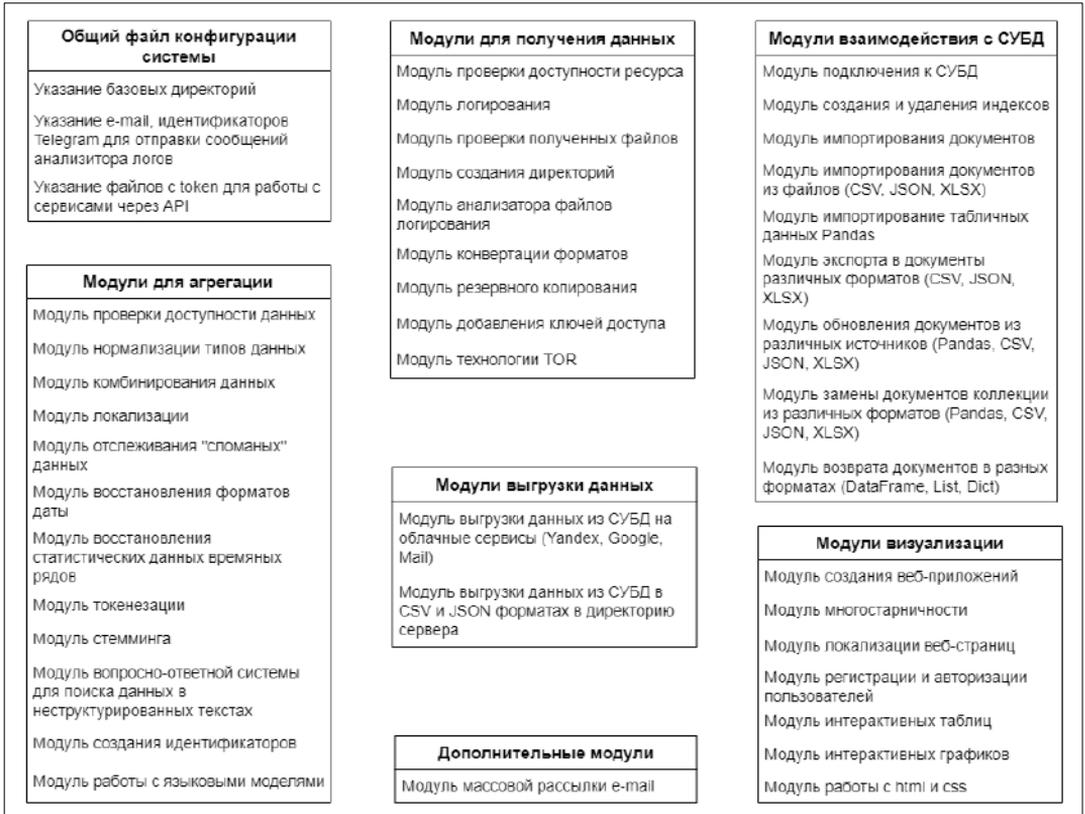


Рис. 3: Модули программного комплекса ферма данных (ПКФД)

Fig. 3: Modules of the data farm software package (DFSP)

3.1 Модуль сбора данных

Мы используем единый подход к сбору данных. Он заключается в стандартизации используемых функций, которые могут быть применимы к различным источникам. Такой подход позволяет быстро добавлять скрипты для ранее не рассматриваемых ресурсов, а также не вызывает затруднений в работе команды, так как в процессе используются одинаковые наименования и конструкции кода. Кроме того, применяются общие файлы настройки для определения директорий хранения данных, что позволяет упростить код скриптов, а также при необходимости ускорить процесс переноса системы на другие платформы (Рис. 4):

```
1# Common path download data
2main_path = '/home/user/Project/'
3download_path = main_path + 'Data_cache/'
4download_path_fast_test = download_path + 'Fast_test/'
5logs_path = main_path + 'Logs/'
6token_vk_download = main_path + 'Work/download_data/module_for_scripts/access_token.txt'
7
8# StopCoronaVirus - download separately
9StopCoronaVirus = download_old_path + 'StopCoronaVirus/'
10
11# GOGOV.RU - download separately
12Vaccinations = download_old_path + 'Vaccinations/'
13
14# Covid-19 test data St. Petersburg - download separately
15Covid_test_RU_SPE = download_old_path + 'Covid_test_RU-SPE/'
16
17# --- Actual data acquisition scripts (as well as reworked old versions of scripts) ---
18# Statistical information (Stop corona virus / Gogov and some more)
19
20Stop_corona_virus = download_path + 'Stop_corona_virus/'
21Gogov = download_path + 'Gogov/'
22Yanstat = download_path + 'Yanstat/'
23Stop_corona_vaccination = download_path + 'Stop_corona_vaccination/'
24Stop_corona_restrict = download_path + 'Stop_corona_restrict/'
25Stop_corona_russia = download_path + 'Stop_corona_russia/'
26
27# --- VK operational headquarters ---
28# --- Information in unstructured text form ---
29
30# Самарская область (Жигулёвск, Новокуйбышевск, Самара, Сызрань, Тольятти, Чапаевск)
31# Samara region (Zhigulevsk, Novokuibyshevsk, Samara, Syzran, Tolyatti, Chapaevsk)
32VK_RU_SAM = download_path + 'VK_RU_SAM/'
33
34# Алтайский край (Барнаул, Бийск, Новоалтайск, Рубцовск, Белогорск)
35# Altai Territory (Barnaul, Biysk, Novoaltaysk, Rubtsovsk, Belogorsk)
36VK_RU_ALT = download_path + 'VK_RU_ALT/'
```

Рис. 4: Фрагмент файла конфигурации директорий
Fig. 4: Fragment of the directories configuration file

На рис. 5 представлены ресурс данных Novel Coronavirus (COVID-19) Cases Data университета Джона Хопкинса и QR код ссылки на источник. На листинге 1 приведен скрипт получения данных из этого источника.

Рис. 5: Источник данных: Novel Coronavirus (COVID-19) Cases Data университета Джона Хопкинса
Fig. 5: Source data: Novel Coronavirus (COVID-19) Cases Data from John Hopkins University

```
1 from hdx.api.configuration import Configuration
2 from hdx.data.dataset import Dataset
3 from connect import connect
4 from makedir import makedir
5 from clogs import set_log
6 from copyrate import cops
7 from check_size import checking
8 from config import *
9 from rewriting import rewrite
10 from check import checks
11
12 log = log_conf
13
14 director = jhu
15 directory = director + 'Data/'
```

```
16 directory_backup = director + 'Backup/'
17
18 Configuration.create(hdx_site='prod', user_agent='A_Quick_Example',
19 hdx_read_only=True)
20
21 def save(source, direct):
22     dataset = Dataset.read_from_hdx(source)
23     resources = dataset.get_resources()
24     for res in resources:
25         url, path = res.download(folder=direct)
26         print('Resource URL %s downloaded to %s' % (url, path))
27
28 query_dataset = 'https://data.humdata.org/dataset/'
29 query_dataset_url = 'novel-coronavirus-2019-ncov-cases'
30 name_source = 'jhu'
31 set_log(log)
32
33 if connect(query_dataset+query_dataset_url, name_source, log) == True:
34     makedirs(directory)
35     makedirs(directory_backup)
36     before_size = checking.size_change_before(directory_backup)
37     rewrite(directory)
38     save(source=query_dataset_url, direct=directory)
39     after_size = checking.size_change_after(directory)
40     if checking.final_comparison(before_size, after_size, name_source)
41 == True:
42         if cops(directory, directory_backup, name_source) == True:
43             checks(director, directory_backup)
```

Листинг 1: Скрипт получения данных с Novel Coronavirus (COVID-19) Cases Data университета Джона Хопкинса

Listing 1: The script for obtaining data from Novel Coronavirus (COVID-19) Cases Data of Johns Hopkins University

Для отслеживания ошибок в модуле сбора данных задействовано логирование. В лог файл поступает вся ключевая информация о работе того или иного скрипта (рис. 6).

```
3847 2023-04-11 => 00:40:02 => WARNING => stop_corona_russia => source is working
3848 2023-04-11 => 00:40:02 => WARNING => stop_corona_russia => successful download of updated data
3849 2023-04-11 => 00:40:02 => WARNING => stop_corona_russia => backup completed successfully
3850
3851 2023-04-11 => 00:50:02 => WARNING => stop_corona_vaccination => source is working
3852 2023-04-11 => 00:50:02 => WARNING => stop_corona_vaccination => downloading data without updating
3853 2023-04-11 => 00:50:02 => WARNING => stop_corona_vaccination => backup completed successfully
3854
3855 2023-04-11 => 01:00:01 => WARNING => stop_corona_virus => source is working
3856 2023-04-11 => 01:03:50 => WARNING => stop_corona_virus => successful download of updated data
3857 2023-04-11 => 01:03:50 => WARNING => stop_corona_virus => backup completed successfully
3858
3859 2023-04-11 => 10:12:04 => WARNING => jhu => source is working
3860 2023-04-11 => 10:14:46 => WARNING => jhu => downloading data without updating
3861 2023-04-11 => 10:14:46 => WARNING => jhu => backup completed successfully
```

Рис. 6: Фрагмент файла логирования скриптов получения данных

Fig. 6: Fragment of the log file for data acquisition scripts

Анализатор файлов логирования оповещает об ошибках при загрузке данных, что весьма актуально при сборе интернет-данных. Сообщения отправляются как на электронную почту администратора ПКФД, так и в его Telegram канал. Поскольку система предусматривает возможности расширения и добавления новых разработчиков к проекту, в анализатор можно добавлять неограниченное число лиц для получения информации о работоспособности системы сбора данных (Рис. 7).



Рис. 7: Поступление сообщения от анализатора файлов логирования системы

Fig. 7: Message from the system of log file analyzer

3.2 Модуль хранения данных

Для организации хранилища данных (data warehouse), используется MongoDB – документно-ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Это NoSQL база данных, которая использует JSON-подобные документы и предоставляет драйверы и инструменты для взаимодействия с хранилищем данных MongoDB, используя различные языки программирования, такие как Python, JavaScript, Java, Go и C#. Пример документа, который может сохранять MongoDB, показан на листинге 2. Также используется Pymongo – официальная библиотека Python для взаимодействия с MongoDB, которая работает на Python 3.9.5.

```
Dashboard_ver_3> db.russia_lang_eng.find().sort({"Date": -1}).limit(1)
[
  {
    _id: '13719700186478753932',
    Date: ISODate("2023-03-27T00:00:00.000Z"),
    'Confirmed WHO': 0,
    'Deaths WHO': 0,
    'Recovered JHU': 0,
    'Daily confirmed WHO': 0,
    'Daily deaths WHO': 0,
    'Daily recovered JHU': 0,
    'Confirmed': 22595199,
    'Deaths': 397109,
    'Mortality_percent': 0.01,
    'Recovered': 21946453,
    'Infected on date': 251637,
    'Daily confirmed': 8989,
    'Daily deaths': 31,
    'Daily recovered': 8160,
    hash_row: '14139009034644559760'
  }
]
```

Листинг 2: Пример документа в MongoDB

Listing 2: An example of a document stored in the MongoDB

Работа с базой MongoDB предусматривает не только использование встроенной консоли mongoshell, но и модульные скрипты для получения данных с различных ресурсов (пример скрипта приведен на листинге 3).

```
1 from config import *
2 from mkdir import mkdir
3 from tools_mongo import *
4 from tools_aggregation import *
5 from clogs import set_log
```

```
6 import logging
7
8 log = log_conf_agg
9 set_log(log)
10 type_agg = name_db_add_world
11 database = dashboard_database_ver_3
12 address = dashboard_address
13
14 list_csv_path = find_final_csv(agggregated_final + world_path)
15 start_quick_save_2(type_agg, list_csv_path, database, address,
                      key_for_update='_id')
```

Листинг 3: Пример скрипта для добавления данных в MongoDB

Listing 3: Example of a script for adding data to MongoDB

3.3 Модуль обработки данных

Процесс агрегации данных рассматривается как предварительная обработка. Он заключается в следующем:

- агрегация данных из предварительного хранилища, сбор одинаковых данных из разных источников;
- проверка целостности данных – проверка и обработка недостающих, дублирующийся данных, проверка ошибки формата данных,
- приведение значений к общим наименованиям, локализация значений, добавление координат;
- сохранение в MongoDB – размещение в основном хранилище;
- расписание работы – запуск скриптов для получения данных;

На листинге 4 продемонстрирован пример подобного скрипта.

```
1 from config import *
2 from mkdir import mkdir
3 pd.options.mode.chained_assignment = None
4 from tools_aggregation import *
5 from tools_mongo import *
6 from clogs import set_log
7 import logging
8 log = log_conf_agg
9 set_log(log)
10
11 covid_city_name = 'covid_city.csv'
12 covid_city = raw_data + 'covid_city/'
13 covid_city_final = raw_data + 'covid_city/Final/'
14
15 os.chdir(covid_city)
16 extension = 'csv'
17 filenames_gogov_data = [i for i in
18 glob.glob('*.{0}'.format(extension))]
19 subset_columns_covid_city = ['date', 'city_name']
20
21 comb_file(name_combined_file=covid_city_name,
22 directory_rebuild=covid_city, directory_final=covid_city_final,
23 subset_columns = subset_columns_covid_city, key='date')
```

```
24 df0 = df0.drop_duplicates(subset=['date', 'city_name'], keep='first')
25
26 ru_coordinate = raw_data + 'city_coordinate/koord_russia.csv'
27 df1 = pd.read_csv(ru_coordinate, encoding='cp1251', sep=';')
28
29 df1 = df1.rename({'Город': 'city_ru_name'}, axis=1)
30
31 st = df0.merge(df1, left_on='city_ru_name', right_on='city_ru_name',
how='left')
32
33 st = st.drop(columns=['region_ru_name', 'region_name', 'city_name'])
34 st = st.rename({'Регион': 'region', 'Федеральный округ':
'federal_district', 'lat': 'latitude_dd', 'lng': 'longitude_dd',
'city_ru_name': 'city_name'}, axis=1)
35 st[['area', 'latitude_dd', 'longitude_dd']] = st[['area',
'latitude_dd', 'longitude_dd']].astype('string')
36 st['area'] = st['area'].str.replace(',', '.')
37 st['area'] = st['area'].str.replace(' ', '')
38 st['latitude_dd'] = st['latitude_dd'].str.replace(',', '.')
39 st['longitude_dd'] = st['longitude_dd'].str.replace(',', '.')
40
41 st[['city_name', 'region_code', 'region', 'federal_district']] =
st[['city_name', 'region_code', 'region',
'federal_district']].astype('string')
42 st[['area', 'latitude_dd', 'longitude_dd']] = st[['area',
'latitude_dd', 'longitude_dd']].astype('float')
43 st[['mask', 'gloves']] = st[['mask', 'gloves']] .astype('bool')
44 st['date'] = pd.to_datetime(st['date'], format='%Y-%m-%d')
45
46 hash_rows = pd.util.hash_pandas_object(st, index=True).values
47 hash_rows = pd.DataFrame(hash_rows, columns =['hash_row'])
48 st = st.join(hash_rows)
49 st['hash_row'] = st['hash_row'].astype('string')
```

Листинг 4: Пример скрипта агрегации для объединения CSV данных COVID-19 по городам Российской Федерации, добавлением координат, настройкой типов данных в столбцах и добавлением идентификаторов

Listing 4: An example of an aggregation script for combining COVID-19 CSV data for cities in the Russian Federation, adding coordinates, setting up data types in columns, and adding identifiers

Кроме того, при запуске скриптов используется системный сервис Cron для контроля времени их работы (рис. 8):

```
# Aggregation
0 3 */1 * * /home/user/anaconda3/bin/python3 /home/user/Project/Work/aggregation_data/RU_AMU.py >
10 3 */1 * * /home/user/anaconda3/bin/python3 /home/user/Project/Work/aggregation_data/RU_ALT.py
20 3 */1 * * /home/user/anaconda3/bin/python3 /home/user/Project/Work/aggregation_data/RU_KEM.py
30 3 */1 * * /home/user/anaconda3/bin/python3 /home/user/Project/Work/aggregation_data/RU_CHE.py
```

Рис. 8: Запуск скриптов агрегации по ежедневному расписанию

Fig. 8: Running aggregation scripts on a daily schedule

Модуль обработки естественного языка собирает, сохраняет и обрабатывает текстовые данные из социальных сетей, таких как VKontakte, Twitter, Telegram и др. Этот модуль настроен таким образом, чтобы вычислить или оценить параметры модели, разработанной в конкретной предметной области. В нашем случае, предметной областью были данные по COVID-19. Пример исходной записи обрабатываемых данных представлен далее (рис. 9):

7 октября в регионе подтверждено 644 случая новой коронавирусной инфекции: - 248 в г. самара - 118 в г. тольятти - 26 в г. новокуйбышевск - 23 в г. отрадный - 21 в г. кинель - 16 в нефтегорском районе - 16 в г. сызрань - 12 в кинель-черкасском районе - 12 в сергиевском районе - 11 в волжском районе - 10 в г. чапаевск - 9 в красноармейском районе - 9 в ставропольском районе - 8 в приволжском районе - 7 в алексеевском районе - 7 в кинельском районе - 7 в клявлинском районе - 7 в кошкинском районе - 6 в похвистневском районе - 6 в большеглушицком районе - 6 в богатовском районе - 6 в г. похвистнево - 5 в исакинском районе - 5 в пестравском районе - 5 в борском районе - 5 в г. жигулевск - 5 в г. октябрьск - 5 в хворостянском районе - 4 в большечерниговском районе - 4 в челно-вершинском районе - 3 в красноярском районе - 3 в шигонском районе - 3 в елховском районе - 2 в камышлинском районе - 2 в безенчукском районе - 1 в шенталинском районе - 1 в сызранском районе. 612 человек обследовано с диагнозом внебольничная пневмония и орви. 32 человека выявлено при профилактическом обследовании лиц, не имеющих клинических проявлений. нарастающим итогом в регионе зафиксировано 109622 случая коронавируса, скончались 4048 человек, в том числе женщины 60, 60, 60, 81, 87, 87, 84, 84 лет, мужчины 73, 78 лет, страдавшие заболеванием сердечно-сосудистой системы; женщины 58, 62, 84 лет, мужчина 81 года, страдавшие заболеваниями сердечно-сосудистой и эндокринной систем; женщины 72, 78, 78, 79, 79, 81, 83 лет; мужчины 55, 59 лет; женщина 67 лет, страдавшая заболеванием сердечно-сосудистой системы, онкологическим заболеванием; женщины 72, 85 лет, страдавшие заболеваниями сердечно-сосудистой системы и органов дыхания; мужчина 58 лет, страдавший инфекционным заболеванием; мужчина 67 лет, страдавший заболеванием пищеварительной системы, показатель смертности на 100 тыс. населения в области - 126,78, по стране - 146,27, коэффициент летальности в области - 3,69, в стране - 2,78. проведено лабораторных исследований на кви за сутки - 14432, нарастающим итогом 3741448.

Рис. 9: Пример полученной записи из группы оперативного штаба ВКонтакте по Самарской области
 Fig. 9: Example of a record received from the VKontakte operational headquarters group for the Samara region

date	city	infected_city	pneumonia_ARVI	death_rate_per_100_thousand_population_region	death_rate_per_100_thousand_population_country	without_clinical_manifestation
01-01-2021	самара	127	258	19.35	39.42.	45
01-04-2021	самара	80	169	34.67	67.97.	30
01-05-2021	самара	74	145	47.35	75.69.	15
01-06-2021	самара	18	64	61.35	83.47.	8
01-07-2020	самара	41	32	1.75	6.53.	19

Рис. 10: Фрагмент полученного набора данных с использованием методов обработки естественного языка
 Fig. 10: Fragment of the obtained dataset using natural language processing methods

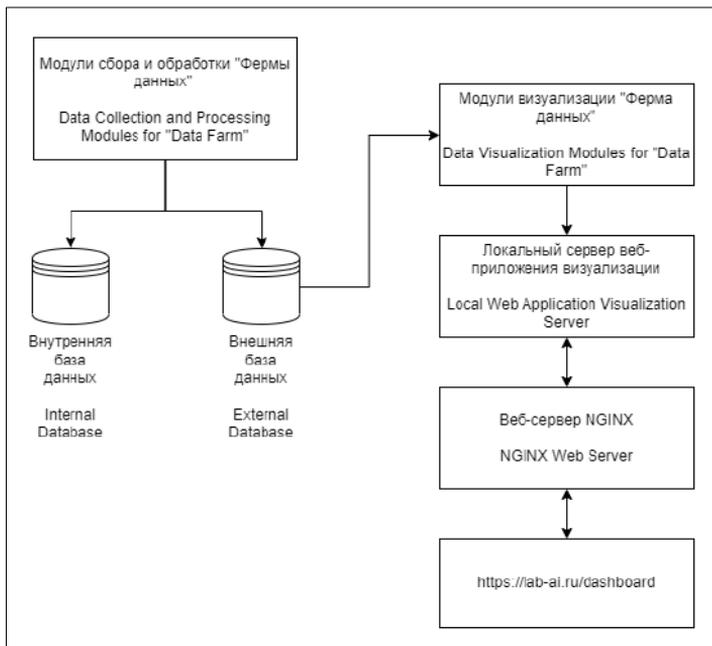


Рис. 11: Архитектура визуализации данных
 Fig. 11: Data visualization architecture

3.4 Модуль визуализации данных

Визуализация данных представлена в виде дашборда, позволяющего анализировать и сравнивать собранные и обработанные данные о ходе заболевания COVID-19 в странах мира и регионах РФ. Уникальность визуализации состоит в том, что представленные на дашборде результаты исследования позволяют точную настройку ряда управляющих параметров моделей распространения коронавирусной эпидемии. Все размещенные данные доступны для скачивания в формате CSV. На Рис. 11 показана архитектура визуализации данных:

На Рис. 12 представлен скриншот страницы дашборда, расположенного по адресу: <https://lab-ai.ru/dashboard>

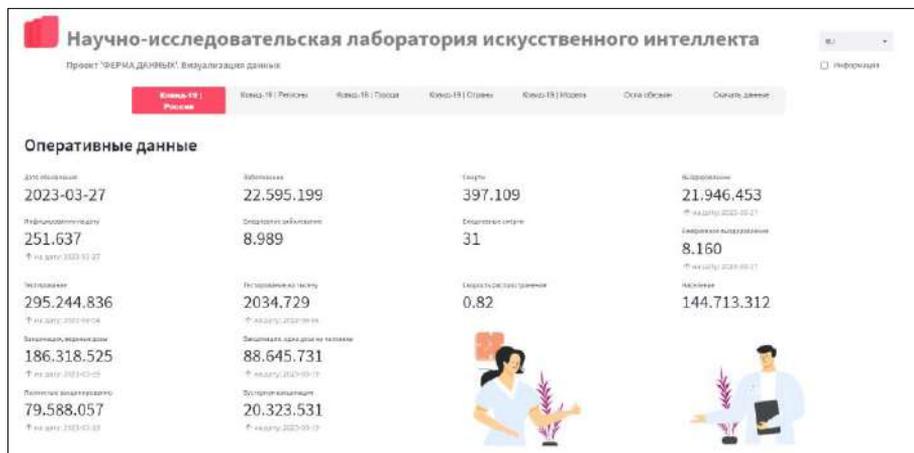


Рис. 12: Страница дашборда визуализации данных по COVID-19
Fig. 12: COVID-19 data visualization dashboard page

4. Заключение

В статье представлена оригинальная разработка автоматизированной системы сбора, хранения и обработки данных из разнородных источников («ферма данных»). Данная система призвана вычислить или оценить параметры модели в любой предметной области. Подчеркнем, что наша задача, как разработчиков, была добиться максимальной автоматизации всех процессов, используемых на ферме.

Описана архитектура фермы, ее модули и функционалы, а также их реализация на языке программирования Python. Ферма была развернута и протестирована на удаленном сервере. В качестве предметной области использовались источники данных по COVID-19 в России и других странах мира. Моделью этой предметной области служила модель распространения вирусных инфекций, разработанная РФЯЦ – ВНИИТФ Росатома. На выходе системы – файлы временных рядов в формате CSV и JSON, т.е. большие данные готовые для машинного обучения и анализа. Данные по количеству инфицированных, умерших, выздоровевших, сделанных тестов, положительных тестов, вакцинированных, случаев применения ИВЛ и т.п. представлены для городов России с населением свыше 50,000 человек.

Ферма работает в режиме реального времени 24/7, собирая, сохраняя и обрабатывая большие объемы данных. При ее разработке, широко применяются самые современные методы, технологии, среды, библиотеки и алгоритмы искусственного интеллекта и науки о данных. Вот некоторые из них: NoSQL база данных MongoDB; Anaconda; Jupyter; Pandas; Scikit-learn; NLTK; интеллектуальная оптимизация; обработка естественного языка и многие другие.

Отметим, что ферма данных находится в стадии активной разработки: добавляются новые модули и она адаптируется к другим предметным областям. В зависимости от сложности и

объема решаемых задач, а также размера данных предметной области, ферма может быть преобразована в фабрику или даже концерн данных.

Список литературы / References

- [1] Müller A.C., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016, 398 p.
- [2] Куцев Р. Разметка данных в машинном обучении: процесс, разновидности и рекомендации / Kutsev R. Data labeling in machine learning: process, variations and recommendations. Available at: <https://habr.com/ru/company/ods/blog/327242/>, accessed March 14, 2023 (in Russian).
- [3] Lucas T.W., Kelton W.D. et al, Changing the Paradigm: Simulation, Now a Method of First Resort. *Naval Research Logistics*, vol. 62, issue 4, 2015, pp. 293–305.
- [4] A. Kusiak, Data Farming: A Primer. *International Journal of Operations Research*, vol. 2, issue 2, 2005, pp. 48-57.
- [5] Экспериментальный образец программного комплекса «Автоматическая интеллектуальная система сбора данных из различных интернет источников» / Experimental sample of the software complex «Automatic intelligent system for collecting data from various Internet sources». Available at: https://actcognitive.org/files/aicrawler_2_rukovodstvo_operatora.pdf, accessed April 14, 2023 (in Russian).
- [6] Bannister K. Understanding Sentiment Analysis: What It Is & Why It's Used. Available at: <https://www.brandwatch.com/blog/understanding-sentiment-analysis/> accessed April 14, 2023.
- [7] Отчет о патентных исследованиях по тематике «ферма данных» / Patent Research Report on Data Farm. Available at: https://ai.psuti.ru/docs/Patent_search.pdf, accessed April 14, 2023 (in Russian).
- [8] Левашкин С.П., Агапов С.Н. и др, Исследование адаптивно-компарментной модели распространения КОВИД-19 в некоторых регионах РФ методами оптимизации, Математическая биология и биоинформатика, том 16, вып. 1, 2021 г., стр. 136-151 / Levashkin S.P., Agapov S.N. et al. Study of SEIRD Adaptive-Compartmental Model of COVID-19 Epidemic Spread in Russian Federation Using Optimization Methods. *Mathematical Biology and Bioinformatics*, vol. 16, issue 1, 2021, pp. 136-151.
- [9] Проект 'ФЕРМА ДАННЫХ'. Визуализация данных. Научно-исследовательская лаборатория искусственного интеллекта / Project 'DATA FARM'. Artificial Intelligence Research Laboratory. Available at: <https://lab-ai.ru/dashboard>, accessed April 14, 2023 (in Russian).
- [10] Левашкин С.П., Захарова О.И. и др. Модульная система сбора данных. Свидетельство о регистрации программы для ЭВМ, № 2022617725. Дата государственной регистрации в реестре программ для ЭВМ 25.04.2022 / Levashkin S.P., Zakharova O.I. et al. Modular data collection system. Certificate of registration of a computer program, № 2022617725. Date of state registration in the register of computer programs 25.04.2022 (in Russian).

Информация об авторах / Information about authors

Сергей Павлович ЛЕВАШКИН – профессор, кандидат физико-математических наук, PhD in Computer Science, действительный член Академии наук Мексики, заведующий научно-исследовательской лабораторией искусственного интеллекта. Ученый с более чем 20-летним опытом работы в университетах и компаниях России, Северной Америки и Европы в области искусственного интеллекта и машинного обучения.

Sergey Pavlovich LEVASHKIN is a Professor, a Candidate of Physical and Mathematical Sciences, PhD in Computer Science, a Full Member of the Academy of Sciences of Mexico, and the head of the Artificial Intelligence Research Laboratory. He is a scientist with more than 20 years of experience working in universities and companies of Russia, North America, and Europe in the field of artificial intelligence and machine learning.

Константин Николаевич ИВАНОВ – магистрант, инженер НИЛ ИИ ПГУТИ. Научные интересы включают: сбор данных, обработку естественного языка, построение веб-сервисов и приложений, визуализация данных.

Konstantin Nikolaevich IVANOV is a Master's Student and an engineer at the Artificial Intelligence Research Laboratory. His research interests include data collection, natural language processing, building web services and applications, data visualization.

Сергей Владимирович КУШУКОВ – магистрант, инженер НИЛ ИИ ПГУТИ. Научные интересы включают: информационные системы и технологии, сбор и обработка информации, написание веб-приложений.

Sergey Vladimirovich KUSHUKOV is a Master's Student and an engineer at the Artificial Intelligence Research Laboratory. His scientific interests include data collection and processing, as well as web application development.

DOI: 10.15514/ISPRAS-2023-35(2)-6



Типизированные неизвестные значения: шаг к решению проблемы представления отсутствующей информации в реляционных базах данных

С.Д. Кузнецов, ORCID: 0000-0002-8257-028X <kuzloc@ispras.ru>

*Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25*

*Московский государственный университет им. М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1*

*Московский физико-технический институт,
141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9
НИУ «Высшая школа экономики»,
101978, Россия, Москва, ул. Мясницкая, д. 20*

Аннотация. Состояние дел области управления отсутствующей информацией в реляционных базах данных оставляет желать лучшего. В стандарте SQL для представления отсутствующих данных используется универсальное null-значение, а управление основано на трехзначной логике, в которой null-значение отождествляется с третьим логическим значением. Это решение концептуально противоречиво и часто приводит к интуитивно непонятному поведению СУБД. Альтернативный подход с использованием типизированных специальных значений перекладывает всю обработку отсутствующих данных на пользователей. В этой статье мы анализируем многолетнюю историю исследований и разработок, которая привела к такой ситуации. Мы приходим к выводу, что в стандарте SQL и не могло появиться другое решение из-за выбора более 50 лет тому назад механизма универсального null-значения, а альтернативный механизм не может обеспечить системную поддержку специальных значений из-за использования двухзначной логики. Мы предлагаем комбинированный подход с использованием типизированных специальных значений на основе трехзначной логики. Этот подход позволяет использовать семантику типов данных при обработке запросов с условиями, включающими неизвестные данные. Кроме того, наш подход позволяет определить полноценную трехзначную логику, в которой специальное значение булевского типа является третьим логическим значением.

Ключевые слова: реляционные базы данных; отсутствующая информация; null-значение; трехзначная логика; типизированные специальные значения

Для цитирования: Кузнецов С.Д. Типизированные неизвестные значения: шаг к решению проблемы представления отсутствующей информации в реляционных базах данных. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 73-100. DOI: 10.15514/ISPRAS-2023-35(2)-6

Typed unknown values: a step towards solving the problem of representing missing information in relational databases

S.D. Kuznetsov, ORCID: 0000-0002-8257-028X <kuzloc@ispras.ru>

*Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

Lomonosov Moscow State University,

GSP-1, Leninskie Gory, Moscow, 119991, Russia

Moscow Institute of Physics and Technology (State University),

9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia

National Research University Higher School of Economics

20, Myasnitskaya Ulitsa, Moscow, 101978, Russia

Abstract. The state of affairs in the area of missing information management in relational databases leaves much to be desired. The SQL standard uses the universal null value to represent missing data, and the control is based on three-valued logic, in which the null value is identified with a third boolean value. This solution is conceptually inconsistent and often results in DBMS behavior that is not intuitive. An alternative approach using typed special values leaves all handling of missing data to users. In this article, we analyze the long history of research and development that led to this situation. We come to the conclusion that no other solution could have appeared in the SQL standard due to the choice of the mechanism of the universal null value more than 50 years ago, and the alternative mechanism cannot provide system support for special values due to the use of two-valued logic. We propose a combined approach using typed special values based on three-valued logic. This approach allows you to use the semantics of data types when processing queries with conditions that include unknown data. In addition, our approach allows us to define a full-fledged three-valued logic in which a special value of a Boolean type is the third boolean value.

Keywords: relational databases; missing information; null value; three-valued logic; typed special values

For citation: Kuznetsov S.D. Typed unknown values: a step towards solving the problem of representing missing information in relational databases. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 73-100 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-6

1. Введение

Проблема отсутствующей информации появилась одновременно с первыми попытками реализовать реляционную модель Эдгара Кодда. В первых своих статьях [1, 2], описывающих реляционную модель данных, Кодд вовсе не говорил о существовании этой проблемы. Похоже, что первый раз о потребности сохранять в реляционных базах данных отношения, в которых не во всех атрибутах кортежей содержатся реальные данные (потому что они неизвестны) Кодд написал в [3]. По-видимому, не просто так заметка [3] появилась в том же выпуске Бюллетеня ACM-SIGMOD за 1975-й год, что и отчет о панельной дискуссии, которая была проведена в том же году на Национальной компьютерной конференции [4]. Похоже, что на Кодда сильно повлияли замечания участников проектов по реализации реляционных систем баз данных, на что, в частности указывают слова благодарности в [3] в адрес участников проекта System R. Координированность работ Кодда и проекта System R подтверждается и тем фактом, что уже в 1976-м году была опубликована первая развернутая статья про язык баз данных SEQUEL 2 [5], в которой, по сути, повторялись мысли Кодда, зафиксированные в [3].

Еще через три года Кодд опубликовал свою знаменитую статью [6], в которой, в частности, определил реляционную алгебру в расчете на тот случай, когда в отношениях базы данных могут содержаться неопределенные значения. Можно считать, что с этого времени в языке SQL были закреплены механизм неопределенных значений и трехзначная логика, хотя, по-видимому, многим авторитетным специалистам этот подход не нравился, потому что в стандарте SQL он был в некотором виде зафиксирован только в 1999 г. [7].

Похоже, что и сам Кодд был не слишком удовлетворен средствами поддержки работы с отсутствующей информацией, которые вошли в SQL при его благословении. В конце 1980-х он опубликовал книгу [8], представляющую новый вариант реляционной модели данных, в котором были исправлены некоторые недостатки первой версии модели, попавшие в SQL. Исправления коснулись и неопределенных значений: были введены два вида маркеров, один из которых использовался для представления неизвестных значений, а другой – для обозначения того, что для конкретного кортежа значение некоторого атрибута определено быть не может. Соответственно, для корректного использования этого подхода потребовалась уже четырехзначная логика. В сообществе SQL этот подход востребован не был.

Многим людям не нравился (и не нравится) механизм неопределенных значений и трехзначной логики, предложенный Коддом во второй половине 1970-х и поддерживаемый в SQL. Но наиболее активным и последовательным противником этого механизма всегда являлся ученик Кодда, последователь исходных идей реляционной модели данных, всемирно известный писатель и лектор Кристофер Дейт. Дейт критикует SQL вообще и средства поддержки неопределенных значений, включая трехзначную логику в частности, во множестве статей и большинстве книг. В этой статье мы не будем обсуждать эту критику. Заметим лишь, что она всегда очень подробно и убедительно.

Однако здесь для нас более важно то, что с середины 1980-х Дейт и его многолетний соратник и соавтор Хью Дарвен пытались (и до недавнего времени продолжали пытаться) найти решение проблемы отсутствующей информации в базах данных, не используя понятие нетипизированного неопределенного значения и не прибегая к трехзначной логике. Первый подход [9, 10], которого они в течение долгого времени придерживались, основывается на понятии специальных значений, расширяющих любой тип данных и приводящих к получения булевского значения *true* только при сравнении на равенство специального значения с ним же самим. Однако, как указывали критики этого подхода, он не работает в важных на практике случаях, когда на множестве значений базового типа, расширяемого специальным значениям, имеется отношение порядка. И хотя Дейт и Дарвен продолжали придерживаться подхода со специальными значениями до середины первого десятилетия 2000-х, позже они отказались от специальных значений и занялись поиском других решений. Мы кратко обсудим эти решения в основной части статьи, но сейчас отметим, что несмотря на отсутствие в них видимых ошибок, ни одно из них не годится для практического использования. Ни одно из них даже не включено в последнее издание книги Дейта и Дарвена, в которой они излагают свое видение современной реляционной модели данных [11].

В результате многолетних размышлений у нас сложилось мнение, что

- a) неопределенные значения и трехзначная логика, принятые в стандарте SQL:1999 [7], никуда не годятся из-за того, что NULL не типизирован, а без этого трехзначная логика становится, мягко говоря, неполноценной;
- b) специальные значения Дейта и Дарвена [9, 10] (почти) никуда не годятся, поскольку их невозможно использовать без поддержки трехзначной логики;
- c) значит, стоит попробовать скрестить специальные значения с трехзначной логикой, что
- d) не решает проблему в целом, но выглядит приличнее, чем подход стандарта SQL и подход специальных значений.

Во втором разделе оставшейся части статьи сравнительно подробно описываются наиболее важные вехи истории проблемы отсутствующей информации. В третьем разделе мотивируется и обсуждается предлагаемое решение. Наконец, четвертый раздел содержит заключение с обсуждением ограничений и недостатков описанного подхода.

2. Исторический контекст

За десятилетия, прошедшие после введения Эдгаром Коддом реляционной модели данных [1, 2], была опубликована масса статей, так или иначе касающихся проблемы отсутствующих данных в реляционных базах данных. В этой статье мы не будем пытаться привести какой-либо обзор результатов во всех направлениях исследований, касающихся этой проблемы. Нас здесь интересует один аспект: представление в атрибутах кортежей отношений (или столбцах строк таблиц) отсутствующей информации и механизм манипулирования такими отношениями (или таблицами). Однако и этому аспекту было посвящено много публикаций чисто теоретического характера, которые (по нашему мнению) не оказали заметного влияния на технологию реляционных систем управления данными. Соответствующие результаты также не будут рассматриваться в этой статье.

В первом подразделе этого раздела будут обсуждаться основные вехи, которые привели к механизму работы с отсутствующими данными, зафиксированному в стандарте SQL:1999 [7]. Здесь, конечно, на первом месте стоят работы Кодда. Второй подраздел посвящен попыткам обойтись в реляционной модели данных без null-значения и трехзначной логики и опирается, главным образом, на работы Дейта и Дарвена. Основная цель раздела, кроме представления обзора работ, состоит в том, чтобы показать недостатки имеющихся подходов и мотивировать попытку автора данной статьи улучшить текущее состояние дел.

2.1 Путь к SQL:1999

2.1.1 Реляционная модель данных: версия 1

Как отмечалось в начале статьи, этот путь начался с публикации Коддом в 1975 г. небольшой заметки [3]. Основные результаты этой заметки (относящиеся к рассматриваемой здесь проблеме) состоят в следующем.

- 1) Предлагается решение только одной части проблемы – работа с отсутствующими данными без уточнений сути других частей. Для этого вводится понятие *null-значения* (*null value*)¹, обозначаемого в [3] символом @.
- 2) Арифметические операции сложения, вычитания, умножения и деления возвращают null-значение, если хотя бы операнд является null-значением.
- 3) Поскольку результатом сравнения $x = y$, в котором неизвестны какой-либо один или оба аргумента, не является ни *true*, ни *false*, вводится неизвестное логическое значение, что делает используемую логику трехзначной. Для обозначения неизвестного логического значения используется тот же символ @, потому что логические значения могут сохраняться в базе данных, и желательно обрабатывать все неизвестные или неопределенные значения единообразно². Предлагаемая трехзначная логика определяется следующими таблицами истинности.

AND	false	@	true
false	false	false	false
@	false	@	@
true	false	@	true

OR	false	@	true
false	false	@	true
@	@	@	true
true	true	true	true

$$\text{NOT}(false) = true; \text{NOT}(@) = @; \text{NOT}(true) = false.$$

¹ Следуя традиции, мы используем в своей статье этот термин, хотя, вообще говоря, он некорректен. Null-значение – это не значение, потому что любое значение должно принадлежать хотя бы одному типу данных, а null-значение нетипизировано. Мы предлагаем считать этот термин просто сокращенной формой от *обозначения отсутствия значения*.

² Здесь Кодд говорит об этом мимоходом, но на самом деле предложенный подход к пониманию сути третьего логического значения имел далеко идущие последствия, о которых мы поговорим позже в этом разделе.

- 4) Для определения значения логического выражения предлагается следующий *принцип замены неопределенных значений*. Значением логического выражения является @ тогда и только тогда, когда одновременно выполняются два следующие условия:
 - a. можно заменить каждое вхождение @ некоторым допустимым реальным значением (возможно, разными значениями для разных вхождений @) таким образом, что значением полученного выражения является *true*;
 - b. можно заменить каждое вхождение @ некоторым допустимым реальным значением (возможно, разными значениями для разных вхождений @) таким образом, что значением полученного выражения является *false*.
- 5) В соответствии с принципом замены неопределенных значений значение @ вырабатывается любым сравнением $x \theta y$, где $\theta - <, \leq, >, \geq$, если один из аргументов или оба являются @.
- 6) Для каждого натурального числа n допустимым является кортеж, состоящий из n неопределенных значений. Однако никакое n -арное отношение не может содержать более одного такого кортежа. Обычным правилом является то, что никакие отношения не могут содержать кортежи-дубликаты (в некоторых реляционных системах это правило строго не соблюдается, и для этих систем описываемая схема нуждается в модификации). В соответствии с правилом недопущения дубликатов, неопределенное значение в одном кортеже считается таким же, как неопределенное значение в другом кортеже³. Это отождествление одного неопределенного значения с другим может казаться противоречащим нашему назначению истинностного значения для сравнения @ по равенству. Однако выявление кортежей для удаления дубликатов не следует путать со сравнением по равенству при вычислении условий поиска.
- 7) Кроме того, в статье приводится набросок того, как должны выглядеть реляционная алгебра и реляционное исчисление при наличии неопределенных значений.

Уже на следующий год участниками проекта System R была опубликована статья [5], в которой, в частности, кратко описывается внедрение идей Кодда [3] с очень незначительными (но важными) изменениями в прототип языка SQL – SEQUEL 2. Основные черты языка (и его реализации), связанные с поддержкой неопределенных значений, можно охарактеризовать следующим образом.

- 1) Допускается существование в базе данных неизвестных, или null-значений. На null-значение можно сослаться с помощью ключевого слова NULL.
- 2) При вычислении всех агрегатных функций (в System R они назывались встроенными, *built-in*), кроме COUNT), null-значения игнорируются.
- 3) При определении того, удовлетворяет ли данный кортеж условию раздела WHERE запроса, предикатам, в которых участвуют атрибуты, содержащиеся в соответствующем кортеже null-значение, присваивается неизвестное истинностное значение (обозначается знаком вопроса ?)⁴.
- 4) Для вычисления всего условия раздела WHERE используется интерпретация логических операций AND, OR и NOT в трехзначной логике на основе тех же таблице истинности, что в [3].
- 5) Кортеж, для которого вычисляется условие раздела WHERE запроса, удовлетворяет этому условию в том и только в том случае, когда оно вычисляется в *true*.
- 6) Условия, задаваемые в ограничениях целостности (*assertion* в терминологии System R), также вычисляются в трехзначной логике и считаются удовлетворенными тогда и только тогда, когда результатом этого вычисления не является *false*.

³ В одном и том же атрибуте.

⁴ В этом состоит существенное отличие от [3], где Кодд используется одно и то же обозначение как для неизвестного значения вообще (не принадлежащего никакому типу данных), так и для третьего логического значения, которое должно было бы принадлежать булевскому (трехзначному) типу.

Больше в статьях, посвященных System R, проблематика отсутствующих значений и трехзначной логики не затрагивалась. Как мы увидим дальше, в контексте SQL инициатива (достаточно вялая) перешла к стандартизаторам этого языка.

В 1979 г. Кодд опубликовал статью [6], которая, как подчеркивает ее название, была главным образом направлена на расширение реляционной модели данных с целью обеспечения более высокого семантического уровня данных, хранимых в реляционных базах данных. Однако в [6] Кодд затрагивает и проблему отсутствующей информации, но при этом лишь дословно повторяет то, о чем говорилось в [3].

2.1.2 Реляционная модель данных: версия 2

Следующий раз Эдгар Кодд обратился к проблеме отсутствующей информации в 1986 г. в [12]. Кратко говоря, в этой статье он впервые предлагает расширение реляционной модели данных, позволяющее работать в двумя видами отсутствующих данных – неизвестными, но возможными (*missing and applicable*, A) и недопустимыми (*missing and inapplicable*, I) данными. Вот суть новых предложений.

- 1) Отсутствующие A- и I-данные обозначаются A- и I-маркерами (*mark*) соответственно.
- 2) Семантика того факта, что некоторое значение в базе данных отсутствует, отлична от семантики значения, хранимого в базе данных. Первый факт применим к любому значению, независимо от его типа. Семантика реального значения сильно зависит от его домена (или типа данных), из которого атрибут отношения получает свои значения.
- 3) Если x обозначает некоторое реальное значение, хранимое в базе данных, A – A-маркер и I – I-маркер, то

$$\begin{array}{lll} x + x = 2x & x + A = A & A + x = A \\ A + A = A & A + I = I & I + A = I \\ I + I = I & x + I = I & I + x = I \end{array}$$

Аналогичные таблицы справедливы и для арифметических операций вычитания, умножения и деления, а также для операции конкатенации символьных строк.⁵

- 4) Введение второго неизвестного значения наводит на мысль об уместности применения четырехзначной логики, однако Кодд считает это *несрочным делом, которое можно отложить, пока у практиков не появится соответствующая потребность*. Поэтому в [12] сохраняется трехзначная логика. Это обосновывается следующим образом. *Сравнение по равенству двух маркеров недопустимых значений имеет смысл обрабатывать так же, как и сравнение двух фактических значений. Однако любое вхождение I-маркера может быть обновлено специально уполномоченным пользователем, преобразуясь в A-маркер, а затем, возможно, в фактическое значение.*

Поэтому можно считать, что в операции реляционного языка всякий раз, когда A- или I-маркер приравнивается в условии к фактическому значению или к A-маркеру, или к I-маркеру истинностным значением такого условия всегда считается MAYBE (так в [12] называется третье истинностное значение). Аналогично, полагается, что результатом любого сравнения, в котором A- или I-маркер сравнивается с фактическим значением или A-маркером, или к I-маркером, является MAYBE.

Правила построения сложных условий на основе простых сравнений определяются такими же таблицами истинности булевских операций дизъюнкции, конъюнкции и отрицания, как и в [3, 6], но теперь операндами логических операций являются

⁵ Удивительно, что такие таблицы появились у Кодда только спустя десять лет после введения им неопределенных значений.

высказывания P и Q, каждое из которых может иметь значение *true* (*t*), *maybe* (*m*) или *false* (*f*)⁶.

P	not P
t	f
m	m
f	t

P or Q		Q		
		t	m	f
P	t	t	t	t
	m	t	m	m
	f	f	f	m

P & Q		Q		
		t	m	f
P	t	t	m	f
	m	m	m	f
	f	f	f	f

В 1987 г. Кодд опубликовал статью [13], в которой содержатся разъяснения идей, высказанных в [12], а также несколько изменений, которые сам Кодд считает незначительными, однако нам они представляются очень существенными.

- 1) Кодд все-таки вводит четырехзначную логику. Как и в [12], отсутствующие A- и I-данные обозначаются A- и I-маркерами соответственно. Пусть имеется скалярная функция с одним или несколькими аргументами. Считается, что I-маркер сильнее A-маркера. В общем случае, если самым сильным маркером аргументов вызова является I-маркер, то результат вызова скалярной функции – I-маркер. Если, с другой стороны, самым сильным маркером является A-маркер, то результат – A-маркер.

Например, арифметические выражения вычисляются по следующим правилам (символ % обозначает любую арифметическую операцию, z – обычное значение, a – A-маркер, i – I-маркер):

$$\begin{array}{llll}
 z \% a = a & z \% i = i & a \% z = a & i \% z = i \\
 a \% a = a & a \% i = i & i \% a = i & i \% i = i
 \end{array}$$

- 2) Булевский тип (домен) расширяется двумя дополнительными значениями a-MAYBE и i-MAYBE. Кодд обозначает их так же, как и A-маркер и I-маркер, -- просто символами a и i соответственно, но подчеркивает, что t (*true*), f (*false*), a и i – это реальные значения булевого типа, а не маркеры отсутствующих данных. Ни в [13], ни в [8] явно не говорится, как вычисляются сравнения скалярных значений, но если следовать общему правилу, сформулированному в предыдущем разделе, то должна быть справедлива сконструированная нами таблица (Кодд ничего подобного не приводит и вообще не говорит о сравнениях в четырехзначной логике). В этой таблице по необходимости мы используем разные обозначения для маркеров отсутствующих данных (a и i) и для соответствующих булевских значений (a_B и i_B). Символ Θ обозначает произвольную операцию сравнения скалярных значений ($=, \neq, <, \leq, >, \geq$), z – обычное значение.

$$\begin{array}{llll}
 z \Theta a = a_B & z \Theta i = i_B & a \Theta z = a_B & i \Theta z = i_B \\
 a \Theta a = a_B & a \Theta i = i_B & i \Theta a = i_B & i \Theta i = i_B
 \end{array}$$

- 3) В соответствии с тем же правилом строятся таблицы истинности в четырехзначной логике для операций NOT, OR и AND. Ниже приводятся таблицы из [13], в которых для обозначения истинностных значений используются символы t, f, a и i.

P	not P
t	f
a	a
i	i
f	t

P or Q		Q			
		t	a	i	f
P	t	t	t	t	t
	a	t	a	a	a
	i	t	a	i	f
	f	t	a	f	f

P & Q		Q			
		t	a	i	f
P	t	t	a	i	f
	a	a	a	i	f
	i	i	i	i	f
	f	f	f	f	f

- 4) В [13] Кодд считает, что дополнительная сложность четырехзначной логики в настоящее время не оправдана, тем более что проектировщикам и пользователям СУБД не очень просто жить и с трехзначной логикой. Поэтому пока уместно продолжать встраивать в

⁶ То есть здесь устранена путаница между третьим логическим значением и обозначением отсутствующих данных.

СУБД трехзначную логику. Однако внешние спецификации СУБД должны позволять переход с трехзначной на четырехзначную логику без ущерба для пользователей ~ без переписывания приложений (или с минимальным переписыванием). Если в СУБД встраивается четырехзначная логика, то либо она должна согласовываться подходом, описанным выше, либо отклонения от этого подхода должны быть хорошо обоснованы.⁷

В начале 1990-го г. Кодд опубликовал книгу [8], в которой систематически и подробно описал вторую версию реляционной модели данных (RM/V2), считая первой версией (RM/V1) ту, которая была введена в начале 1970-х [1, 2]. Расширения реляционной модели (RM/T), введенные в [6], в RM/V2 не вошли, но Кодд планировал добавить их в следующих версиях реляционной модели данных. К сожалению, следующие версии реляционной модели так и не появились, и похоже, что работы Кодда, приведшие к написанию [8], остались недооцененными сообществом баз данных.

Что касается средств управления отсутствующей информацией, описываемых в [8], они в точности совпадают с теми, которые были введены в [12, 13]. Единственным существенным дополнением является то, что в RM/V2 строго требуется поддержка четырехзначной логики. Больше Кодд к этой теме не обращался, да и вообще не публиковал работ про реляционную модель данных.

2.1.3 Первые стандарты языка SQL

Тем временем, в 1986 г. Американским национальным институтом стандартов (ANSI) был принят первый национальный стандарт языка SQL (SQL-86) [14], который через год был одобрен Международной организацией по стандартизации (ISO) в качестве международного стандарта [15]. По поводу отсутствующих данных в стандарте практически полностью повторяются соображения разработчиков System R, приведенные в [5] (в еще более расплывчатом виде). Основные соответствующие спецификации выглядят так.

- 1) Тип данных – это множество представимых значений. Логическим представлением значения является литерал. Физическое представление значения определяется в реализации. Значение является примитивным в том смысле, что в рамках данного стандарта оно не подразделяется на более мелкие части. Значение является либо null-значением, либо не-null-значением. Null-значение – это определяемое в реализации зависящее от типа специальное значение, отличное от всех не-null-значений этого типа. Не-null-значение может быть либо строкой символов, либо числом.
- 2) Если в арифметическом выражении встречается хотя бы одно null-значение, то значением всего выражения является null-значение. Кроме арифметических выражений над данными какого-либо числового типа, в SQL-86 поддерживаются выражения над строками символов, но в них не может быть операций, так что это вырожденные выражения, всего лишь специфицирующие символьные строки.
- 3) Если в предикате сравнения $x \text{ comp-op } y$ ($\text{comp-op} = | <> | < | > | <= | >=$, x – выражение, y – выражение или подзапрос, выдающий не более одного значения) x и y являются выражениями, значением хотя бы одного из которых является null-значение, или если y – подзапрос, выдающий пустое множество результатов, то значением сравнения $x \text{ comp-op } y$ является *unknown*.

⁷ На самом деле, подход Кодда с четырехзначной логикой ненамного лучше того, который в настоящее время принят в SQL (об этом см. ниже в данном подразделе). Булевский тип выглядит неполноценным, потому что либо нужно прибегнуть в тем же ухищрениям, что и в SQL (запретить хранить в базах данных булевские значения, запретить прямо использовать в логических выражениях литералы булевского типа и т.д.), либо мы столкнемся, например, с потребностью представлять в базе данных отсутствующие значения булевского типа и, соответственно, каким-то образом вычислять логические выражения, в которых присутствуют маркеры отсутствия данных.

- 4) Для проверки того, что в указанном столбце текущей строки находится null-значение, в стандарте определен предикат IS NULL, выдающий значения *true* или *false*.
- 5) Условие выборки (*search condition*) – это, по сути, логическое выражение, строящееся на основе предикатов с использованием логических операций NOT, AND и OR с естественными приоритетами операций и возможностью расстановки скобок. Результатом условия выборки для текущей строки может быть *true*, *false* и *unknown* на основе интерпретации логического выражения в трехзначной логике с теми же таблицами истинности, что и в [5], с заменой обозначения неизвестного истинностного значения ? на *unknown*.
- 6) Результатом выполнения раздела WHERE является таблица, включающая те строки таблицы, полученной после обработки раздела FROM, для которых значением условия выборки является *true*.

Вдгонку к SQL-86, в 1989 г. был принят следующий международный стандарт SQL-89 [16]. Как следует из его полного названия, основные расширения языка касались средств поддержки целостности баз данных. В стандарте SQL-86 обеспечивались всего два вида определений ограничений целостности: NOT NULL на уровне определения столбца таблицы (запрет наличия в этом столбце любой строки соответствующей таблицы null-значения) и UNIQUE на уровне определения столбца или вне таких определений с возможностью определить возможный ключ, состоящий из нескольких столбцов (ни в одном столбце любого возможного ключа не допускается наличие null-значения). Эти возможности сохранились в SQL-89, но появились возможности определения в операции CREATE TABLE ограничений внешнего ключа, проверочных ограничений и, кроме того, определения первичного ключа таблицы.

- 1) Ссылочное ограничение внешнего ключа можно определить на уровне определения столбца (REFERENCES), если внешним ключом является соответствующий столбец, или в виде отдельного табличного ограничения (FOREIGN KEY), допускающего внешние ключи из нескольких явно указываемых столбцов. В обоих случаях указывается имя таблицы, на строки которых будут указывать значения внешнего ключа, а также, возможно, список столбцов этой таблицы, составляющих какой-либо ее возможный ключ. В определении внешнего ключа допускается отсутствие явного указания этого списка, и тогда по умолчанию считается, что внешнему ключу соответствует первичный ключ соответствующей таблицы, который в этом случае должен быть в ней явно объявлен.

Это единственный повод для введения SQL-89 возможности указания в определении возможного ключа наряду со спецификацией UNIQUE возможности указания PRIMARY KEY (не более чем для одного возможного ключа). Во всем остальном первичный ключ обладает ровно теми же свойствами, что и любой возможный ключ.

- 2) Концептуально любое ссылочное ограничение таблицы (как и любое проверочное ограничение, см. ниже) проверяется при выполнении любой операции обновления базы данных и считается удовлетворенным, если каждое значение внешнего ключа (возможно, составного), содержащееся в строках ссылающейся таблицы, ни один компонент которого не является null-значением, совпадает со значением соответствующего возможного ключа в какой-либо строке таблицы, на которую указывает ссылка, либо хотя бы один компонент значения внешнего ключа является null-значением (это соответствует варианту MATCH SIMPLE определения внешнего ключа в современных вариантах стандарта SQL).
- 3) Наконец, проверочное ограничение CHECK, которое также можно определить как на уровне определения столбца, так и в виде отдельного табличного ограничения, содержит условие выборки, которое должно вычисляться для каждой отдельной строки соответствующей таблицы и считается удовлетворенным, если ни для какой строки не принимает истинностного значения *false*.

В 1992 г. был принят следующий стандарт SQL-92. По сравнению с двумя первыми стандартами SQL-92 был очень значительно расширен, о чем можно судить даже просто по его объему в более чем 600 стр. против 120 стр. в SQL-89. К сожалению, средства поддержки отсутствующих данных остались на очень посредственном уровне. Перечислим части стандарта, затрагивающие интересующую нас тему.

- 1) Определение: null-значение (null) – это специальное значение, или маркер, используемый для обозначения отсутствия какого-либо типизированного значения.
- 2) В разделе «Типы данных»: тип данных – это множество представимых значений. Логическим представлением значения является литерал. ... Значение является null-значением или не-null-значением.

Null-значение – это зависящее от реализации специальное значение, отличное от любого не-null-значения типа данных, отсутствие значений которого оно обозначает. В действительности имеется только одно null-значение, и оно является элементом каждого типа данных SQL. Для null-значения отсутствует литерал, хотя в некоторых местах используется ключевое слово NULL для обозначения потребности в null-значении.⁸

- 3) В SQL-92 допускается использование значительно большего числа типов данных, чем в SQL-89: CHARACTER, CHARACTER VARYING, BIT, BIT VARYING, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE PRECISION, DATE, TIME, TIMESTAMP, INTERVAL. Соответственно, увеличилось и число возможных операций, видов выражений и предикатов. Но здесь для нас наибольший интерес представляют предикаты сравнения и IS NULL. Теперь сравниваться могут не только типизированные значения, но также и так называемые строчные значения (*row value*), которые представляют собой упорядоченные последовательности значений допустимых в SQL-92 типов данных, задаваемые с помощью конструктора строчного значения (*row value constructor*)⁹.

Поскольку на множествах значений всех типов данных SQL-92 имеется отношение порядка, и элементы строчного значения упорядочены, определение результатов почти всех операций сравнения не вызывает затруднений: два строчных значения, которые должны быть одной и той же степени, сравниваются в лексикографическом порядке, и значение *unknown* вырабатывается в том и только в том случае, когда при очередном сравнении элементов строчных значений один или оба элемента являются null-значениями.

Не очень красивая ситуация имеет место в случае предиката IS NULL. Для строчного значения R условие R IS NULL принимает значение true тогда и только тогда, когда все элементы R являются null-значениями. Условие же R IS NOT NULL принимает значение true тогда и только тогда, когда ни один элемент R не является null-значением. Поэтому

⁸ Очень некорректно написано. Если тип данных – это множество представимых значений, для представления которых используются литеры, то единственным возможным толкованием оборота «элемент типа данных» является «элемент множества значений типа данных». Но это значение, для которого должен иметься литерал. А у null-значения литерала нет, значит оно не может являться элементом множества значений типа данных. Кроме того, никакой «элемент» не может быть элементом множества значений более одного типа данных.

⁹ Вообще-то, введение операции сравнения над строчными значениями без введения типа строчных значений само по себе некорректно. Одним из последствий отсутствия типа строчных значений является невозможность использования null-значения для обозначения неизвестного строчного значения, которое в этом случае представляется последовательностью null-значений той же степени, что и само строчное значение. В соответствии с этим, если строчное значение конструируется с использованием подзапроса, возвращающего не более одной строки, то случай пустого результата интерпретируется как наличие строки соответствующей степени, состоящей только из null-значений. Очевидно, что подзапрос мог бы явно выдать одну строку null-значений, и эти два случая неразличимы.

вопреки интуиции для R степени больше 1 $NOT (R IS NULL) \Leftrightarrow R IS NOT NULL$. Вот все возможные комбинации.

	R IS NULL	R IS NOT NULL	NOT R IS NULL	NOT R IS NOT NULL
degree 1: null	true	false	false	true
degree 1: not null	false	true	true	false
degree >1: all null	true	false	false	true
degree >1: some null	false	false	true	true
degree >1: non null	false	true	true	false

- 4) Условия поиска в SQL-92 строятся так же, как в SQL-89, но в основе построения логических выражений можно использовать не предикаты, но и явные проверки булевских значений (*boolean test*) вида $B IS [NOT] \{TRUE | FALSE | UNKNOWN\}$, где B – это предикат или условие поиска в круглых скобках. Эти операции всегда принимают значения *true* или *false* в зависимости от значения B. Это позволяет конструировать условия поиска, вычисляемые в двухзначной логике. Кроме того, введение обозначений логических значений можно считать шагом на пути к принятию полноценного булевого типа данных со своими литералами, хотя в SQL-92 такой тип так и не был введен.

2.1.4 SQL:1999 и SQL:200n

Ко времени написания этой статьи в 21-м веке были приняты пять вариантов стандарта SQL [18-23]. Объем стандарта все время увеличивается: если основная часть (*Foundation*) стандарта SQL:1999 [19] занимала 1150 стр., то в стандарте SQL:2016 [23] аналогичная часть насчитывает уже более 1700 стр. Однако все, что касается работы с неизвестными значениями и трехзначной логикой, появилось в SQL:1999 и переносилось в следующие версии стандарта практически без изменений. Суть соответствующих спецификаций состоит в следующем.

- 1) Начиная с SQL:1999, во всех версиях стандарта SQL содержится одно и то же определение null-значения, которое содержится в первой части стандарта (*Framework*) [18] и мало отличается от определения, приведенного в стандарте SQL-92; null-значение: специальное значение, используемое для обозначения отсутствия какого бы то ни было типизированного значения. Больше в стандарте SQL ничего не говорится о природе null-значений (не считая спецификации булевого типа, см. ниже), но из этого определения следует, что, во-первых, null-значение может помечать отсутствие значения любого типа данных и, во-вторых, само null-значение не является значением никакого типа данных.
- 2) Появился конструктор типов строчных значений. Как и следовало ожидать, строчный тип определяется как последовательность пар (<имя поля> <тип данных>). Возможность определения строчного типа влечет возможность определения в таблице базы данных столбца такого типа и, соответственно, возможность использования null-значения в случае отсутствия соответствующего строчного значения в этом столбце в некоторых строках таблицы (если это не запрещено ограничением NOT NULL данного столбца). Поэтому в спецификации предиката сравнения в SQL:1999 прежде всего говорится, что результатом сравнения двух строчных значений является *unknown*, если хотя бы один из операндов является null-значением.

Однако, как в SQL-92, во всех следующих версиях стандарта полагается, что если строчное значение конструируется с использованием подзапроса, и этот подзапрос выдает пустой результат, то образуется строчное значение, все элементы которого являются null-значениями. Заметим, что теперь (см. сноску 9 в п. 2.1.3) это (пустое) значение является типизированным, его строчный тип выводится из спецификации

подзапроса. Легко проверить, что предикат сравнения выдаст один и тот же результат *unknown* в обоих случаях. Более того, результатом применения IS NULL будет в обоих случаях *true*. Вообще, возможность наличия типизированного составного значения, все компоненты которого не типизированы, кажется очень подозрительной идеей.

- 3) В SQL:1999 появился очень странный булевский тип. Определение заслуживает дословного перевода.

Тип данных boolean состоит из двух различных истинностных значений true и false. Если это не запрещено ограничением NOT NULL, в типе данных boolean также поддерживается истинностное значение unknown как null-значение. В этой спецификации не делается различия между null-значением типа данных boolean и истинностным значением unknown, которое является результатом применения предиката, условия поиска или вычисления выражения с булевым значением; они могут использоваться равноправно, означая в точности одно и то же.

Все значения типа данных boolean и истинностные значения SQL являются взаимно совместимыми и присваиваемыми. Значение true больше, чем false; любое сравнение, в котором участвуют null-значение или истинностное значение unknown, приводит к результату unknown. Значения true и false могут быть размещены в любом месте, имеющем тип данных boolean; возможность размещения unknown, или null-значения зависит от того, допускается ли наличие null-значения в целевом объекте.

Почему нам не нравится булевский тип современных стандартов SQL? Во-первых, абсолютно непонятно, как можно отождествлять в булевском типе реальное значение *unknown* и специальное null-значение (псевдозначение). Null-значение обозначает отсутствие типизированного значения и не может поэтому являться типизированным значением.

Во-вторых, следствием этого неправомерного отождествления является отсутствие реальной упорядоченности значений булевского типа. В трехзначной логике третье логическое значение имеет естественную семантику *maybe*, в соответствии с которой $true > unknown > false$. Казалось бы, что либо на множестве значений типа данных есть отношение порядка, либо его нет. Странно, когда два из трех значения булевского типа упорядочены, а у третьего своего места нет.

Наконец, начиная с первых попыток Кодда расширить реляционную модель средствами представления отсутствующей информации, считалось, что если в любом выражении встречается null-значение, то результатом вычисления этого выражения является null-значение. Как мы покажем в следующем разделе статьи, часто это соглашение является мягко говоря сомнительным, но стандарт SQL его придерживается для всех разновидностей выражений, кроме булевских. Почему арифметическое выражение $0 \times NULL = NULL$, а логическое выражение $false \text{ AND } NULL = false$?

2.2 Представление отсутствующей информации без null-значений

Этот подраздел практически целиком опирается на работы Дейта и Дарвена. Для начала заметим, что, начиная с 1984 г. (до выхода первого стандарта SQL) [24], Дейт систематически, почти во всех своих публикациях критиковал различные аспекты языка SQL. По нашему мнению, эта критика очень полезна для многочисленных разработчиков приложений баз данных, использующих SQL на практике. Она же в большой степени мотивировала и стимулировала разработку Дейтом и Дарвеном их собственной современной версии реляционной модели данных [11]. Однако в этой статье нас интересуют идеи Дейта и Дарвена по поводу средства работы с отсутствующей информацией.

2.2.1 Значения по умолчанию

Впервые к теме неопределенных значений Дейт обратился в четвертом издании своей знаменитой книги «Введение в системы баз данных» [25, 5.5 “Null Values”]. Через три года

этот материал в немного переработанной форме был опубликован в виде отдельной статьи [26]. Наиболее важными моментами являются следующие.

- 1) В [25] говорится, что проблема null-значения недостаточно хорошо изучена, из-за чего включение поддержки null-значений в работающую систему следует считать преждевременным. В [26] Дейт уже строго заявляет, что использование null-значения для представления отсутствующей информации не является удовлетворительным решением этой проблемы. Концепция null-значения создает гораздо больше проблем, чем решает.
- 2) Тем не менее, далее приводится схема работы с null-значениями полностью в духе Котта [3] с той же путаницей между null-значением и третьим значением булевского типа. После этого следует критика с демонстрацией различных аномалий, к которым приводит наличие null-значения в языке SQL (Дейт ссылается на вариант SQL, использовавшийся в IBM и послуживший основой первых стандартов SQL). И, наконец, предлагается альтернативный подход к решению проблемы представления в базах данных отсутствующей информации на основе значений атрибутов отношений, назначаемых этим атрибутам по умолчанию.
- 3) В определении каждого атрибута любого именованного отношения присутствует раздел DEFAULT, в котором определяется значение по умолчанию этого атрибута, либо задается спецификация NODEFAULT, означающая отсутствие у атрибута значения по умолчанию. Для всех атрибутов, входящих в состав первичного ключа, обязательно отсутствие значений по умолчанию. Для всех прочих атрибутов при отсутствии раздела DEFAULT предполагается наличие предопределенного (зависящего от типа атрибута) значения по умолчанию (например, пробелов для символьных строк или нулей для числовых типов).
- 4) При вставке в отношение новой строки пользователь должен обеспечить явные значения всех атрибутов, у которых нет значений по умолчанию. Для других атрибутов при отсутствии явных значений система использует значения по умолчанию.
- 5) Имеется встроенная функция DEFAULT (R.A), возвращающая значение по умолчанию атрибута A базового отношения R. Если у атрибута A нет значения по умолчанию, фиксируется ошибка. С использованием этой функции можно корректно писать запросы, включающие вычисления агрегатных функций.

Как видно, в [25, 26] приводится лишь набросок предлагаемого подхода, и этот набросок повторяется спустя еще несколько лет в [27]. Поэтому [27] не способствует пониманию подхода Дейта, но в этой статье приводятся очень здравые соображения о невозможности определения трехзначного булевского типа при наличии универсального null-значения. Эти соображения созвучны тем, которые приводились в конце предыдущего подраздела по поводу ущербности булевского типа в современных стандартах SQL, и мы коротко их перескажем.

- 1) Если обозначить через UNK null-значение, обозначающее неизвестность данных, то результатом вычисления любого допустимого скалярного выражения $x \text{ alpha } y$, где alpha – бинарная операция, а x и y – скалярные значения, является UNK, если хотя бы одним операндом является UNK.
- 2) Результатом любого допустимого выражения $x \text{ theta } y$, где theta – операция скалярного сравнения, а x и y – скалярные значения, является третье логическое значение *unknown* (*unk*), если хотя бы одним операндом является UNK.
- 3) UNK и *unk* – это не одно и то же, т.е. истинностное значение *unknown* означает не то же самое, что «истинностное значение неизвестно». Если v – это переменная булевского типа, имеющая значение *unk*, то известно, что она имеет это значение. Это означает совсем не то же самое, что «значение v неизвестно».
- 4) Пусть для логических операций NOT, AND и OR действуют традиционные для трехзначной логики таблицы истинности. Предположим, что значение логической переменной неизвестно, т.е. обозначено через UNK. Можно согласиться, что результатом выражения *true* OR UNK может быть *true*, а результатом *false* AND UNK – *false*. Но

результатом операции NOT UNK может быть только UNK. Значит ли это, что в действительности логика должна быть четырехзначной?

В законченном виде описание подхода к представлению отсутствующей информации с помощью значений атрибутов по умолчанию появилось спустя еще два года [9]. В описании выделяются структурные аспекты предлагаемой схемы, аспекты, связанные с поддержкой целостности баз данных, и манипуляционные аспекты. В отличие от предыдущих публикаций, в [9] используется терминология стандарта SQL (*таблицы вместо отношений, столбцы вместо атрибутов, строки вместо кортежей*), а вместо DEFAULT используется более уместное в этом контексте UNK (UNKNOWN).

1) *Структурные аспекты.* В определении любого столбца каждой таблицы либо должен присутствовать раздел UNK, в котором задается представление UNK для этого столбца, либо должна присутствовать спецификация UNKS NOT ALLOWED. UNK-значение для данного столбца (если оно существует) должно являться значением домена этого столбца. Для некоторых столбцов может оказаться, что любая допустимая конфигурация битов является возможным реальным значением этого столбца (не UNK). По мнению Дейта, на практике такая ситуация будет возникать достаточно редко, и поэтому конкретные средства для определения UNK не предлагаются. В таких случаях придется явно вводить отдельные, контролируемые пользователями столбцы-индикаторы.

При вставке строки в базовую таблицу пользователь должен указать значение для любого столбца с UNKS NOT ALLOWED. Для остальных столбцов, если пользователь не указывает значение, система вставляет соответствующее UNK-значение.

При добавлении столбца к базовой таблице определение этого столбца должно содержать раздел UNK, и во всех существующих строках таблицы в новый столбец заносится соответствующее UNK-значение.

2) *Аспекты, связанные с поддержкой целостности.* От правила *целостности сущностей* (значения первичных ключей базовых отношений не должны содержать null-значений) можно полностью отказаться, поскольку нет null-значения. Из этого следует также, что исчезает различие между первичным ключом отношения и любым его возможным ключом.

Правило ссылочной целостности слегка упрощается: для каждого значения внешнего ключа должно существовать значение соответствующего целевого первичного ключа. Однако из этого упрощенного правила следует, что если значение внешнего ключа является UNK-значением, то в целевой таблице должна существовать строка, в которой значением первичного ключа также является UNK-значение.

3) *Манипуляционные аспекты.* Если R – это переменная диапазона, пробегающая по строкам некоторой базовой таблицы, а C – некоторый столбец этой таблицы, то встроенная функция UNK (R.C) возвращает UNK-значение этого столбца, если оно для него определено. Если столбец C определен с указанием UNKS NOT ALLOWED, фиксируется ошибка.

Встроенная функция IS_UNK (R.C) эквивалентна истинностному выражению $R.C = UNK$ (R.C), т.е. она возвращает *true* для тех и только тех строк таблицы, с которой связана переменная R, для которых значением столбца C является UNK-значение.

Встроенная функция IF_UNK (R.C, exp) является сокращенной формой выражения IF IS_UNK (R.C) THEN exp ELSE R.C.

С одной стороны, описанный подход действительно позволяет сохранять в базе данных информацию о неизвестных данных и обеспечивает пользователям возможность писать корректные запросы за счет явного выявления наличия UNK-значений. С другой стороны, UNK-значения – это допустимые значения доменов соответствующих столбцов. Пользователи могут включать в запросы арифметические выражения и сравнения, содержащие UNK-значения разных столбцов. Результаты таких выражений, очевидно, бессмысленны, как и результаты соответствующих запросов. Другими словами, система

полностью перекладывает на пользователей ответственность за корректность работы с отсутствующей информацией.

2.2.2 Специальные значения

Дейт обычно ссылается на [10], как на основную работу, описывающую его усовершенствованный подход к представлению отсутствующих данных на основе двухзначной логики. Однако в [28], где Дейт и Дарвен в соответствующих местах также ссылаются на [10], в действительности содержатся некоторые важные исправления и уточнения. Поэтому при рассмотрении сути этого подхода мы будем одновременно использовать и [10], и [28]¹⁰.

- 1) При определении любого (скалярного) типа данных¹¹ можно дополнительно к множеству его основных значений специфицировать любое число специальных значений¹², поведение которых отличается от поведения основных значений. Для каждого специального значения типа T задаются его идентификатор (например, UNK (*unknown*) или NA (*non-applicable*)), а также литеральные представления этих значений (например, '?' для UNK или '! ' для NA).¹³

Система не знает смысла специальных значений, их интерпретация известна только пользователям. Однако система знает, что эти значения являются не обычными, а специальными значениями типа T , и для каждого специального значения предопределяет операции UNK_T () и NA_T (), возвращающие значения T ('?') и T ('! ') соответственно.

Для каждого специального значения типа T также предопределяются операции вида IS_UNK (*scalar_exp*), где *scalar_exp* – выражение типа T . Операция возвращает истинностное значение *true*, если значением этого выражения является UNK_T (), и *false* в противном случае.

Наконец, предлагаются две операции внешнего соединения отношений (LEFT JOIN и RIGHT JOIN), которые похожи на соответствующие операции NATURAL OUTER JOIN SQL, но вместо null-значений используются явно указываемые специальные значения.

¹⁰ Хотя при этом возникают трудности, связанные с разным контекстом: в [10] используется контекст стандарта SQL, а в [28] – контекст определяемого варианта реляционной модели данных. В своем изложении мы будем стараться сгладить различия этих контекстов.

¹¹ В [28] под *скалярным типом данных* понимается инкапсулированный тип, *реальные представления* значений которого скрываются от пользователей за одним или несколькими *возможных представлений*, каждое из которых является некоторой структурой (записью) с типизированными полями (ранее определенных или предопределенных типов), взаимно однозначно отображаемой на реальное представление типа его реализатором. Операции над значениями скалярного типа определяются в терминах одного из его возможных представлений и автоматически отображаются в операции над его реальным представлением.

¹² Именно этим объясняет Дейт в [10] переход от термина *значение по умолчанию* к термину *специальное значение*: специальное значение не может служить в качестве значения по умолчанию, потому что в одном типе может быть много специальных значений. Однако, по нашему мнению, не менее, а скорее более важным отличием специальных значений от значений по умолчанию является то, что в [9] значением столбца по умолчанию служит выбираемое пользователем обычное значение домена (типа данных), а в [10, 28] специальные значения являются явно вводимыми дополнительными значениями, про которые системе известно, что каждое из них отличается как от любого обычного значения, так и любого другого специального значения того же типа.

¹³ Сравнительно подробно схема специальных значений обсуждается только для скалярных типов данных, хотя в [28] вводятся еще две разновидности типов – кортежные типы (похожие на строчные типы SQL) и типы отношений, основанные на кортежных типах. Типы данных этих разновидностей являются безымянными, так что напрямую подход, применяемый для именованных скалярных типов, для них не работает. В одной из сносок в [28] мимоходом говорится, что нужно было бы что-то придумать по этому поводу, но до этого дело не дошло.

2) Для любых двух значений a и b (обычных и специальных) типа T определены операции сравнения по равенству и неравенству. Сравнение $a = b$ возвращает значение *true* в том и только в том случае, когда a и b являются одним и тем же значением. Сравнение $a \neq b$ возвращает значение *true* в том и только в том случае, когда a и b не являются одним и тем же значением. Однако при попытке определить на основе двухзначной логики, например, операции сравнения $a > \text{UNK}_T()$ и $a \leq \text{UNK}_T()$ как возвращающие *false* для любого значения a , отличного от $\text{UNK}_T()$, возникает противоречие. Действительно, предположим, что $(a > \text{UNK}_T()) = \text{false}$. Тогда $(a \leq \text{UNK}_T()) = \text{NOT}(a > \text{UNK}_T()) = \text{NOT}(\text{false}) = \text{true}$.

В результате в [10] Дейт приходит к выводу, что при использовании типов со специальными значениями сравнение значений a и b на $>, <, \geq, \leq$ допускается в тех и только тех случаях, когда a и b являются обычными значениями. Если хотя бы один из операндов такой операции сравнения является специальным значением, возникает исключительная ситуация. Заметим, что в [28] операции сравнения на $>, <, \geq, \leq$ для типов со специальными значениями уже не упоминаются.

3) Вычислительные выражения со значениями типов со специальными значениями никогда не приводят к результатам, являющимся специальными значениями. В частности, отсутствует предписание, в соответствии с которым $a + \text{UNK}_T() = \text{UNK}_T()$ для любого значения a типа T . Это мотивируется следующими соображениями. Если $a + \text{UNK}_T() = \text{UNK}_T()$, то из соображений согласованности должно быть и $a - \text{UNK}_T() = \text{UNK}_T()$. Но тогда должно выполняться и $\text{UNK}_T() - \text{UNK}_T() = \text{UNK}_T()$, что, по мнению Дейта, может привести к аномалиям, похожим на аномалии null-значений в SQL.

Судя по всему, Дейт и Дарвен были не слишком удовлетворены своей схемой специальных значений (мы вернемся к ней в следующем разделе статьи), потому что в последнем издании их книги о современном представлении реляционной модели данных [11] вообще отсутствуют предложения по поводу средств работы с отсутствующей информацией.

2.2.3 Сохранение в базе данных отсутствующей информации без поддержки null- или специальных значений

Однако спустя четыре года после публикации [11] Дейт и Дарвен выпустили книгу [29], в которой отдельная часть из четырех глав посвящена различным подходам, позволяющим сохранять и использовать в базах данных отсутствующую информацию без использования каких-либо средств явного представления такой информации. Эта часть книги написана Хью Дарвенем, который во введении отмечает, что не стремится отстаивать достоинства ни одной из описываемых схем, а публикуются они в целях их обсуждения.

Первая из описываемых в [29] схем основывается на *декомпозиции отношений*. Идея принадлежит Дарвену и была изначально представлена в [30].

1) Предположим, что имеется переменная отношения $r \{key, a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$, в любом допустимом значении которой информация может отсутствовать только в атрибутах b_1, b_2, \dots, b_n . Тогда на первом шаге r вертикально декомпозируется на $m + 1$ переменных отношений $ra \{key, a_1, a_2, \dots, a_n\}$, $rb_1 \{key, b_1\}$, ..., $rb_m \{key, b_m\}$, в допустимых значениях которых информация может отсутствовать только в атрибутах b_i ($i = 1, \dots, m$).

2) На втором шаге каждая переменная отношения $rbi \{key, b_i\}$ горизонтально декомпозируется на переменные отношений $rbi \{key, b_i\}$, $rbi1 \{key\}$, ..., $rbi k \{key\}$ такие, что в допустимых значениях rbi атрибут b_i содержит обычные значения, а в допустимых значениях $rbi1$ содержатся кортежи, в которых значению ключа соответствует отсутствие информации в атрибуте b_i вида 1 (например, значение b_i неизвестно), ..., в допустимых значениях $rbi k$ содержатся кортежи, в которых значению

ключа соответствует отсутствие информации в атрибуте bi вида k (например, значение bi неприменимо).

- 3) Чтобы можно было «склеить» значения переменных отношений $ra, rbi, rbi1, \dots, rbik$ ($i = 1, \dots, m$), нужно, чтобы поддерживалось специальное ограничение целостности: для каждого значения ключа ra должно найтись совпадающее с ним значение ключа в значении ровно одной переменной $rbi, rbi1, \dots, rbik$.
- 4) Запросы над такими декомпозированными базами данных с неявно содержащейся отсутствующей информации можно выполнять с использованием переименования атрибутов, добавления столбцов, заполнения их определяемыми пользователями обозначениями отсутствующих данных и выполнения операций естественного соединения.

На маленьком примере, приведенном Дарвенем, все это выглядит нестрашно, но для отношений с большим числом атрибутов, в которых могут отсутствовать данные, по нашему мнению, практически неприменимо.

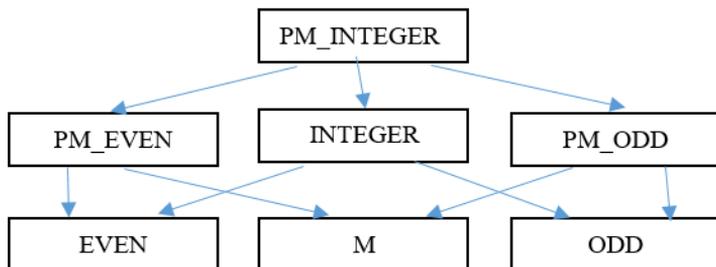
Второй подход основан на понятии *мультиотношения*. В некотором смысле этот подход похож на подход на основе декомпозиции, поскольку в хранимых кортежах отсутствуют атрибуты с отсутствующими данными, но позволяет обойтись без порождения многочисленных переменных отношений и без потребности в их соединении при выполнении запроса. Приводится только набросок этой схемы, но и он слишком объемён, чтобы можно было описать его подробно в этой статье. Ограничимся очень кратким изложением основных идей.

- 1) Как и у обычных отношений, у каждого мультиотношения mr есть заголовок MRH и тело MRB . Однако в тело mr могут входить все кортежи, соответствующие любому подмножеству MRH (включая пустое подмножество), т.е. кортежи mr могут иметь степени от 0 до n , где n – степень заголовка.
- 2) Участником мультиотношения mr называется отношение p с заголовком PH , являющимся некоторым подмножеством MRH , и телом PB , включающим все кортежи из MRB , которые соответствуют PH . Тем самым, у любого мультиотношения степени n имеется 2^n участников, некоторые из которых могут быть пустыми.
- 3) Для переменной мультиотношения MR можно определять ограничения целостности. Обязательным ограничением любой MR является ограничение ключа, в котором задается такое подмножество MRH $\{a1, a2, \dots, an\}$, что все атрибуты ai участвуют в каждом кортеже любого допустимого в MR значения mr и во всех кортежах составные значения $\{a1, a2, \dots, an\}$ различны.
- 4) Определяется набор алгебраических операций над мультиотношениями, часть которых является прямым переложением определений реляционных операций в терминах мультиотношений. Однако определение некоторых операций настолько же неочевидно, как и в схеме специальных значений (см. п. 2.2.2). К числу таких операций относится совершенно необходимая операция ограничения (фильтрации) мультиотношений. В условии ограничения mr могут входить имена атрибутов, не входящих в некоторые кортежи mr , и чтобы решить проблему вычисления условия для этих кортежей без потребности в трехзначной логике, предлагается в обязательном порядке использовать предикаты PRESENT и ABSENT, являющиеся прямыми аналогами IS [NOT] NULL в SQL.

Дарвен полагает, что на основе схемы мультиотношений можно было бы разработать новый язык баз данных, который позволял бы обращаться с таблицами SQL как с мультиотношениями, полагая что в строке таблицы, в которой некоторый столбец содержит null-значение, этот столбец можно считать отсутствующим. Однако, по нашему мнению, это не принесет особых преимуществ, поскольку существующий SQL и так позволяет избежать трехзначной логики, если в каждом условии использовать IS [NOT] NULL.

Следующий подход, обсуждаемый в [29], опирается на использование механизма наследования. Идея этого подхода принадлежит Эрвину Смауту [31]. В детали мы здесь вдаваться не будем, поскольку для этого потребовалось бы описать суть необычного и нетривиального (хотя очень логичного) механизма наследования типов данных, определенного в [11].

- 1) Суть идеи состоит в том, что в корневом супертипе любой иерархии наследования типов¹⁴ можно определить, кроме множества обычных значений этого типа, значение *MISSING*. После этого можно определить любой подтип в двух вариантах – Т и PM_Т, в первом из которых содержатся только обычные значения, а во втором – еще и значение *MISSING*. Вот пример графа подтипизации, в котором определяются типы PM_INTEGER (все целые числа плюс *MISSING*), INTEGER (все целые числа), PM_EVEN (четные целые плюс *MISSING*), EVEN, PM_ODD (нечетные целые плюс *MISSING*), ODD, а также тип M, включающий единственное значение *MISSING*¹⁵.



- 2) Значение *MISSING* не может участвовать в выражениях любого типа, которому принадлежит *MISSING*. Допускается сравнение по равенству и неравенству всех значений любого типа, включающего *MISSING*, с очевидной семантикой на основе интерпретации *MISSING* как неизвестного значения. Сравнения на $>$, $<$, \geq , \leq с участием значения *MISSING* запрещаются (исключительная ситуация?).

Все это дает основания полагать, что схема на основе наследования не дает ничего такого, чего бы не обеспечивала схема специальных значений (в [29] к этому выводу приходит сам Дарвен).

Наконец, последний подход основан на использовании *атрибутов со значениями-отношениями*. Основная идея состоит в том, чтобы использовать пустое множество в качестве значения атрибута, реальное значение которого отсутствует. Более конкретно, предлагается следующее.

- 1) Если требуется определить в базе данных переменную отношения $R \{a1:T1, a2:T2, \dots, an:Tn\}$ такую, что в допустимых значениях r переменной R у некоторого атрибута ai в некоторых кортежах r могут отсутствовать значения, то в действительности определяется переменная отношения $ZR \{a1:T1, a2:T2, \dots, zai:RELATION \{ai:T\}, \dots, an:Tn\}$. Другими словами, RH отличается от ZRH тем, что типом (переименованного) i -го атрибута становится тип унарного отношения с атрибутом ai типа T . При этом в любом кортеже любого допустимого значения zr переменной ZR значение-отношение атрибута zai не должно иметь мощность больше 1¹⁶. Такие отношения степени 1 и мощности не большей 1 Дарвен называет *ZOO*-отношениями (cardinality Zero or One, degree One).

¹⁴ На самом деле, это не совсем иерархия, поскольку в [11] поддерживается множественное наследование: тем не менее, у любой такой «иерархии» имеется корневой супертип.

¹⁵ Тип M появляется в графе наследования, поскольку в применяемой модели наследования каждое значение супертипа должно являться значением хотя бы одного его непосредственного подтипа.

¹⁶ Это ограничение переменной отношения ZR .

2) Если в условии и/или результате запроса к zr участвовали значения атрибута ai , то (1) из zr удаляются все кортежи, с пустыми значениями атрибута zai ¹⁷; (2) в добавляется атрибут $ai: T$, и в каждом кортеже получаемого отношения $zr1$ значением ai становится значение атрибута ai в единственном кортеже значения-отношения атрибута zai того же кортежа $zr1$; (3) из $zr1$ удаляется атрибут zai , и над получаемым отношением обычным образом выполняется фильтрация и формирование результирующего отношения.

Подводя итог этого пункта, явно заметим, что только первый из четырех рассмотренных подходов предлагает решение, реально выходящее за пределы схемы специальных значений, но приводящее в общем случае к нереалистично большому числу переменных отношений в базе данных, усложнению и потенциальному замедлению выполнения запросов. Каждый из трех последних подходов логичен и красив, но по сути это переложение идей специальных значений.

3. Типизированные неизвестные значения и трехзначная логика

Как показывает разд. 2, в настоящее время отсутствует удовлетворительное решение проблемы поддержки в базах данных отсутствующей информации. Несмотря на многолетние попытки решить эту проблему подход, применяемый в SQL-ориентированных СУБД на основе стандарта SQL, и схемы, предлагавшиеся Дейтом и Дарвенном, обладают, на наш взгляд, существенными недостатками. В этом разделе мы предпринимаем еще одну попытку предложить подход к решению проблемы отсутствующей информации. Но сначала обсудим причины неудач предыдущих подходов, что требуется для мотивации наших собственных идей.

3.1 Почему имеющиеся подходы неудачны?

Подходы, идущие от Кодда и приведшие к схеме, которая используется в современном SQL, и предлагавшиеся Дейтом и Дарвенном, неудачны по-разному, и причины этих неудач нуждаются в раздельном обсуждении.

3.1.1 Тупик на пути Кодда

Корень зла кроется, конечно, в понятии универсального null-значения, которое не является значением ни одного типа данных, но может использоваться вместо любого значения любого типа данных в любом месте, где могло бы появиться это значение. Конечно, при таком определении null-значения естественно полагать, что результатом вычисления любого выражения, в котором вместо реального значения встречается NULL, является NULL.

Однако, не говоря уже о более серьезных последствиях этого подхода, которые мы обсудим немного позже, сам факт разрешения использовать в выражениях чего-бы то ни было, кроме значений типа данных этого выражения, является парадоксальным и неприемлемым. Конечно, эту ситуацию никак не исправляет отказ Кодда в его поздних работах от термина *null-значение* с его заменой на термин *маркер*.

Но даже если закрыть глаза на теоретическую недопустимость использования null-значения в типизированных выражениях, правила вычисления выражений, содержащих универсальное null-значение, нарушают принцип замены неопределенных значений самого Кодда [3]. Этот принцип почему-то формулируется только для операций сравнения (операция сравнения выдает третье логическое значение *unknown* в том и только в том случае, когда при замене всех вхождений null-значения одним набором реальных значений оно выдает *true*, а при их замене другим набором реальных значений – *false*).

¹⁷ Конечно же, на основе предиката IS_EMPTY.

Очевидно, что аналогичный принцип должен использоваться при вычислении всех видов операций: операция выдает null-значение в том и только в том случае, когда при замене всех вхождений null-значения разными наборами реальных значений она может выдавать разные значения. Тривиальным примером является арифметическое выражение $0 \times \text{NULL}$, значением которого, конечно, должен быть 0, а не NULL. Но для этого требуется, чтобы в контексте вычисления арифметического выражения null-значение интерпретировалось не как универсальный маркер отсутствующего значения, а как значение соответствующего числового типа (хотя и неизвестное).

Зато принцип замены неопределенных значений прекрасно работает для булевских операций: операция *true* OR NULL, очевидно, должна выдавать *true*, *false* OR NULL = NULL, NOT NULL = NULL и т.д. Другими словами, при следовании этому принципу для булевого типа данных универсальное null-значение ведет себя точно так же, как третье логическое значение *unknown*. При этом к самому принципу замены претензий нет, он полностью логичен.

Таким образом, мы видим, что появление в современных стандартах SQL неполноценного трехзначного булевого типа данных (см. конец п. 2.1.4) вполне закономерно, и единственной причиной этой неполноценности является поддержка единственного и универсального null-значения. Разработчики стандарта SQL:1999 [19] и более поздних стандартов справедливо хотели оснастить стандарт современной системой типов данных. Обойтись без булевого типа, значения которого можно сохранять в базах данных, было невозможно. Но тогда из-за универсальности null-значения приходится допустить наличие в базе данных неизвестных булевских значений, обозначаемых null-значением, что, в свою очередь, приводит к недопустимому отождествлению значения булевого типа *unknown* и null-значения, которое является лишь обозначением отсутствия реального значения. По-другому просто не получается.

Еще одним следствием ущербности булевого типа SQL является потеря упорядоченности булевских значений в трехзначной логике. Было бы логично полагать, что $\text{false} < \text{unknown} < \text{true}$. Если сопоставить логическим значениям *false*, *unknown*, *true* числа 0, 0.5, 1 в том же порядке, то логические операции выражаются через численные операции: NOT $x = 1 - x$; x AND $y = \min(x, y)$, x OR $y = \max(x, y)$. Из-за отождествления *unknown* и null-значения неизвестное логическое значение теряет свою позицию.

Можно сказать, что булевский тип в стандарте SQL – это не только не настоящий трехзначный булевский тип, но и не тип данных вообще. С нашей точки зрения, это полный тупик в многолетней истории попыток решить проблему отсутствующей информации на основе универсального null-значения.

В заключение этого пункта сделаем несколько замечаний по поводу последней схемы Кодда с двумя маркерами (A-маркер *a* для неизвестных значений и I-маркер *i* для неприменимых значений) и четырехзначной логикой (см. п. 2.1.2).

Во-первых, совершенно непонятны правила вычисления арифметических выражений

$$\begin{array}{cccc} z \% a = a & z \% i = i & a \% z = a & i \% z = i \\ a \% a = a & a \% i = i & i \% a = i & i \% i = i \end{array},$$

в которых маркер неизвестных значений ведет себя совершенно так же, как и маркер неприменимых значений. Если, например, можно согласиться с тем, что $z + a = a$, то никак нельзя принять, что $z + i = i$ (из последнего следует, например, что нельзя посчитать месячный доход служащего, которому не полагается премия).

Во-вторых, аналогичные сомнения применимы и к правилам определения результатов операций сравнения¹⁸

$$z \Theta a = a_B \quad z \Theta i = i_B \quad a \Theta z = a_B \quad i \Theta z = i_B$$

¹⁸ Для справедливости еще раз повторим, что Кодд в явном виде эти правила не определял. Они построены автором данной статьи «по образу и подобию» других конструкций Кодда.

$$a \ominus a = a_B \quad a \ominus i = i_B \quad i \ominus a = i_B \quad i \ominus i = i_B \quad ,$$

в соответствии с которым четвертое логическое значение i_B формируется точно так же, как и третье логическое значение a_B (*unknown*). Но если мы будем продолжать понимать *unknown* в смысле *maybe*, то i_B должно означать *not-maybe*. Можно согласиться с тем, что $z > a = a_B$, но никак нельзя принять, что $z > i = i_B$ (наверное, это следует понимать так: не может существовать служащий с размером зарплаты, большим размера премии, если этот служащий вообще не получает премию). Да и то, что, $i \ominus i = i_B$, вызывает большие сомнения.

Далее, еще в 1990-м г. Джин Гессерт [32] обратил внимание на то, что определение Коддом операции отрицания, в соответствии с которым $\text{NOT}(a_B) = a_B$ и $\text{NOT}(i_B) = i_B$, приводит к тому, что в его четырехзначной логике перестают действовать законы де Моргана. В частности, $i_B = \text{NOT}(a_B \text{ AND } i_B) = \text{NOT}(a_B) \text{ OR } \text{NOT}(i_B) = a_B \text{ OR } i_B = a_B$.

В [33] Дейт анализирует всю историю четырехзначного подхода Кодда и отмечает, что в ней имелось три итерации. После публикации в [8, 13] первого варианта этой схемы, в которой $\text{false OR } i_B = i_B \text{ OR } \text{false} = \text{false}$, во втором тираже [13] таблица истинности для OR была изменена так, что $\text{false OR } i_B = i_B \text{ OR } \text{false} = i_B$ (как замечает, Дейт, это по сути ничего не изменило). Наконец, по словам Дейта [33], впоследствии Кодд еще раз пересмотрел свои таблицы истинности, определив $\text{NOT}(a_B) = i_B$ и $\text{NOT}(i_B) = a_B$ (соответствующая публикация, скорее всего, сделана не была). После этого законы де Моргана заработали. Кроме того, если обозначить логические значения *false*, i_B , a_B , *true* числами $0, \frac{1}{3}, \frac{2}{3}, 1$ в том же порядке, то логические операции, как и в трехзначной логике, выражаются через численные операции: $\text{NOT } x = 1 - x$; $x \text{ AND } y = \min(x, y)$, $x \text{ OR } y = \max(x, y)$.

Однако, даже если смириться со всеми неясностями и странностями подхода с двумя маркерами и четырехзначной логикой, возникает та же проблема, что и при наличии одного null-значения (только еще более неприятная), вызываемая универсальностью и общеупотребимостью A-маркера и I-маркера. Если разрешить сохранять логические значения в базах данных (т.е. разрешить включение в отношения атрибута типа BOOLEAN), то естественно допустить, что в некоторых кортежах значения этого атрибута может быть неизвестным, а в некоторых – неприменимым.

Это приводит к необходимости определения результатов операций $\text{NOT}(a)$, $\text{NOT}(b)$, $\text{true AND } a$, $\text{true AND } b$ и т.д. Напрягая воображение, можно согласиться с тем, что двуместные логические операции с операндами a и b ведут себя так же, как соответствующие операции с операндами a_B и i_B , т.е., например, $\text{true AND } a = a$, $\text{true AND } b = b$, но никакое воображение не может допустить, что $\text{NOT}(a) = b$. Другими словами, чтобы можно было отождествить в контексте булевского типа логические значения a_B и i_B с маркерами a и b соответственно, нужно вернуться к исходному определению Коддом операции отрицания ($\text{NOT}(a_B) = a_B$ и $\text{NOT}(i_B) = i_B$), которое приводят к тому, что законы де Моргана не работают.

Вывод состоит в том, что схема Кодда с двумя маркерами и четырехзначной логикой не позволяет определить даже ущербный булевский тип в духе стандарта SQL.

3.1.2 Рыцари двухзначной логики

Начиная с 1984-го года, в бесчисленных публикациях Дейт сначала в одиночку, а потом вместе с Дарвенем критиковал Кодда и язык SQL за использование null-значения и трехзначной логики. Практически всегда эта критика была обоснованной с приведением примеров запросов, которые выдают противоречащие интуиции результаты. Одновременно Дейт и Дарвен пытались придумать решение проблемы отсутствующей информации с использованием традиционной двухзначной логики. Однако, по нашему мнению, им так и не удалось изобрести схему, которая могла бы конкурировать с плохо обоснованной, логически

противоречивой, но пригодной для практического использования схемой работы с отсутствующей информацией стандарта языка SQL.

В действительности, наличие трехзначной логики не слишком беспокоит пользователей SQL-ориентированных СУБД, поскольку возможность включать в выражения null-значение и интерпретация результата *unknown*, возникающего при вычислении условных выражение разделов WHERE и HAVING запросов, как запрещающего, т.е. эквивалентного *false* по своему действию, позволяет в большинстве случаев формулировать интуитивно понятные запросы, выдающие результаты, также соответствующие интуиции. В сомнительных случаях при построении условий выборки можно использовать предикат IS [NOT] NULL, позволяющий избежать null-значения в выражениях и, тем самым, остаться в рамках двухзначной логики.

Кратко подытожим плюсы и минусы подходов Дейта и Дарвена, обсуждавшихся в подразделе 2.2. Очевидным минусом всех подходов (с нашей точки зрения) является перекладывание на пользователей учета наличия отсутствующих данных в отношениях, к которым адресуются запросы. Поскольку пользователи, формулирующие запросы, не обязаны знать, содержат ли все атрибуты во всех кортежах отношений только реальные данные или же в некоторых кортежах значения некоторых атрибутов отсутствуют, все запросы должны содержать условия типа IS [NOT] UNK и явно описываемые операции, которые нужно выполнить, если такое условие принимает значение *true*.

Не будем здесь повторять, чем нас не устраивает подход значений по умолчанию (см. конец п. 2.1.1). Но вот у следующего подхода, основанного на расширении типов данных специальными значениями, имеется очень серьезный потенциал, не используемый его создателями. Главными чертами этого подхода является то, что (а) специальные значения не универсальны и (б) для каждого типа данных можно определить свои специальные значения. В принципе можно было позволить системе или пользователям определять поведение специальных значений в операциях соответствующего типа данных.

Но даже если ограничиться только одним специальным значением UNK_T () (см. п. 2.2.2), в двухзначной логике, которой принципиально придерживаются Дейт и Дарвен, определить эти операции оказывается невозможно. В соответствии с семантикой UNK_T () (*maybe*), $a - UNK_T () = UNK_ () - a = UNK_T () - UNK_T () = UNK_T ()$, $a \theta UNK_T () = UNK_T () \theta a = UNK_T () \theta UNK_T () = UNK_BOOLEAN ()$ ¹⁹ для каждого неспециального значения *a* типа T. Как видно, здесь UNK_BOOLEAN () – это специальное значение, добавленное к двухзначному булевскому типу. Очевидно, что попытки определить поведение этого специального значения в операциях NOT, AND и OR приводят нас к традиционной трехзначной логике, в которой UNK_BOOLEAN () играет роль третьего логического значения.

Но поскольку трехзначная логика запрещена, Дейт полностью запрещает и использование специальных значений в выражениях и операциях сравнения для всех видов сравнения, кроме = и ≠. Но даже при таких ограничениях кажется сомнительным, что UNK_T () = UNK_ (), а $a \neq UNK_T ()$, $UNK_ () \neq a$ для каждого неспециального значения *a* типа T. В результате единственным способом формулировать запросы к базе данных, атрибуты отношений которой могут содержать специальные значения типа T, является явное использование предиката IS_UNK.

Как мы отмечали в конце п. 2.2.3, из четырех подходов, предложенных Дарвеном после схемы специальных значений, только в первом (подход с декомпозицией отношений) специальные значения явно или неявно не используются. Вообще, этот подход кажется нам перспективным, если не замахиваться на поддержку произвольного числа причин отсутствия

¹⁹ Мы умышленно используем обозначения Дейта, чтобы показать ход рассуждений, который можно было применить на его месте.

данных в атрибутах отношений. В общем виде, представленном в [29], он не кажется практически применимым.

3.2 Если бы губы Никанора Ивановича да приставить к носу Ивана Кузьмича

Собственно, это мы и пытаемся сделать в этом подразделе, а именно, мы скрещиваем специальные значения Дейта с трехзначной логикой Кодда и, отчасти, с подходом Дарвена с декомпозицией отношений. Конечно, для этого нужно многое пересмотреть в обоих подходах, но и получаемые преимущества выходят за пределы простого объединения положительных черт каждого из этих подходов.

В предлагаемой схеме мы ограничиваемся только одной разновидностью специальных значений с семантикой *maybe*. Этому есть несколько объяснений:

- чаще всего точные данные невозможно занести в базу данных именно потому, что они неизвестны;
- семантика неизвестных данных кажется нам значительно понятнее, чем семантика неприменимых данных (*non-available*, или *not-maybe*);
- мы не хотим выходить за пределы трехзначной логики;
- в дополнение к основной схеме мы предлагаем использовать идеи Дарвена, что в случае потребности справиться с неприменимыми данными без потребности в четырехзначной логике;
- наконец, прочие разновидности причин отсутствия данных кажутся нам надуманными и/или редко встречающимися.

При этом мы хотим добиться следующих целей:

- сделать специальные значения полноценными и полноправными значениями соответствующих типов данных с возможностью применения к ним всех операций с поддержкой интуитивной семантики, вообще говоря, индивидуальной для каждого типа данных;
- сделать трехзначный булевский тип полноценным и полноправным типом данных за счет интерпретации специального значения, расширяющего двухзначный булевский тип, как третьего логического значения трехзначного булевского типа.

Обсудим, каким образом можно достичь этих целей. Поскольку, в конечном счете, мы стремимся «облагородить» механизмы null-значений и трехзначной логики стандарта SQL, будем использовать терминологию SQL²⁰.

- 1) Множество значений {t} каждого допустимого в стандарте SQL типа данных T расширяется специальным значением *t_ukn*, отличным от любого t. Для предопределенных и конструируемых типов такое расширение с определением семантики операций, включающих операнды-специальные значения, можно было бы предопределить. Для типов данных, определяемых пользователями, это должны делать разработчики соответствующих новых типов данных.
- 2) Операции над значениями расширенных типов должны быть уточнены с учетом семантики основных значений и возможности использования специального значения в

²⁰ Это не означает, что описываемые предложения можно напрямую использовать для переделки стандарта SQL. Стандарт SQL при всех его недостатках – это гигантская, тщательно продуманная и сбалансированная спецификация, в которой изменения в одной части неминуемо влекут изменения в других частях (скорее всего, во многих других местах). И, конечно, это в полной мере относится к старейшему унаследованному механизму null-значений. Поэтому мы даже не будем пытаться формулировать все технические проблемы, с которыми пришлось бы столкнуться при замене этого механизма на нечто, основанное на наших идеях.

качестве операнда. Например, для всех числовых типов данных, очевидно, $0 \times t_ukn = t_ukn \times 0 = 0$, а для типов множеств $\{\emptyset\}$ $INTERSECT t_ukn = t_ukn INTERSECT \{\emptyset\} = \{\emptyset\}$.

Для типов символьных и битовых строк, а также для типов массивов результатом конкатенации обычного значения со специальным значением в действительности должен являться первый операнд, за которой следует специальное значение. Для такой «строки» можно было бы выполнять операции взятия или поиска позиции подстроки, не обязательно получая в результате t_ukn соответствующего типа.

- 3) Двухзначный булевский тип также расширяется специальным значением, которое в силу его повышенной важности мы будем называть особым образом – *unknown*. По определению специальное значение *unknown* является результатом выполнения любой операции сравнения, если эта операция не вырабатывает обычное булевское значение *true* или *false*. Для множества значений расширенного булевского типа вводится отношение порядка $false < unknown < true$.

Естественным образом с учетом семантики двухзначного булевского типа и специального значения *unknown* доопределяется набор операций расширенного булевского типа:

$$NOT\ unknown = unknown,$$

$$(unknown\ AND\ unknown) = (true\ AND\ unknown) = (unknown\ AND\ true) = unknown,$$

$$(false\ AND\ unknown) = (unknown\ AND\ false) = false,$$

$$(true\ OR\ unknown) = (unknown\ OR\ true) = true,$$

$$(unknown\ OR\ unknown) = (false\ OR\ unknown) = (unknown\ OR\ false) = unknown.$$

Другими словами, $\{false, unknown, true\}$ – это множество значений полноценного трехзначного булевского типа и при этом *unknown* остается специальным значением этого типа с семантикой *maybe*.

- 4) Хотя, вообще говоря, результатом любой операции сравнения $t_ukn \theta t_ukn, t \theta t_ukn, t_ukn \theta t$, где $\theta = =, \neq, >, \geq, <, \leq$, а t – обычное значение некоторого типа T , является *unknown*, в операциях сравнения должны учитываться табличные ограничения тех таблиц, из столбцов строк которых выбираются операнды. Например, если для столбца целого типа установлено ограничение $value > 0$, и в некоторых строках соответствующей таблицы значением этого столбца является t_ukn , то для этих строк результатом сравнения $t_ukn > 0$, должно являться *true*. Аналогично, пусть в таблице имеются два столбца $C1$ и $C2$ одного и того же типа, и уставлено табличное ограничение $CHECK\ C1 > C2$. Тогда, если в некоторых строках этой таблицы значением столбца $C2$ является t_ukn , то для этих строк результатом сравнения $C1 > t_ukn$, должно являться *true*.
- 5) Условие выборки запроса к базе данных, вычисляемое для очередной строки (группы строк) таблицы, является разрешающим в том и только в том случае, когда результатом вычисления является *true*. Чтобы можно было формулировать запросы, направленные на выборку строк, которые содержат специальные значения, для каждого типа данных T определяется операция $IS\ t_ukn(T_exp)$, где T_exp – выражение типа T , результатом которой является *true*, если результатом T_exp является t_ukn , и *false* в противном случае.
- 6) В SQL имеются две полезные конструкции *SCALAR SUBQUERY* и *ROW SUBQUERY* с неочевидной семантикой.

SCALAR SUBQUERY – это вложенный подзапрос, результатом которого должна являться таблица SS степени 1, содержащая ровно одну строку. По правилам SQL результат скалярного подзапроса неявно преобразуется в значение единственного столбца единственной строки таблицы SS . При этом, если мощность SS оказывается больше 1, возникает исключительная ситуация, а если мощность SS равна 0, то выдается NULL.

Точно так же ведет себя операция *ELEMENT*, извлекающая ровно одно значение из заданного мультимножества, мощность которого должна быть равна 1.

Можно понять причины, по которым в стандарте SQL были приняты такие правила, поскольку скалярные подзапросы могут участвовать в выражениях в разных местах SQL-запроса. Даже неочевидное назначение NULL в качестве значения подзапроса с пустым результатом можно оправдать, поскольку в SQL NULL – это универсальное обозначение отсутствия данных. Но в предлагаемой нами схеме совершенно недопустимо считать, что результатом пустого скалярного подзапроса является специальное значение типа столбца таблицы SS.

Похожим образом ведет себя *ROW SUBQUERY* – вложенный подзапрос, результатом которого должна являться таблица RS, содержащая ровно одну строку. По правилам SQL результат строчного подзапроса неявно преобразуется в строчное значение единственной строки таблицы RS. Тип этого значения и типы его элементов определяются списком выборки строчного подзапроса. При этом, если мощность RS оказывается больше 1, возникает исключительная ситуация, а если мощность RS равна 0, то выдается строка той же степени, что и степень RS, каждым элементом которой является NULL. По-видимому, так же ведет себя операция *ELEMENT*, извлекающая ровно одно строчное значение из заданного мультимножества мощности 1 с элементами строчного типа.

Опять же, эти правила SQL можно понять, поскольку строчные подзапросы могут участвовать в операциях сравнения. Даже неявное отождествление NULL и строки, составленной из NULL, можно (хотя и с трудом) объяснить универсальностью null-значения в SQL. Но в предлагаемой нами схеме совершенно недопустимо считать, что результатом пустого строчного подзапроса является строка, составленная из специальных значений типов элементов строчного типа. Во-первых, такая строка могла бы появиться и в том случае, когда мощность RS равна 1. Во-вторых, строка, составленная из специальных значений типов элементов строчного типа, является обычным значением этого строчного типа, у которого имеется собственное специальное значение.

Мы видим два возможных выхода из этого положения. Самым простым решением является генерация исключительной ситуации в случаях, когда мощность SS, RS или мультимножества, к которому применяется операция *ELEMENT*, оказывается не равной 1. Другим вариантом является введение предиката IS [NOT] EMPTY, применяемого к выражению *exp* типа мультимножества (которое, возможно, включает подзапросы) и выдающего *true (false)* в том и только в том случае, когда результат вычисления *exp* пуст (непуст). Конечно, такую операцию можно было бы использовать при построении выражений с переключателем, а также можно было бы ввести сокращенную форму, аналогичную *COALESCE* в SQL:

COALESCE_ON_EMPTY (exp, v)

≡ *CASE WHEN exp IS NOT EMPTY THEN ELEMENT(exp) ELSE v END.*

- 7) Что касается данных, отсутствующих по причине их неприменимости, мы предлагаем использовать для работы с ними упрощенную схему декомпозиции отношений Дарвена. Упрощения происходят из-за того, что мы имеем дело только с одной причиной отсутствия данных, и трудно представить, что в одной таблице во многих столбцах могли бы отсутствовать данные по причине их неприменимости в некоторых строках (скорее всего, это свидетельствовало бы о наличии очень плохой схемы базы данных²¹).

В этом случае мы прибегаем к схеме декомпозиции, а не вводим еще одну разновидность специальных значений по нескольким причинам. Во-первых, схема декомпозиции

²¹ Для большей аккуратности стоит уточнить, что нам неизвестны ситуации, в которых нельзя было бы избежать на стадии проектирования базы данных появления таблиц с большим числом столбцов с отсутствующими данными по причине их неприменимости.

позволяет не сохранять явно информацию об отсутствии значения. Во-вторых, семантика вычислительных операций и в особенности операций сравнения по меньшей мере неочевидна. Наконец, у четвертого логического значения, которое неизбежно появилось бы при введении второй разновидности специальных значений, отсутствует интуитивно понятный смысл, не говоря уже о наличии отрицательного опыта Кодда с использованием четырехзначной логики (см. п. 3.1.1).

Предлагаемая упрощенная схема декомпозиции выглядит следующим образом. Пусть имеется исходная таблица $r\{key, a1, \dots, an, b1, \dots, bk\}$, где key – первичный или возможный ключ, в столбцах $a1, \dots, an$ не могут отсутствовать данные по причине их неприменимости, в столбцах $b1, \dots, bk$ данные могут отсутствовать. Тогда, если $k = 1$, то r декомпозируется на две таблицы $rab\{key, a1, \dots, an, b1\}$ и $ra\{key, a1, \dots, an\}$, в которых rab содержит все допустимые строки с присутствующим значением столбца $b1$, а ra – те строки, в которых значение столбца $b1$ отсутствует.

Если $k = 2$, то r декомпозируется на четыре таблицы $rab1b2\{key, a1, \dots, an, b1, b2\}$, $rab1\{key, a1, \dots, an, b1\}$, $rab2\{key, a1, \dots, an, b2\}$ и $ra\{key, a1, \dots, an\}$, где $rab1b2$ содержит все допустимые строки с присутствующими значениями столбцов $b1$ и $b2$, $rab1$ – все строки, в которых значения $b1$ присутствуют, а значения $b2$ отсутствуют, $rab2$ – все строки, в которых значения $b2$ присутствуют, а значения $b1$ отсутствуют и ra – все строки, в которых отсутствует значения и столбца $b1$, и столбца $b2$. Для произвольного k появятся $C_k^0 + C_k^1 + \dots + C_k^{k-1} + C_k^k = 2^k$ таблиц, что, конечно, недопустимо для больших k .

Если пользователям нужно формулировать запросы к таблице, которая включает все значения ключа исходной таблицы r , можно определить представление (*VIEW*), в котором (для $k = 2$, например) объединяются (*UNION*) таблицы $rab1b2, rab1$ с добавленным столбцом $b2$, заполненным во всех строках каким-либо реальным значением, которое уместно в данном приложении, $rab2$ с добавлением и заполнением столбца $b1$ и ra с добавленными столбцами $b1$ и $b2$ заполненными во всех строках теми же значениями, что в $rab2$ и $rab1$ соответственно.

4. Заключение

Современное состояние средств поддержки отсутствующей информации в реляционных базах данных оставляет желать лучшего. Следуя предписаниям основателя реляционной модели данных Эдгара Кодда, разработчики стандарта языка SQL затвердили в нем механизм универсального null-значения, приводящий к многочисленным противоречащим интуиции ситуациям при формулировке и выполнении запросов, и вынуждающий использовать странный трехзначный булевский тип данных, в котором нетипизированное null-значение отождествляется с третьим значением булевского типа.

В течение многих лет альтернативные подходы к управлению отсутствующей информацией разрабатывались и предлагались Кристофером Дейтом и Хью Дарвенем. В этих подходах для представления отсутствующих данных использовались сначала выделенные пользователями значения типов данных, а затем расширяющие типы данных специальные значения на основе традиционной двухзначной логики. Фактически, при следовании этим подходам вся нагрузка по управлению отсутствующими данными перекладывается на пользователей.

Тщательно проанализировав всю историю формирования и развития этих подходов, мы пришли к выводу, что ситуация, сложившаяся в стандарте SQL, является прямым следствием универсальности null-значения. Ограниченность же подходов Дейта и Дарвена неизбежна при использовании двухзначной логики. При этом и подход, идущий от Кодда, и схема специальных значений Дейта и Дарвена обладают и очевидными достоинствами: первый подход облегчает формулировку запросов, второй позволяет избавиться от универсального null-значения.

В предложенном нами подходе механизм специальных значений используется в комбинации с трехзначной логикой. На основе специальных значений мы обеспечиваем управление отсутствующими данными только одной категории – данные неизвестны. За счет введения специального значения в булевский тип, мы получаем полноценную трехзначную логику, в которой специальное значение играет роль третьего логического значения. Типизация специальных значений (отказ от универсального null-значения) позволяет учитывать при вычислении выражений особенности семантики типов данных и избегать ошибок, которые могут возникать в SQL-запросах. Для поддержки отсутствующих данных категории «данные неприменимы» мы предлагаем использовать упрощенный вариант схемы Дарвена на основе декомпозиции таблиц.

По нашему мнению, применение типизированных специальных значений вместо универсального null-значения позволило бы улучшить стандарт SQL. Однако мы отдаем себе отчет в том, что это очень большая работа, требующая пересмотра многих спецификаций. Имеется и техническая проблема, связанная с тем, что большинство типов данных в SQL являются безымянными, так что не очень понятно, как обозначать литералы специальных значений. Возможно, эту проблему можно было бы решить путем переосмысления понятия домена SQL таким образом, чтобы с помощью этого механизма можно было определять именованные типы данных со специальным значением.

Список литературы / References

- [1] Codd E.F. Derivability, Redundancy and Consistency of Relations Stored in Large Data Banks. IBM Research Report RJ599 (# 12343), 1969. Reprinted at ACM SIGMOD Record, 2009, Vol. 38, No. 1, 2009, pp. 17-36. Имеется перевод на русский язык: Э.Ф. Кодд. Выводимость, избыточность и согласованность отношений, хранимых в крупных банках данных. URL: http://citforum.ru/database/classics/first_rel_paper/.
- [2] Codd E.F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Volume 13, Number 6, 1970, pp. 377-387. Имеется перевод на русский язык: Е.Ф. Кодд. Реляционная модель данных для больших совместно используемых банков данных. URL: <http://citforum.ru/database/classics/codd/>.
- [3] Codd E.F. Understanding Relations (Installment #7). Bulletin of ACM-SIGMOD: The Special Interest Group on Management of Data, vol. 7, no. 3-4, 1975, pp. 23-28.
- [4] Codd E.F. Implementation of Relational Data Base Management Systems (NCC 1975 Panel). Bulletin of ACM-SIGMOD: The Special Interest Group on Management of Data, vol. 7, no. 3-4, 1975, pp. 3-22.
- [5] Chamberlin D.D., Astrahan M.M. et al. SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control. IBM Journal of Research and Development, V. 20, No. 6, 1976, pp. 560-575. Имеется перевод на русский язык: Д.Д. Чамберлин, М.М. Астрахан, К.П. Эсваран, П.П. Гриффитс, Р.А. Лори, Д.В. Мел, П. Райшер, Б.В. Вейд. SEQUEL 2: унифицированный подход к определению, манипулированию и контролю данных. URL: http://citforum.ru/database/classics/sequel_2/.
- [6] Codd E.F. Extending the Database Relational Model to Capture More Meaning. ACM Transactions on Database Systems, vol. 4, issue 4, 1979, pp. 397-434. Имеется перевод на русский язык: Э.Ф. Кодд. Расширение реляционной модели для лучшего отражения семантики. URL: http://citforum.ru/database/classics/codd_2/.
- [7] ISO/IEC 9075-2:1999. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation).
- [8] Codd E.F. The Relational Model for Database Management: Version 2. Addison-Wesley, 1990, 538 p.
- [9] Date C.J. The Default Values Approach to Missing Information. In C.J. Date (with Hugh Darven). Relational Database: Selected Writings 1989-1991. Addison-Wesley, 1992, pp. 343-354.
- [10] Date C.J. Faults and Defaults (in five parts). In C. J. Date (with Hugh Darven and David McGoveran). Relational Database: Selected Writings 1994-1997. Addison-Wesley, 1998, 608 p.
- [11] Date C.J. Hugh Darven. Databases, Types and the Relational Model: The Third Manifesto. 3rd Edition. Addison-Wesley, 2006, 556 p.

- [12] Codd E.F. Missing information (applicable and inapplicable) in relational databases. ACM SIGMOD Record, vol. 15, issue 4, 1986, pp. 53-78.
- [13] Codd E.F. More commentary on missing information in relational databases (applicable and inapplicable information). ACM SIGMOD Record, vol. 16, issue 1, 1987, pp. 42-50.
- [14] ANSI X3.135-1986. Information Technology – Database Languages – SQL.
- [15] ISO 9075:1987. Information processing systems — Database language — SQL.
- [16] ISO/IEC 9075:1989. Information processing systems — Database Language SQL with integrity enhancement.
- [17] ISO/IEC 9075:1992. Information technology — Database languages — SQL.
- [18] ISO/IEC 9075-1:1999. Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework).
- [19] ISO/IEC 9075-2:1999. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation).
- [20] ISO/IEC 9075-2:2003. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation).
- [21] ISO/IEC 9075-2:2008. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)
- [22] ISO/IEC 9075-2:2011. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)
- [23] ISO/IEC 9075-2:2016. Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)
- [24] Date C.J. A critique of the SQL database language. ACM SIGMOD Record, vol. 14, issue 30, 1984, pp 8-54.
- [25] Date C.J. Introduction to Database Systems, volume 2. Addison-Wesley, 1983, 383 p.
- [26] Date C.J. Null Values in Database Management. In C.J. Date. Relational Database: Selected Writings. Addison-Wesley, 1986, pp. 313-334.
- [27] Date C.J. NOT is Not “Not”! (Notes on Three-Valued Logic and Related Matters). In C.J. Date with a Special Contribution by Andrew Warden. Relational Database Writings 1985-1989. Addison-Wesley, 1990, pp. 217-248.
- [28] Date C.J., Darwen H. Foundation for Future Database Systems: The Third Manifesto. 2nd Edition. Addison-Wesley, 2000, 608 p.
- [29] Date C.J., Darwen H. Database Explorations: Essays on The Third Manifesto and Related Topics. Trafford Publishing, 2010, 548 p.
- [30] Darwen H. How to Handle Missing Information without Using NULL. Presentation Slides. Available at: <https://www.dcs.warwick.ac.uk/~hugh/TTM/Missing-info-without-nulls.pdf>, accessed 11.05.2023.
- [31] Darwen H., Smout E. How to Handle Missing Information Using S-by-C. Available at: <https://www.dcs.warwick.ac.uk/~hugh/TTM/HTHMIUS-by-C-review-draft.pdf>, accessed 11.05.2023.
- [32] Gessert G.H. Four Valued Logic for Relational Database Systems. ACM SIGMOD Record, vol. 19, issue 10, 1990, pp 29-35
- [33] Date C.J. Why Three- and Four-Valued Logic Don't Work. In C.J. Date. Date on Database. Writings 2000–2006, Apress, 2012, pp. 329-342.

Информация об авторе / Information about the author

Сергей Дмитриевич КУЗНЕЦОВ – доктор технических наук, профессор, главный научный сотрудник ИСП РАН, профессор кафедр системного программирования МГУ, МФТИ и ВШЭ. Научные интересы: управление данными, архитектуры систем управления данными, модели и языки данных, управление транзакциями, оптимизация запросов.

Sergey Dmitrievich KUZNETSOV – Doctor of Technical Sciences, Professor, Chief Researcher at ISP RAS, Professor at the Departments of System Programming of MSU, MIPT, and HSE. Research interests: data management, architectures of data management systems, data models and languages, transaction management, query optimization.

DOI: 10.15514/ISPRAS-2023-35(2)-7



Программный комплекс SIO для работы со структурированными данными

A.O. Игнатъев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

*Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13*

Аннотация. Решение задач численного моделирования различных физических процессов предполагает использование вычислительных ресурсов на разных стадиях подготовки, проведения расчетов и обработки их результатов. При этом существует проблема передачи данных между различными прикладными программными комплексами, используемыми в том числе на гетерогенных вычислительных ресурсах с разной архитектурой. В статье рассматриваются основные подходы по разработке и использованию программного обеспечения для работы со структурированными данными прикладных программных комплексов на примере HDF и SIO.

Ключевые слова: вычислительная система; структурированные данные; высокопроизводительные вычисления; математическое моделирование; форматы данных; HDF; SIO

Для цитирования: Игнатъев А.О., Мокшин С.Ю. Программный комплекс SIO для работы со структурированными данными. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 101-110. DOI: 10.15514/ISPRAS-2023-35(2)-7

SIO program suite for structured data processing

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

*E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia*

Abstract. Numerical modeling of various physical processes involves the use of computing resources at different stages of preparation, calculations and processing of their results. At the same time, there is a problem of data transfer between various application software used on heterogeneous computing resources with different architectures. The main approaches to the development and use of software for working with structured data of application software on the example of HDF and SIO discussed in this article.

Keywords: computing system; structured data; high-performance computing simulation; mathematical simulation subsystem; data format; HDF; SIO

For citation: Ignatyev A.O., Mokshin S.Yu. SIO program suite for structured data processing. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 101-110 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-7

1. Введение

Современные подходы по организации работы с генерируемыми на вычислительных системах и персональных компьютерах наборами структурированных данных предполагают использование некоторых, стандартизованных в мировом сообществе разработчиков

прикладного программного обеспечения, библиотек ввода-вывода. На данный момент основными требованиями к формату хранения данных можно назвать следующие:

- возможность хранения именованных данных разного типа и размера,
- возможность использования данных без дополнительного преобразования на вычислительных системах с гетерогенной аппаратной платформой и различными операционными системами.

Рассмотрим подходы к хранению данных с точки зрения удовлетворения данных требований.

2. Формат HDF

Безусловным лидером среди открытых проектов в области разработки стандартизованного формата хранения данных, удовлетворяющего вышеназванным требованиям, является проект HDF [1].

HDF – это многообъектный файловый формат с самоописанием, используемый для хранения и распространения научных данных. HDF был создан Национальным центром суперкомпьютерных приложений (NCSA) для удовлетворения потребностей различных групп ученых в различных инженерных проектах. Формат HDF обладает следующими свойствами:

- самоописание: для каждого объекта данных в файле HDF есть исчерпывающая информация (метаданные) о данных;
- универсальность: многие типы данных могут быть встроены в файл HDF;
- гибкость: HDF позволяет пользователям группировать связанные объекты данных вместе, помещать их в иерархическую структуру и добавлять описания и теги к объектам данных, позволяет пользователям помещать научные данные в несколько файлов;
- расширяемость: HDF может легко приспособиться к новым добавленным режимам данных в будущем и легко совместим с другими стандартными форматами;
- кроссплатформенность: HDF - это независимый от платформы формат файла (файлы HDF можно использовать на разных программно-аппаратных платформах без преобразования).

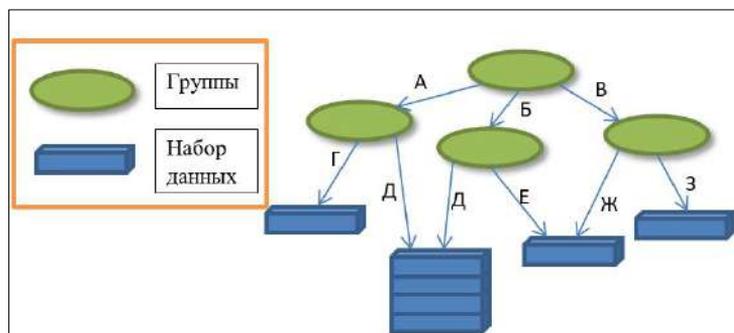


Рис. 1. Структура HDF-контейнера
Fig.1. HDF container structure

На данный момент разработчиками формата поддерживается версия HDF5. Для лучшего понимания логики использования HDF рассмотрим структуру файла HDF5. Формально файл HDF5 – это контейнер для хранения различных научных данных с помощью двух основных объектов данных – групп и наборов данных, как показано на рис. 1.

При этом набор данных HDF5 представляет собой многомерный массив элементов данных и некоторых вспомогательных метаданных, как показано на рисунке 2.

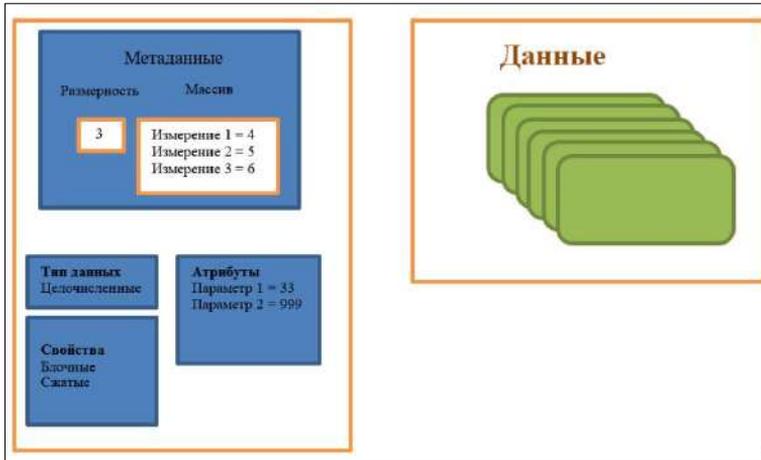


Рис. 2. Структура некоторого набора данных HDF
Fig.2. HDF dataset structure

В примере, приведенном на рис. 2, данные сохраняются, как трехмерный целочисленный массив размерностью 4x5x6. Он содержит некоторый набор атрибутов – параметров. Данные в массиве разделены на блоки и сжаты.

При всех преимуществах, такой подход обладает определенными недостатками:

- изначальная ориентация на максимальную универсализацию (хранение произвольных типов данных, допустимость сложной иерархической структуры данных) привела к излишнему усложнению как формата данных, так и интерфейса работы с данными,
- стремление к универсализации привело к многократному пересмотру спецификации HDF (за последние 20 лет были выпущены спецификации HDF3, HDF4, HDF5), приведшие к тому, что данные, записанные в старом формате, не читаются при использовании нового формата,
- поддержка новых форматов данных предъявляет специфические требования к системному окружению, в котором работает HDF, что, зачастую приводит к невозможности установки старых версий библиотек HDF на современные операционные системы.

Кроме того, многолетний опыт эксплуатации HDF в Российском Федеральном Ядерном Центре – Всероссийском научно-исследовательском институте технической физики имени академика Е.И. Забабахина (РФЯЦ-ВНИИТФ) показал постоянное наличие ошибок в данном программном обеспечении, в том числе влияющих как на информационную безопасность, так, иногда и на точность вычислений при важных научных расчетах. Эти ошибки, безусловно, оперативно выявляются и исправляются открытым сообществом разработчиков, но, в то же время, постоянно появляются новые.

3. Программный комплекс SIO

В то же время, в РФЯЦ-ВНИИТФ в 1998 году был разработан и внедрен в эксплуатацию программный комплекс SIO [2], полностью удовлетворяющий приведенным ранее требованиям к формату сохранения данных. Данный программный комплекс успешно применялся для хранения данных, совместно используемых на ЭВМ «Эльбрус-2» и другом вычислительном оборудовании вычислительного центра РФЯЦ-ВНИИТФ, а также фактически на всех ЭВМ с архитектурой x86-64. На данный момент программный комплекс SIO успешно развивается и интегрируется в прикладное программное обеспечение разработки РФЯЦ-ВНИИТФ.

SIO предназначен для хранения именованных объектов следующего вида:

- именованные записи (одномерные массивы) типа `integer*2`, `integer*4`, `integer*8`, `real*4`, `real*8` и `char` произвольной длины;
- многомерные массивы типа `integer*2`, `integer*4`, `integer*8`, `real*4`, `real*8`;
- номерованные структуры (имя структуры и номер экземпляра), где элементом структуры (полем) являются те же типы данных.

Программный комплекс SIO состоит из библиотеки `sio` и программы для просмотра файлов формата SIO – `sioview`. Библиотека `sio` содержит функции для работы с файлами и хранимыми объектами. Интерфейс `sio` состоит из следующих основных функций:

- работа с файлами (открытие и закрытие, вывод элементов каталога);
- работа с именованными записями (чтение, запись);
- работа со структурами (описание полей, чтение/запись поля экземпляра группы);
- работа с многомерными массивами (определение массива, чтение/запись массива).

Рассмотрим основные особенности работы с этими типами данных.

3.1 Хранение данных

Программный комплекс SIO предназначен для хранения большого объема структурированных данных и не имеет ограничений как по размеру хранимых данных, так и по их количеству. Хранение структурированных данных требует ведение каталога, в котором для каждого объекта присутствует имя объекта, его атрибуты (тип, формат и количество элементов) и адрес в файле. Каталог в SIO имеет сегментно-списочную структуру, т.е. состоит из сегментов фиксированной длины, объединенных списком. Каждый сегмент состоит из описания блоков данных и, при необходимости, включает в себя описание следующего сегмента каталога, как показано на рис. 3.



Рис. 3. Структура файла SIO

Fig. 3. SIO file structure

При создании файла в формате SIO создается первый сегмент каталога. При добавлении нового объекта в файл сначала проверяется, есть ли место в текущем сегменте. Если сегмент заполнен, то выделяется новый сегмент (ссылка на который включается в текущий сегмент)

и новый сегмент объявляется текущим. После этого новый объект включается в текущий сегмент каталога.

Набор функций для работы с каталогом включает в себя:

- получить количество объектов в каталоге файла,
- получить описание объекта из каталога по его порядковому номеру,
- получить описание объекта по имени.

Таким образом, SIO, как и HDF, является самоописываемым форматом хранения данных.

3.2 Именованные записи

Именованная запись представляет собой одномерный массив, имеющий имя, количество и формат элементов.

Набор функций для работы с именованными записями включает в себя:

- запись,
- чтение.

В параметрах этих функций указывается имя, формат элементов, количество элементов и, возможно, смещение относительно начала именованной записи в файле, что позволяет работать с именованной записью, как с подмассивом.

3.3 Структуры

Структуры предназначены для хранения множества экземпляров неоднородных элементов. Каждая структура имеет набор полей, идентичных по смыслу именованным записям.

Набор функций для работы с структурами включает в себя:

- определение структуры,
- описание полей структуры,
- запись экземпляра поля структуры,
- чтение экземпляра поля структуры.

При определении структуры указывается её имя и число полей в ней. Далее, для каждой структуры указывается имя поля, формат и количество элементов. Для доступа к объекту структуры в операциях чтения/записи указывается номер экземпляра структуры и имя поля.

3.4 Многомерные массивы

В последних версиях SIO появилась поддержка многомерных массивов. При этом размерность массива в SIO не ограничена ничем, кроме объема доступной памяти. В программном комплексе SIO реализованы функции, позволяющие:

- определить многомерный массив,
- получить информацию о размерности массива и длинах его по измерениям,
- записывать/читать массив как целиком, так и частями (подмассивами).

Необходимость работы с подмассивами связана с традиционной необходимостью обработки данных значительного объема на высокопроизводительных вычислительных системах, в этом случае часто возникает необходимость доступа только к части обрабатываемых данных. Это может быть связано как с недостаточностью ресурсов для обработки данных целиком, так и с распараллеливанием обработки данных для сокращения времени этой обработки.

Зачастую вместо понятия «подмассив» используется термин «вырезка» (slice). Под вырезкой из многомерного массива в общем случае понимается получение нового массива размерности той же или меньшей, элементы которого берутся из части исходного массива с отсеканием «ненужных» частей. Пример вырезки из двумерного массива показан на рис. 4.

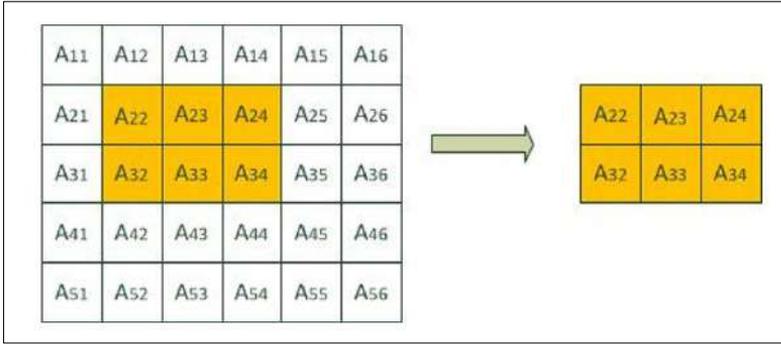


Рис. 4. Вырезка из двумерного массива
Fig.4. Slice from two-dimensional array

В SIO массивы в файле хранятся в виде одномерной последовательности элементов с построчной разверткой в стиле языка C (на рисунке 5 показан порядок хранения элементов для массива, приведенного на рис. 4).

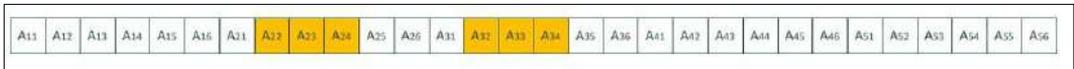


Рис. 5. Порядок хранения элементов двумерного массива
Fig.5. The order of storing elements of a two-dimensional array

Выбранная вырезка определяется тремя параметрами, первый из которых определяет начальный индекс вырезки в массиве, второй описывает размерность вырезки, а третий – количество элементов по каждому измерению массива, составляющих вырезку.

В общем случае, для выполнения вырезки из массива требуется либо прочитать весь массив в оперативную память с последующей пересылкой элемента из исходного массива в массив для вырезки в программе, либо несколько раз производить процедуру чтения из файла требуемых частей массива и, опять же, пересылать полученные элементы в результирующий массив, увеличивая накладные расходы на выполнение операции. Поэтому предпочтительно использовать в качестве вырезок построчные сечения.

Сечение отличается от вырезки тем, что начальные и конечные значения индексов в сечении совпадают с теми же значениям в исходном массиве. При этом размерность сечения становится меньше, чем у исходного массива. Построчное сечение определяется фиксированным первым индексом массива. Таким образом, информация состоит из одной строки или набора смежных по памяти строк массива.

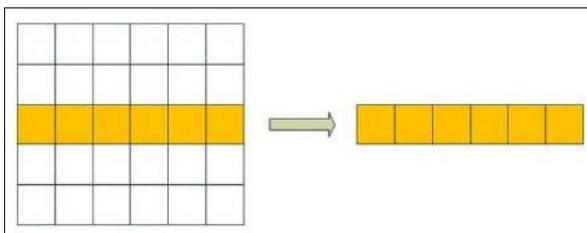


Рис. 6. Одномерное сечение двумерного массива
Fig.6. One-dimensional cross section of a two-dimensional array

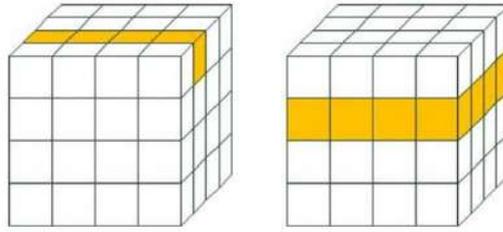


Рис. 7. Одно- и двумерное сечения трехмерного массива
Fig.7. One- and two-dimensional sections of a three-dimensional array

На рис. 6 показано одномерное сечение двумерного массива. На рис. 7 показаны одномерное и двумерное сечения трехмерных массивов. Выбором подходящего сечения можно добиться сокращения числа чтений из файла до одного и исключения необходимости перемещения информации по оперативной памяти.

Для двумерных массивов A_{ij} таким сечением является строка при фиксированном значении i . Для трехмерных массивов A_{ijk} таких сечений два типа:

- строка при фиксированных значениях i и j (одномерное сечение);
- плоскость при фиксированном значении i (двумерное сечение).

На рис. 8 приведены примеры хранения трехмерного массива с длинами по измерениям, равными двум, и выбираемые для двух сечений элементы массива.



Рис. 8. Порядок хранения элементов трехмерного массива и выбираемого сечения
Fig.8. The elements order of a three-dimensional array and its selected slice

3.5 Пример работы с библиотекой sio

Рассмотрим приведенный на листинге 1 пример исходных текстов программы, которая сначала заполняет массив по колонкам, а потом читает по строкам.

```

1 #define SIO_REAL16 18 // формат элементов
2 const int lcat = 8;
3 int ko, i, j;
4 char* aname="d2_2"; // имя массива
5 int idx2[2]; // массив для индексов вырезки
6 int lens2[2]; // массив для длин по измерениям вырезки
7 double column[L1]; // массив для колонки
8 double line[L2]; // массив для строки
9
10 ko = siofop(fname,0,STNEW,lcat,0,&hf); // создание нового файла
11 ko = sioadef(hf, aname, 2, dims2, SIO_REAL16); // создание 2D массива
12
13 for (j = 0; j < L2; j++) { // цикл по колонкам
14     for (i = 0; i < L1; i++) { // формирование элементов колонки
15         column[i] = - (double)(i * LM + j);
16     }
17     idx2[0] = 0; idx2[1] = j; // начальный индекс для колонки
18     lens2[0] = L1; lens2[1] = 1; // длины по измерениям колонки
19     ko = sioawrs2(hf, aname, 1, idx2, lens2, column); // запись колонки
20 }
    
```

```
21
22 ko = siofcl(hf, 0);
23
24 ko = siofop(fname, 0, STOLD, lcat, 0, &hf); // открытие файла
25
26 for (i = 0; i < L1; i++) {
27     idx2[0] = i; idx2[1] = 0; // индекс строки
28     lens2[0] = 1; lens2[1] = L2; // длины по измерениям строки
29     ko = sioards2(hf, aname, 1, idx2, lens2, line); // чтение строки
30     printf("I=%2d: %7.2f %7.2f .. %7.2f %7.2f\n", i, line[0],
31         line[1], line[L2-2], line[L2-1]); // проверка содержимого строки
32 }
33 ko = siofcl(hf, 0);
```

Листинг 1. Пример исходного текста программы
Listing 1. Program source code example

В строках 1-8 приводится описание требуемых для работы констант и переменных. Строка 10 содержит обращение к функции создания нового файла. Строка 11 содержит обращение к функции описания двумерного массива. В строках 13-20 производится построчная запись в массив сформированной информации. Строка 22 содержит обращение к функции закрытия файла. Строка 24 содержит обращение к функции открытия этого же файла на чтение. В строках 26-31 производится построчная выборка из массива и печать выбранных значений. Строка 33 содержит обращение к функции закрытия файла.

3.6 Дополнительные инструменты SIO

Для упрощения процедуры перехода с формата HDF на формат SIO специалистами РФЯЦ-ВНИИТФ был разработан преобразователь данных hdf2sio, который из указанного HDF файла выбирает необходимый набор данных и дописывает его в создаваемый файл SIO.

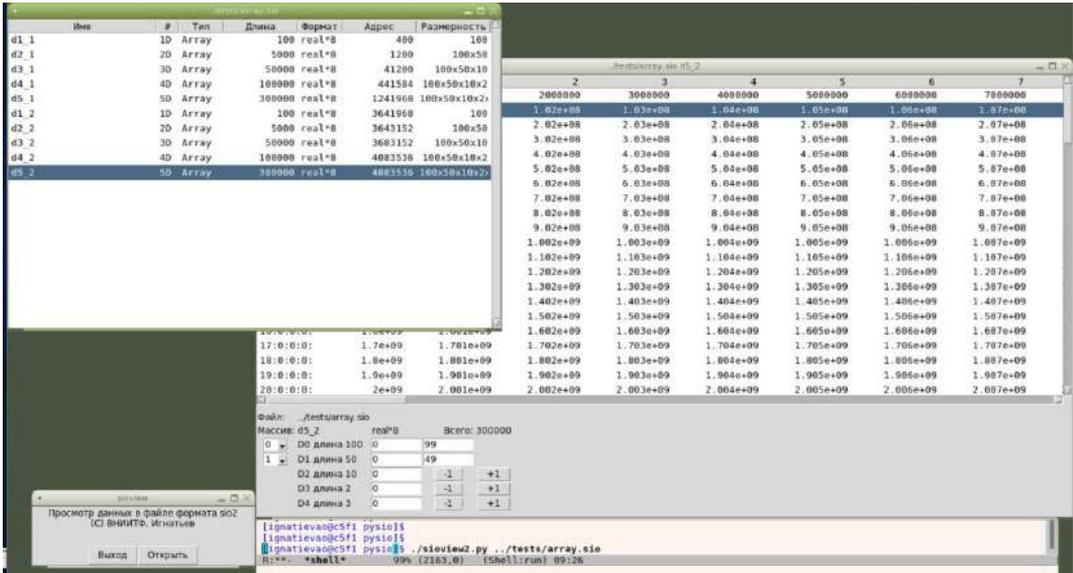


Рис. 9. Просмотр каталога файла и выбранных массивов
Fig.9. File directory and selected arrays viewing

Кроме того, для интерактивного просмотра содержимого файла в формате SIO реализована графическая утилита sioview, входящая в состав программного комплекса SIO и написанная на языке программирования Python [4] с использованием графической библиотеки Tk [5]. Эта

утилита открывает указанный в параметре или выбранный в диалоге файл, выдает его каталог и, после выбора именованной записи, в отдельном окне выдается содержимое записи в выбранном диапазоне.

На рис. 9 показан вывод каталога выбранного файла и выбранного массива. Поля ввода «от» и «длиной» определяют область выводимой для записи информации. Одномерный массив $M(I)$ выдается в режиме построчной развертки. Двумерный массив $M(I,J)$ выдается в виде таблицы, где строки соответствуют значению индекса I , а по столбцам выдается значение J . Трехмерный массив $M(I,J,K)$ выдается в виде таблицы для выбранного значения индекса I , строки соответствуют значению J , а по столбцам выдается значение индекса K . Кнопки навигации позволяют изменять начальные индексы по измерениям.

4. Заключение

Рассмотренный в данной статье программный комплекс SIO обеспечивает выполнение сформулированных ранее требований к формату хранения данных. Так же, как и HDF, SIO является самоописываемым форматом хранения. Реализация SIO обеспечивает возможность работы с данными, полученными на разных вычислительных платформах и под различными операционными системами. SIO не обладает рядом возможностей HDF (иерархия и многообразие хранимых объектов), поэтому программный код SIO существенно проще и не требует при своей установке и использовании дополнительных программных продуктов. Это существенно облегчает его использование на различных вычислительных платформах и операционных системах.

Программный комплекс SIO успешно используется в РФЯЦ-ВНИИТФ уже более 24 лет, последняя версия SIO включена в реестр отечественного программного обеспечения РФ. Авторы выражают надежду, что изложенная в статье информация о структуре и функциональных возможностях SIO окажется полезной другим специалистам, занимающимся разработкой прикладного программного обеспечения в РФ.

Список литературы / References

- [1] HDF Group. Available at: <https://www.hdfgroup.org/solutions/hdf5/>, accessed 10.04.2023.
- [2] Программный комплекс SIO. / SIO program suite. Available at: <http://vniitf.ru/article/programmniy-kompleks-sio>, accessed 10.04.2023.
- [3] GNU Bash. Available at: <https://www.gnu.org/software/bash/>, accessed 10.04.2023.
- [4] Python. Available at: <https://www.python.org/>, accessed 10.04.2023.
- [5] Tk Library. Available at: <https://docs.python.org/3/library/tk.html>, accessed 10.04.2023.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of Laboratory. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Сергей Юрьевич МОКШИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.



Пути организации параллельного доступа к структурированным данным

A.O. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

Д.В. Иванков, ORCID: 0000-0003-4254-0104 <d.v.ivankov@vniitf.ru>

Е.А. Бекетов, ORCID: 0009-0008-0766-6258 <e.a.beketov@vniitf.ru>

*Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13*

Аннотация. В данной работе исследуются пути достижения максимально возможной производительности обменов с файлами, содержащими структурированные данные. Исследования проводились на файловых системах с параллельным доступом вычислительных систем, предназначенных для решения задач физико-математического моделирования различных процессов и объектов. На примере параллельной файловой системы Lustre рассматривается параллельный доступ к «сырым» данным. Предлагается способ организации параллельного доступа к структурированным данным на основе специального разработанного формата хранения PSIO и библиотеки доступа psio. Выполняется сравнительный анализ производительности ввода-вывода разработанного формата хранения данных и формата параллельной версии HDF5.

Ключевые слова: вычислительная система; структурированные данные; высокопроизводительные вычисления; математическое моделирование; форматы данных; параллельный ввод/вывод.

Для цитирования: Игнатьев А.О., Мокшин С.Ю., Иванков Д.В., Бекетов Е.А. Пути организации параллельного доступа к структурированным данным. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 111-126. DOI: 10.15514/ISPRAS-2023-35(2)-8

Благодарности: Авторы публикации выражают особую благодарность Р.М. Шагалиеву и В.А. Свиридову за предоставленные для проведения тестов вычислительные ресурсы Полигона НЦФМ (г. Саров).

Ways to organize parallel access to structured data

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

D.V. Ivankov, ORCID: 0000-0003-4254-0104 <d.v.ivankov@vniitf.ru>

E.A. Beketov, ORCID: 0009-0008-0766-6258 <e.a.beketov@vniitf.ru>

*E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia*

Abstract. This paper explores ways to achieve the highest possible exchange performance with files containing structured data. The research was carried out on file systems with supercomputer systems parallel access designed to solve problems of physical and mathematical modeling of various processes and objects. For example, parallel access to raw data is considered using the Lustre file system. The article suggests a way to organize parallel access to structured data based on a specially developed PSIO storage format and the psio access library. A comparative analysis of the I/O performance of the developed data storage format and the HDF5 parallel version format is performed.

Keywords: computing system; structured data; high-performance computing simulation; mathematical simulation subsystem; data format; HDF; HPC; SIO.

For citation: Ignatyev A.O., Mokshin S.Yu., Ivankov D.V., Beketov E.A. Ways to organize parallel access to structured data. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 111-126 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-8

Acknowledgements. The authors express special gratitude to R.M. Shagaliev and V.A. Sviridov for the NCFM Polygon computing resources provided for the tests (Sarov city).

1. Введение

Развитие и усложнение цифровых моделей различных физических объектов неизбежно приводит к увеличению объемов обрабатываемой на вычислительных системах (ВС) информации, а также широкому использованию параллельных файловых систем (ПФС) и механизмов массового распараллеливания прикладных программ, использующих структурированные данные, что, в свою очередь, неизбежно ставит вопрос о повышении скорости доступа к этим данным.

Простой путь увеличения производительности доступа к данным заключается в том, что данные каждого процесса расчетной задачи размещаются в отдельном структурированном файле, таким образом, доступ к ним осуществляется в два этапа:

- сначала определяется имя файла исходя из номера процесса;
- потом, по соответствующему запросу, выбирается блок данных из этого файла (например, с использованием функционала HDF5 [1]).

Для хранения объектов также можно использовать отдельные файлы, в имя которых входит номер процесса и имя объекта.

Таким образом, при таком подходе, структурированный набор данных представляется множеством файлов в общем каталоге. Благодаря встроенному в ПФС механизму размещения данных среди группы устройств хранения ПФС, все входящие в этот структурированный набор файлы более-менее равномерно распределяются по всем Object Store Target (OST) ПФС Lustre [2]. Такая технология называется «файл на процесс» и получила распространение во всем мире, а также широко используется в РФЯЦ-ВНИИТФ. Эта же технология используется, как одна из возможных, при определении производительности ПФС на тестах IO-500 [3].

Производительность при этом подходе обеспечивается базовыми возможностями параллельной файловой системы (ПФС), благодаря автоматическому распределению потоков ввода/вывода между всеми доступными устройствами хранения ПФС.

При очевидной простоте этого подхода он имеет несколько недостатков:

- двухэтапная схема адресации объектов: сначала имя файла, потом объект в файле;
- существенное увеличение количества файлов в ПФС, прямо пропорциональное количеству процессов в задаче, создающее повышенную нагрузку на метасервис ПФС, что отрицательно сказывается на качестве функционирования файловой системы.

Другой путь реализации заключается в заложенных в ПФС возможностях по параллельному доступу процессов задачи к одному файлу. Так устроен механизм MPI-IO [4], используемый в основе современных библиотек структурированного ввода/вывода типа HDF5.

Обширность темы эффективного параллельного ввода-вывода не позволяет рамками отдельной статьи охватить всё многообразие используемых методов и инструментов. Поэтому остановимся на рассмотрении основных принципов и выявлении характерных проблем параллельного ввода-вывода, с которыми могут сталкиваться начинающие разработчики High Performance Computing (HPC) приложений, а также приведём примеры программ, реализующих различные методы параллельного доступа к данным в файловой системе Lustre.

Для этого вначале рассматриваются общие принципы и возможности параллельного доступа к данным на ПФС на примере Lustre, делаются предположения об организации параллельного доступа к структурированным файлам, потом производится проверка на модельной версии библиотеки psio и HDF5.

2. Проверка базовых возможностей Lustre

Параллельная файловая система Lustre предполагает для организации параллельного доступа к наборам данных использовать стрипование (разделение) файлов, при котором файл разбивается на сегменты одинакового размера, определяемые атрибутом `stripe_size` (по умолчанию 1048576 байт или 1 МБ). Сегменты файла распределяются между OST (Objects Store Target – устройство хранения) ПФС Lustre. Число задействованных OST определяется атрибутом `stripe_count`, задается при создании файла и не должно превышать число используемых в ПФС OST. Тем самым, файл с `stripe_count = 1` является обычным файлом с последовательным доступом. Файл же с `stripe_count` больше 1 допускает параллельный доступ к сегментам, расположенным на разных OST. При задании `stripe_count`, равным 0, файл распределяется на все доступные OST.

На рис. 1 показана общая схема стрипования файла для ПФС, содержащей n OST. Файл содержит m групп сегментов по n сегментов в группе. Цвет сегмента определяет размещение его на соответствующем OST.

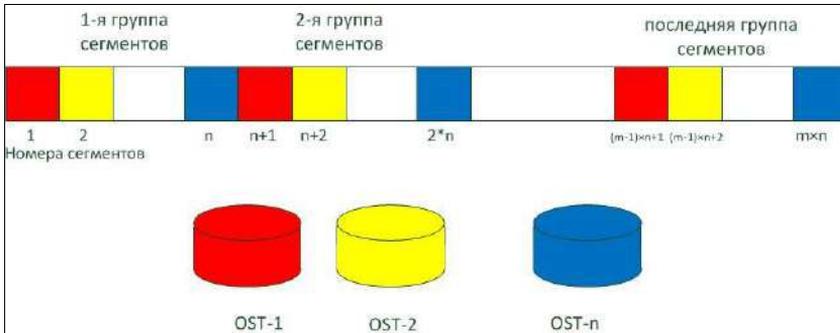


Рис. 1. Общая схема стрипования файлов
Fig.1. The general scheme of file striping

2.1 Исходные данные

Для проверки возможностей параллельного доступа к стрипованному файлу на вычислительном комплексе (ВК) «Полигон НЦФМ» (г. Саров) [5] создано тестовое пространство на ПФС Lustre, производительность которой на тесте IOR из состава набора тестов IO-500 [3] на операции записи составила 9.447 ГБ/с при общем объеме данных 2 ТБ и отсутствии посторонней нагрузки. Этот показатель будем считать максимальной производительностью ПФС и использовать в дальнейшем для сравнения.

2.2 Методика оценки производительности параллельного ввода-вывода

2.2.1 Оценка пропускной способности одного канала доступа к ПФС

Важным показателем при исследовании параллельности обменов с ПФС является ускорение, полученное при использовании множества потоков по отношению к одному, поэтому для нас

интересна как пропускная способность одного канала (используется один поток данных), так и пропускная способность ПФС при использовании нескольких потоков.

Для оценки пропускной способности одного канала осуществляется последовательная запись в файл с атрибутом `stripe_count=1` с одного процесса задачи блоками большой длины, с вытаскиванием файловых кэшей на стороне клиента после записи каждого блока, как показано на рис. 2. Скорость записи определяется как отношение суммарного объема записываемых блоков к времени записи. Второй процесс задачи на другом вычислительном узле (чтобы исключить влияние файловых кэшей на стороне клиента) последовательно читает эти блоки из того же файла. Скорость чтения определяется как отношение суммарного объема прочитанных блоков к времени чтения.

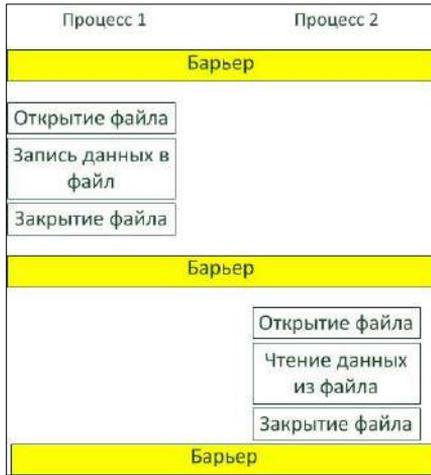


Рис. 2. Схема определения пропускной способности канала ПФС
Fig.2. The scheme of determining Lustre channel bandwidth

2.2.2 Оценка пропускной способности ПФС Lustre на параллельных обменах

Для оценки пропускной способности ПФС при использовании нескольких потоков на первом шаге со стороны всех процессов параллельной задачи производится запись данных в файл с `stripe_count`, отличным от 1, как показано на рис. 3. Количество используемых потоков (процессов) выбирается равным или кратным атрибуту `stripe_count`. Запись организуется в цикле, до достижения требуемого объема передаваемого с процесса информации. При этом каждому процессу назначаются соответствующие области в файле. На первой итерации этого цикла каждый процесс пишет в файл блок размера `stripe_size` со смещением, равным порядковому номеру процесса (его ранку в параллельной задаче), умноженному на `stripe_size`. На следующих итерациях смещение увеличивается на объем записанной информации (`stripe_size × прос`, где `прос` – число процессов в параллельной задаче). На втором шаге производится аналогичным способом чтение блоков данных из этого файла, при этом, для исключения влияния файловых кэшей на стороне клиента, отображение процессов задачи на блоки данных в файле производится со смещением:

- процесс 0 читает блок 1, записанный на процессе с ранком 1,
- процесс 1 читает блок 2, записанный на процессе с ранком 2,
- последний процесс читает блок 0, записанный на процессе с ранком 0.



Рис. 3. Схема определения пропускной способности ПФС
 Fig.3. The scheme of determining Lustre multi-channel bandwidth

2.2.3 Результаты замеров на неструктурированных данных

Для тестирования производительности ввода-вывода использовалась тестовая программа, позволяющая создавать файл при запуске программы с разными атрибутами. При тестировании выполнялось следующее:

- при каждом запуске файл создавался заново с атрибутом `stripe_count`, равным числу потоков данных (пока это число меньше количества OST), или равным количеству OST (в случае если это число больше количества OST);
- для каждого числа потоков на обмен подавались блоки размером 1, 2, 4, 8, 16 и 32 МБ (в таблице колонка BS) с сохранением суммарного объема информации в потоке равным 96 МБ, при этом использовалось 96, 48, 24, 12, 6 и 3 итераций (в таблице колонка Nit);
- параметр `stripe_size` (в таблице колонка SS) при создании файла принимался равным BS;
- каждый тест запускался не менее трех раз, при этом в таблицы приведены лучшие показатели: V_w – производительности на записи, K_w – ускорение записи по отношению с усредненной производительности на одном потоке, V_r – производительности на чтении, K_r – ускорение чтения по отношению с усредненной производительности на одном потоке.

В табл. 1 приведены результаты, полученные по этой методике на ВК. Выделены лучшие результаты для каждой конфигурации прогона.

Пропускная способность канала на ВК составила 655.6 МБ/с и достигается на блоках в 32 МБ с соответствующим размером сегмента. Увеличение размера блока положительно влияет на производительность, поэтому для числа потоков больше четырех использовались только блоки размером 8, 16 и 32 МБ.

Табл. 1. Пропускная способность ПФС на ВК
 Table 1. Cluster Lustre bandwidth

BS	Nit	SS	V_w	V_r	K_w	K_r
1 поток, размер 96 МБ						
1	96	1	293.8	865.2		
2	48	2	342.8	772.8		
4	24	4	395.9	740.9		

8	12	8	504.9	805.0		
16	6	16	596.8	752.9		
32	3	32	655.6	786.3		
2 потока, размер 192 МБ						
1	96	1	575.1	1387.1	1.3	1.7
2	48	2	678.1	1301.3	1.5	1.6
4	24	4	804.3	1428.1	1.8	1.8
8	12	8	974.5	1407.0	2.1	1.8
16	6	16	1183.4	1412.1	2.6	1.8
32	3	32	1300.2	1612.8	2.8	2.0
4 потока, размер 384 МБ						
8	12	8	2146.4	2839.4	4.7	3.6
16	6	16	2364.2	2857.7	5.2	3.6
32	3	32	2580.4	3005.0	5.6	3.8
8 потоков, размер 768 МБ						
8	12	8	4446.9	6483.6	9.7	8.1
16	6	16	5077.6	6300.7	11.1	7.9
32	3	32	5731.9	6615.5	12.5	8.3
16 потоков, размер 1536 МБ						
8	12	8	8063.7	11635.0	17.6	14.6
16	6	16	9597.2	11487.8	20.9	14.4
32	3	32	8991.5	11864.4	19.6	14.9
32 потока размер 3072 МБ						
8	12	8	9001.8	20444.3	19.6	25.7
16	6	16	9495.7	21090.4	20.7	26.5
32	3	32	8159.2	21946.6	17.8	27.5

Максимальная скорость записи на ВК составила 9.49 ГБ/с на 32 потоках с блоком 32 МБ, что соответствует производительности на тесте IOR (9.45 ГБ/с). Ускорение по отношению к усредненному однопоточному результату составило 20.7 раза.

3. Моделирование параллельного ввода-вывода структурированных данных

3.1 Особенности работы со структурированными данными

Структурированные файлы, в общем случае, содержат каталог и блоки данных. Каталог содержит описание блоков данных (идентификатор блока данных, тип данных в блоке, размер блока и смещение по файлу для размещения блока).

Поскольку максимальная производительность обмена процесса с ПФС достигается в том случае, когда потоки данных с процессов на OST не пересекаются по OST, необходимо обеспечить выравнивание блоков данных по границам сегментов файла, т.е. каталогу отвести один или более целых сегментов и каждый блок данных размещать с границы очередного сегмента. Кажущееся увеличение размера файла, за счет пустого пространства в конце каждого сегмента по сравнению с плотной упаковкой данных в файле или представлением набора данных в виде множества файлов, компенсируется тем, что ПФС выделяет дисковое пространство под файлы с точностью до размера сегмента.

Рассмотрим простейшую схему организации структурированного набора данных и работы с ним:

- набор данных содержит в себе именованные блоки данных;
- каждый блок данных представляет собой одномерный массив чисел;
- все блоки данных описываются каталогом, являющимся неотъемлемой частью набора данных;

- для каждого блока данных в каталоге присутствуют имя блока, число и формат элементов, смещение по файлу или индекс сегмента в файле (или множество индексов для блоков данных, размер которых превышает размер сегмента);
- выделяемые сегменты описываются в таблице (карте) сегментов, которая, наравне с каталогом, является метainформацией структурированного набора данных;
- порядок описания блоков данных в каталоге не существен;
- для операции чтения/записи блока данных требуется заранее открытый файл, имя блока данных и массив для обмена информацией в оперативной памяти процесса задачи;
- все операции с набором данных являются коллективными в смысле MPI, т.е. одновременно выполняются на всех процессах, входящих в коммуникационную группу;
- во избежание конфликтов по доступу к OST со стороны процессов задачи, все сегменты одного блока данных должны располагаться на одном OST;
- операции, модифицирующие каталог, выполняются на ранке 0 задачи по групповым запросам от остальных ранков, модифицированный каталог рассылается всем ранкам задачи.

Предлагаемая схема работы с данными набора со стороны задачи идентична схеме, описанной в п. 2.2.1 и 2.2.2 (рис. 2 и 3), за исключением того, что доступ к данным в файле производится не по индексу в файле, а по имени блока данных.

3.2 Реализация работы со структурированными файлами

Для проверки работы со структурированными файлами была написана библиотека psio, содержащая следующие функции:

- PSIO_Fop – открытие (создание) стрипованного файла на ФС Lustre;
- PSIO_Fcl – закрытие файла;
- PSIO_Wwr – запись именованного блока данных;
- PSIO_Brd – чтение именованного блока данных;
- PSIO_Fcat – вывод каталога файла (имен и характеристик блоков данных).

Файл формата PSIO имеет следующую структуру:

- вся информация в файле содержится в сегментах, размером stripe_size байт, и индексируется номером сегмента,
- каждый объект в файле представлен набором сегментов;
- файл содержит каталог и карту сегментов, содержащую пары: индекс объекта и номер сегмента объекта;
- каталогу присваивается номер объекта 0;
- карте сегментов присваивается номер объекта 1;
- блокам данных присваиваются следующие номера объектов;
- каждый блок данных в файле записывается сегментами и представлен записью в каталоге, в которой указывается имя блока, формат и количество элементов в нем, количество сегментов в файле;
- всем сегментам блока выделяется сегменты файла, расположенные на одном OST.

На листинге 1 приведен пример программы, записывающий с каждого ранка параллельной задачи по одному блоку данных двойной точности длиной 8 МБ в файл, стрипованный на все OST с размером сегмента 2 МБ.

```
1 #include <mpi.h>
2 #include <psio.h>
3 PSIO_HF *hf; /* дескриптор файла */
```

```
4 int stripe_count = 0;          /* на все OST */
5 int stripe_size = 2*1024*2014; /* сегмент 2 МБ */
6 int main(int argc, char** argv)
7 {
8     int iRank, iSize;
9     char bname[20];
10    int blen = 1024*2014;
11    double array;
12    /* Инициализация MPI */
13    ierr = MPI_Init(&argc, &argv);
14    MPI_Comm_rank( MPI_COMM_WORLD, &iRank);
15    MPI_Comm_size( MPI_COMM_WORLD, &iSize);
16    /* Формирование опций для создания файла */
17    sprintf(opts, "stripe_count=%d, stripe_size=%d stripe_offset=-1",
18            stripe_count, stripe_size);
19    /* Создание файла для параллельного доступа */
20    if (PSIO_Fop(fname, "new", &hf, opts, MPI_COMM_WORLD)) exit(-1);
21    /* Запись блоков данных */
22    sprintf(bname, "B%03d", iRank);
23    array = GetData(iRank, blen); /* Генерация 8 МБ данных */
24    if (PSIO_Bwr(hf, bname, "real*8", blen, array, MPI_COMM_WORLD))
25        exit(-1);
26    /* Вывод каталога файла */
27    PSIO_Fcat(hf);
28    /* Закрытие файла */
29    if (PSIO_Fcl(hf, MPI_COMM_WORLD)) exit(-1);
30    /* Завершение MPI */
31    MPI_Finalize();
32    return 0;
33 }
```

Листинг 1. Пример исходного текста программы, выполняющей запись данных в параллельном режиме

Listing 1. An example of the source code of a program that writes data in parallel mode

Эта тестовая программа выполняет обмены со структурированным файлом, при этом выполняются следующие действия:

- файл открывается на запись (строка 19),
- все процессы в цикле по указанному в параметрах числу итераций производят запись в файл блоков данных заданной длины. Имя блока строится на основании ранка процесса и номера итерации (строки 22-23);
- файл закрывается (строка 25).

Каждая функция библиотеки оформлена как коллективная операция в смысле MPI, при этом:

- чтение каталога файла и карта сегментов при его открытии выполняется на ранке 0 задачи, после этого каталог и карта сегментов рассылается остальным процессам;
- при записи нового блока с процессов задачи запрос на включение блока в каталог файла отсылается на ранк 0 задачи, где выполняется операция по модификации каталога и карты сегментов, после этого модифицированная метаинформация рассылается остальным процессам задачи;
- при закрытии модифицированного файла сохранение каталога и карты сегментов производится с ранка 0;
- все взаимодействие между процессами задачи для обеспечения одинаковости метаинформации производится средствами MPI.

3.3 Полученные результаты

Поскольку каждая операция с файлом, кроме обменов с ПФС, содержит обращения к функциям MPI, создающие дополнительные накладные расходы, при тестировании использовались блоки разной длины, размещаемые в нескольких сегментах файла. Скорость выполнения обмена определялась как отношение полного объема данных к времени выполнения цикла записи/чтения с учетом времени, необходимого на открытие и закрытие файла.

Поскольку в библиотеке psio блоки данных пишатся сегментами с размером, равным атрибуту stripe size, при тестировании варьировались как размер блока, так и размер сегмента, при этом размер передаваемой на потоке информации увеличен до 1 ГБ.

Проведена следующая серия замеров:

- сначала на одном и двух потоках проверялось влияние размера блока и размера сегмента при больших блоках на скорость записи, при этом обнаружено, что при размере блока в 512 МБ и размере сегмента 8 МБ наступает насыщение;
- затем, для блоков в 128, 256 и 512 МБ и сегменте 8 МБ проведены расчеты на 10, 48 и 96 потоках данных.

В табл. 2 приведены результаты, полученные на ВК.

Получена максимальная скорость записи 9.885 ГБ/с на 32 потоках данных блоками по 512 МБ и размере сегмента 8 МБ, что даже лучше, чем результаты, полученные на тесте IOR и неструктурированных данных. Увеличение производительности, видимо, связано с увеличением объема данных, передаваемых за одну операцию.

Табл. 2. Пропускная способность ПФС на структурированных данных на ВК
Table 2. Cluster Lustre bandwidth on a structured data

BS	Nit	SS	Vw	Vr	Kw	Kr
1 поток, размер 1024 МБ						
1	1024	1	319.3	767.7	0.8	0.9
2	512	1	325.9	726.0	0.8	0.9
4	256	1	314.3	735.7	0.8	0.9
8	128	1	320.7	742.7	0.8	0.9
16	64	1	327.3	721.6	0.8	0.9
32	32	1	319.6	717.2	0.8	0.9
64	16	1	324.2	777.1	0.8	1.0
128	8	1	324.7	727.1	0.8	0.9
128	8	2	395.9	776.0	1.0	1.0
128	8	4	457.9	828.6	1.2	1.0
128	8	8	602.9	1011.5	1.5	1.2
256	4	8	600.5	1008.3	1.5	1.2
512	2	8	596.4	978.4	1.5	1.2
2 потока, размер 2048 МБ						
1	1024	1	515.3	1712.9	1.3	2.1
2	512	1	527.3	1677.3	1.3	2.1
4	256	1	527.6	1747.2	1.3	2.1
8	128	1	526.9	1692.3	1.3	2.1
16	64	1	521.4	1757.2	1.3	2.2
32	32	1	537.5	1716.8	1.4	2.1
64	16	1	552.2	1764.4	1.4	2.2
128	8	1	549.7	1852.1	1.4	2.3
128	8	2	664.1	1905.6	1.7	2.3
128	8	4	801.1	1404.3	2.0	1.7

128	8	8	1074.1	2203.1	2.7	2.7
256	4	8	1088.1	2165.8	2.8	2.7
512	2	8	1102.4	2146.8	2.8	2.6
4 потока, размер 4096 МБ						
128	8	8	1376.5	5432.9	3.5	6.7
256	4	8	1395.3	5356.0	3.5	6.6
512	2	8	1467.7	5209.6	3.7	6.4
8 потоков, размер 8192 МБ						
128	8	8	3158.9	13242.5	8.0	16.3
256	4	8	2762.4	10055.4	7.0	12.4
512	2	8	2781.5	12116.0	7.0	14.9
16 потоков, размер 16384 МБ						
128	8	8	7216.8	13980.0	18.3	17.2
256	4	8	7084.7	13583.5	18.0	16.7
512	2	8	6956.6	13277.0	17.6	16.3
32 потока, размер 32768 МБ						
128	8	8	8516.0	25526.8	21.6	31.4
256	4	8	9160.2	25391.1	23.2	31.2
512	2	8	9885.7	24968.0	25.1	30.7

4. Использование параллельного ввода-вывода в HDF5

4.1 Работа с данными в HDF5

Безусловно, полученные при тестировании модельной версии библиотеки `psio` результаты интересно сравнить в сходной постановке с результатами, полученными при использовании HDF5.

Напомним, что данные в HDF5 представлены наборами данных (`dataset`). Каждый набор данных имеет имя, тип данных (скаляр, вектор, массив и пр.), тип элементов (целое, вещественное с указанной точностью). При работе с набором данных в программе создается представление объекта в памяти (`dataspace`). Набор данных (`dataset`) в HDF5 может быть многомерным, однако при сохранении этого набора данных в файл он должен быть записан как часть одномерного потока данных. Способ, которым многомерный массив размещается в файле, называется схемой размещения (`layout`). Самый очевидный способ размещения – сериализация, т.е. преобразование многомерного набора данных в одномерную последовательность байт, записываемую в файл большим монолитным блоком. Эта схема последовательного (`contiguous`) размещения данных на диске во многом повторяет схему размещения в оперативной памяти.

Альтернативой является кусочная (`chunked`) схема размещения [6], при которой `dataset` разделяется на логические куски (чанки), каждый из которых записывается в определенное место внутри файла. При этом чанки могут записываться и считываться в произвольном порядке. Благодаря этим свойствам «кусочной» схемы, её применение позволяет повысить производительность ввода-вывода на ПФС. Ценой этого улучшения обычно становится рост сложности программы из-за необходимости задействовать низкоуровневые возможности HDF5 и учитывать особенности размещения, выравнивания и кэширования данных. В связи с этим, в рамках данной статьи остановимся на последовательной схеме размещения наборов данных.

Доступ к массивам осуществляется через вырезки (`hyperslab`). Каждая вырезка описывается индексом начального элемента массива и длинам вырезки по измерениям. Размерность вырезки совпадает с размерностью массива. В операциях обмена используется объект `memspace`, описывающий массив в памяти, который имеет размерность (равную размерности массива или меньше ее) и длины по измерениям. Длины по измерениям `memspace` должны

совпадать с длинами измерений `dataspace`. В случае понижения размерности (из массива в операции обмена используется сечение) соответствующая длина по измерению `dataset` должна быть равна 1.

Общий порядок работы с данными в HDF5 при их создании следующий:

- создание файла,
- создание `dataspace` для набора данных (описание размерности массива),
- создание `memspace`, описывающей массив в памяти,
- создание набора данных (присвоение ему имени и выделение пространства в файле),
- определение вырезки на основании `dataspace` с указанием начального индекса массива в файле,
- выполнение операции записи массива из памяти в файл,
- закрытие набора данных,
- создание следующего набора данных,
- закрытие файла.

Общий порядок работы с данными в HDF5 при их чтении следующий:

- открытие файла,
- открытие набора данных,
- получение `dataspace` для набора данных из файла,
- создание `memspace`, описывающей массив в памяти,
- определение вырезки на основании `dataspace` с указанием начального индекса массива в файле,
- выполнение операции чтения массива из файла в память,
- закрытие набора данных,
- чтение следующего набора данных,
- закрытие файла.

4.2 Конфигурирование параллельного ввода-вывода

В параллельном режиме все функции HDF5 используются как коллективные операции MPI, поэтому процедура создания и открытия файла выполняется специальным образом. На листинге 2 в строках 11-12 показано задание параметров стрипования в HDF5 для ПФС Lustre.

```
1  MPI_info info;
2  MPI_Comm comm;
3  /* Создание атрибутов для открытия файла */
4
5  acc_tpl = H5Pcreate(H5P_FILE_ACCESS);
6  i = MPI_Info_create(&info)
7  char ss[20], sc[20];
8  sprintf(ss, "%d", stripe_size);
9  sprintf(sc, "%d", stripe_count);
10 /* Формирование значений stripe_size и stripe_count */
11 i = MPI_Info_set(info, "striping_unit", ss)
12 i = MPI_Info_set(info, "striping_factor", sc)
13 /* Настройка ROMIO на коммуникатор задачи и Lustre */
14 ret = H5Pset_fapl_mpio(acc_tpl, comm, info);
15 /* Создание файла с использованием заданных атрибутов */
16 file_id = H5Fcreate(fname, H5F_ACC_TRUNC, H5P_DEFAULT, acc_tpl);
17 /* Закрытие атрибутов файла */
```

```
18 H5Pclose(acc_tpl);
```

Листинг 2. Настройка программы на параллельный ввод-вывод в HDF5

Listing 2. Configuring a program for parallel I/O in HDF5

4.3 Используемая модель данных и реализация

Для максимального приближения к ранее используемой методике оценки производительности ввода-вывода используется следующая модель данных: базовым объектом является двумерный массив $M_{it}(R,BS)$, где:

- it – номер итерации,
- R – число процессов в задаче,
- BS – длина одиночной порции обмена.

Как и ранее, в подразделе 3.3, с процесса задачи в обмене участвует постоянный объем данных (1 Гбайт). Порция обмена (BS) варьируется от 1 до 512 Мбайт (соответственно используется от 1024 до 2-х итераций).

С процесса задачи в обмене участвует одномерный массив $A(BS)$, представляющий вырезку из массива M_{it} для строки с номером, равным номеру процесса.

Как и ранее, сначала производится формирование наборов данных (в цикле по итерациям запись с каждого процесса задачи соответствующей строки массива), потом использование наборов данных (в цикле по итерациям на каждом процессе задачи чтение соответствующей строки массива).

На листинге 3 приведен фрагмент исходного текста программы, выполняющей запись набора данных в HDF5.

```
1 dslen = len * mpi_size;
2 dimsf[0] = mpi_size; /* число стрк в файле */
3 dimsf[1] = len; /* ширина строки */
4 dimsm[0] = len; /* размер обмена */
5 dataspace = H5Screate_simple(2, dimsf, NULL);
6 memspace = H5Screate_simple(1, dimsm, NULL);
7 sprintf(dsname, "%s-%02d", dsnameproto, it);
8 dset_id = H5Dcreate(file_id, dsname, dstype, dataspace,
9 H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);
10 H5Sclose(dataspace);
11 count[0] = 1; count[1] = len; /*
12 размеры вырезки по измерениям */
13 offset[0] = rank; offset[1] = 0; /*
14 начало вырезки по измерениям */
15 dataspace = H5Dget_space(dset_id);
16 status = H5Sselect_hyperslab(dataspace, H5S_SELECT_SET,
17 offset, NULL, count, NULL);
18 data = (int *) malloc(sizeof(int)*dimsm[0]);
19 for (i=0; i < (int)dimsm[0]; i++) {
20 data[i] = rank * 10000 + it * 100 + i%100;
21 }
22 status = H5Dwrite(dset_id, dstype, memspace,
23 dataspace, H5P_DEFAULT, data);
24 fulllen += len * sizeof(int);
25 free(data);
26 H5Dclose(dset_id);
27 H5Sclose(dataspace);
28 H5Sclose(memspace);
29 H5Fflush(file_id, H5F_SCOPE_LOCAL);
```

Листинг 3. Запись набора данных в HDF5

Listing 3. Writing a dataset to HDF5

В строках 1-4 определяются необходимые размерности. В строках 5,6 создаются описание массивов в файле и памяти. В строках 8,9 создается набор данных для хранения массива. В строках 10-13 создается вырезка для работы с подмассивом. В строках 14-17 формируются данные в подмассиве. В строке 18 выполняется запись подмассива в набор данных. В строках 20-23 производится освобождение памяти и созданных объектов. В строке 24 производится сброс файловых кэшей.

4.4 Полученные результаты

В табл. 3 приведены полученные результаты. Методика проведения замеров такая же, как и в разд. 3. Обозначения в табл. 3 совпадают с обозначениями в табл. 1 и 2.

Видно, что параллельная версия HDF5 на данном тесте обеспечивает скорости, сравнимые в полученными ранее на тестах IOR и на структурированных файлах формата PSIO, однако немного уступает им.

Табл. 3. Пропускная способность ПФС на HDF5 на BK

Table 3. Cluster Lustre bandwidth with HDF5 data

BS	Nit	SS	Vw	Vr	Kw	Kr
1 поток, размер 1024 МБ						
1	1024	1	214.0	4827.2	0.5	0.8
2	512	1	266.4	6034.2	0.6	1.0
4	256	1	313.5	6768.3	0.7	1.1
8	128	1	370.9	6887.1	0.8	1.1
16	64	1	421.8	6483.8	0.9	1.1
32	32	1	462.0	5759.2	1.0	1.0
64	16	1	492.0	6012.7	1.1	1.0
128	8	1	544.5	6030.6	1.2	1.0
128	8	2	541.8	6017.6	1.2	1.0
128	8	4	544.0	5909.7	1.2	1.0
128	8	8	542.9	6036.1	1.2	1.0
256	4	8	566.9	5974.9	1.3	1.0
512	2	8	574.9	6057.8	1.3	1.0
2 потока, размер 2048 МБ						
1	1024	1	282.1	836.8	0.6	0.1
2	512	1	273.1	983.3	0.6	0.2
4	256	1	333.5	919.5	0.7	0.2
8	128	1	391.6	1001.6	0.9	0.2
16	64	1	501.0	991.4	1.1	0.2
32	32	1	643.6	1006.8	1.4	0.2
64	16	1	750.5	1053.7	1.7	0.2
128	8	1	812.0	918.0	1.8	0.2
128	8	2	806.6	989.6	1.8	0.2
128	8	4	813.5	1006.5	1.8	0.2
128	8	8	812.1	1171.1	1.8	0.2
256	4	8	885.6	1133.0	2.0	0.2
512	2	8	936.9	1185.5	2.1	0.2
4 потока, размер 4096 МБ						
128	8	8	1422.0	1832.1	3.2	0.3
256	4	8	1591.3	1851.0	3.6	0.3
512	2	8	1671.3	1822.3	3.7	0.3
8 потоков, размер 8192 МБ						
128	8	8	2792.1	2880.0	6.2	0.5
256	4	8	3310.9	3754.3	7.4	0.6
512	2	8	3849.6	4016.1	8.6	0.7

16 потоков, размер 16384 МБ						
128	8	8	3798.7	5773.4	8.5	1.0
256	4	8	4782.3	6136.9	10.7	1.0
512	2	8	5791.6	6470.2	12.9	1.1
32 потока, размер 32768 МБ						
128	8	8	6758.2	14288.6	15.1	2.4
256	4	8	7235.5	14261.0	16.2	2.4
512	2	8	9231.7	14338.9	20.6	2.4

Рассмотрим размещение объектов в файле для рассматриваемой задачи (см. рис. 4).

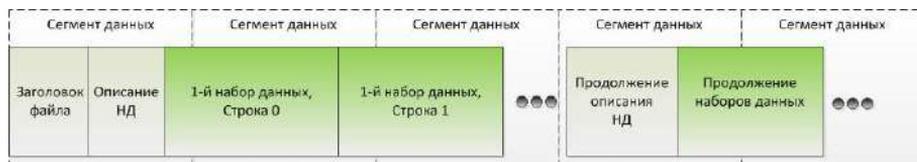


Рис. 4. Размещение данных в файле формата HDF5
Fig.4. The structure of HDF5 data

Как видно, наборы данных перемежаются метainформацией и практически невозможно разместить строки массивов в точных границах сегментов файла. Поэтому при обмене строки массива из процесса задачи с файлом зачастую затрагивается два сегмента файла, расположенных на разных ОСТ, что может приводить к замедлению выполнения операции обмена. Между тем библиотека HDF5 предоставляет приложению возможность запросить выравнивание всех объектов (с размером более некоторого размера) в файле с помощью вызова API функции `H5Pset_alignment` [7]. Это позволяет выравнивать информацию в файле, подлежащую вводу-выводу, на желаемую границу внутри ПФС. Но такие определенные особенности использования HDF5 совместно с ПФС Lustre являются темой отдельного исследования и в данной статье не рассматриваются.

5. Общие выводы и замечания

В краткой статье невозможно показать все особенности параллельного ввода-вывода для структурированных данных. Приведенные в статье примеры демонстрируют несколько вариантов реализации такого ввода-вывода с использованием стрипованных файлов ПФС Lustre. Для достижения желаемой производительности необходимо учитывать следующее:

- процессы задачи не должны одновременно обращаться к одному и тому же OST;
- в ходе операции обмена со стороны процессов задачи должно задействоваться минимальное число OST;
- блоки данных, участвующие в обмене, должны быть выровнены на границы сегментов файла в ПФС.

Разработанные в ходе исследований модельная версия формата данных PSIO и библиотеки `psio` позволяют реализовать параллельный доступ к структурированным данным с характеристиками ввода-вывода, сравнимыми с известной в мире библиотекой HDF5.

Авторы надеются, что данная статья окажется полезной разработчикам параллельных прикладных программ, используемых для проведения расчетов на вычислительных системах.

Список литературы / References

- [1] HDF Group. Available at: <https://www.hdfgroup.org/solutions/hdf5/>, accessed 10.05.2023.
- [2] Lustre. Available at: <https://www.lustre.org>, accessed 10.05.2023.
- [3] The IO500 benchmark. Available at: <http://www.io500.org>, accessed 10.05.2023.
- [4] Thakur R., Ross R. et al. Users Guide for ROMIO: A High-Performance, Portable MPI-IO Implementation. Argonne National Laboratory, Technical Memorandum no. 234, 2004, 17 p.

- [5] Национальный центр физики и математики. г. Саров / National Center for Physics and Mathematics. Sarov. Available at: <https://ncphm.ru/>, accessed 10.05.2023 (in Russian).
- [6] Chunking in HDF5. Available at: <https://support.hdfgroup.org/HDF5/doc/Advanced/Chunking/>, accessed 10.05.2023.
- [7] Howison M., Koziol Q. et al. Tuning HDF5 for Lustre File System. In Proc. of the Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS10), 2010, 11 p.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of Laboratory. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Сергей Юрьевич МОКШИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.

Дмитрий Владимирович ИВАНКОВ – начальник лаборатории. Сфера научных интересов: проектирование многоуровневых систем хранения данных, разработка высокопроизводительных систем хранения данных, исследования методов управления данными.

Dmitry Vladimirovich IVANKOV – Head of Laboratory. Research interests: design of tiered data storage systems, development of high performance storage systems, research in data management methods.

Евгений Александрович БЕКЕТОВ – начальник группы. Сфера научных интересов: проектирование систем хранения данных, разработка средств анализа и управления для систем хранения.

Evgeny Alexandrovich BEKETOV – Head of the work group. Research interests: data storage systems design, analytical methods for storage system management.

DOI: 10.15514/ISPRAS-2023-35(2)-9



Разработка и реализация средства диагностики сетей Ethernet

Г.П. Охрименко, ORCID: 0009-0003-3473-8492 <g.p.okhrimenko@vniitf.ru>

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

А.О. Игнатъев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

*Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13*

Аннотация. В данной работе приводится описание разработанных авторами средств анализа топологии сетей Ethernet, сбора, накопления и отображения статистики их работы. Описываются подходы по оценке качества работы сетевых устройств на основе статистики. Разработанное программное обеспечение используется для анализа работы сетей Ethernet в высокопроизводительных вычислительных системах, предназначенных для решения задач численного моделирования. Данная работа может быть полезна специалистам, занимающимся разработкой и эксплуатацией сетей Ethernet в вычислительных системах на основе ОС Linux.

Ключевые слова: вычислительная система; сети Ethernet; высокопроизводительные вычисления; топология сетей; диагностика сети.

Для цитирования: Охрименко Г.П., Мокшин С.Ю., Игнатъев А.О. Разработка и реализация средства диагностики сетей Ethernet. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 127-142. DOI: 10.15514/ISPRAS-2023-35(2)-9

Design and implementation of a diagnostic tool for Ethernet networks

G.P. Okhrimenko, ORCID: 0009-0003-3473-8492 <g.p.okhrimenko@vniitf.ru>

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

*E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia*

Abstract. This paper describes the tools developed by the authors for analyzing the topology of Ethernet networks, collecting, accumulating and displaying statistics of their work. Approaches for assessing the quality of network devices operation based on statistics are described. The developed software is used to analyze the operation of Ethernet networks in high-performance computing systems designed to solve numerical simulation problems. This work may be useful to specialists involved in the development and operation of Ethernet networks in computing systems based on the Linux operating system.

Keywords: computing system; structured data; Ethernet networks; high-performance computer; network topology; network diagnostic.

For citation: Okhrimenko G.P., Mokshin S.Yu., Ignatyev A.O. Design and implementation of a diagnostic tool for Ethernet networks. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 127-142 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-9

1. Введение

Необходимым условием надежной эксплуатации любой высокопроизводительной вычислительной системы (ВВС) является контроль за состоянием одной из важнейших её подсистем – коммуникационной подсистемы. Коммуникационная подсистема ВВС (КПС) – совокупность коммуникационного оборудования, линий связи и специализированного программного обеспечения (ПО), обеспечивающая передачу данных между компонентами ВВС [1]. КПС состоит из нескольких типов сетей: сети передачи сообщений, сети данных, сети управления, сети мониторинга и т.д. Может включать различные типы сетевого оборудования для выполнения разных целей (Ethernet, Infiniband, Omni-Path и др.). Сеть передачи сообщений и сеть данных, как правило, создаются на базе высокопроизводительных решений, наподобие Infiniband [2]. Сети управления и мониторинга ВВС, а также сети доступа строятся на базе оборудования Ethernet, могут быть функционально совмещёнными. В данной работе авторы рассматривают подходы к анализу сетевых топологий на уровне вычислительных центров, имеющих в своём составе одну и более ВВС, предназначенных для решения задач численного моделирования и объединённых сетями на базе технологии Ethernet.

Необходимым условием успешного функционирования сети Ethernet является наличие развитых средств её диагностики. Топология и статистика работы устройств сети формируются с использованием сетевого протокола SNMPv2c [3–11] посредством опроса и определения значений переменных ряда виртуальных баз данных, используемых для управления сетью Ethernet: IF-MIB [12], BRIDGE-MIB [13], EtherLike-MIB [14].

В работе рассматриваются:

- формирование топологии сетей и разработанные средства анализа топологии;
- выбор переменных виртуальных баз данных MIB, характеризующих работу и условия работы устройств сетей;
- система хранения статистических данных;
- разработанное в ФГУП РФЯЦ-ВНИИТФ (далее по тексту РФЯЦ-ВНИИТФ) программное обеспечение для сбора, накопления и отображения статистики о работе и условиях работы сетей;
- оценка качества работы сетевых устройств на основе статистики.

Основное назначение данной работы – осветить подходы к анализу топологии и работы сетей Ethernet, ознакомить с разработанным в РФЯЦ-ВНИИТФ программным продуктом для диагностики сетей Ethernet, а также предоставить системным программистам и техническим специалистам, обеспечивающим обслуживание сетевого оборудования, необходимый объём информации для сопровождения и развития рассматриваемого программного обеспечения, а также успешной эксплуатации сетей Ethernet.

2. Формирование топологии сети

Разработанный в РФЯЦ-ВНИИТФ в ходе эксплуатации различных ВВС программный комплекс MonE, предназначенный для анализа и мониторинга сетей Ethernet, позволяет администратору ВВС в полуавтоматическом режиме собирать и предоставлять топологию сетей Ethernet как в целом для всего вычислительного центра института, так и для каждой конкретной ВВС. Данный программный комплекс состоит из множества программных компонент, выполняющих разнообразные функции (подпрограмм), реализованных на языке Python [15].

Основой рассматриваемых программных средств анализа сетей Ethernet является формируемая MonE таблица в базе данных (БД) устройств сети – TAB_IP_MAC. Каждая запись в данной базе представляет параметры отдельного интерфейса сети:

- IP-адрес интерфейса;
- имя интерфейса;
- MAC-адрес интерфейса.

Для заполнения БД используется подпрограмма из состава MonE – PingArpChk. Подпрограмма инициирует выполнение команды ping (с параметром «-с 1») по всем IP-адресам заданного сегмента сети (например, со значениями 0 – 255 младшего байта IP-адреса). На этом этапе формируется словарь NAME_HOSTS, ключами которого являются обрабатываемые IP-адреса, а значениями – соответствующие им доменные имена интерфейсов. Результат работы этой подпрограммы заносится в таблицу TAB_IP_MAC. Отметим, что если подпрограмма PingArpChk вызывается для уже сформированного в таблице сегмента сети, то она производит сравнение полученной информации с ранее сохраненной, сигнализируя о возникших расхождениях.

Формирование топологии сети Ethernet фактически сводится к определению связей между коммутаторами сети и доменных имён интерфейсов, подключённых к портам коммутаторов. Исходными для данной работы являются описанная выше таблица TAB_IP_MAC и формируемая отдельно таблица TAB_SW1, которая содержит требуемую информацию о каждом коммутаторе (сетевое имя, тип, IP-адрес, число портов). Информация о коммутаторах сетей и их функционировании извлекается из информационных баз данных этих коммутаторов IF-MIB, BRIDGE-MIB, EtherLike-MIB с использованием сетевого протокола SNMPv2.

Определение MAC-адресов, подключённых к портам интерфейсов, осуществляется на основе следующей информации. Коммутаторы содержат список портов, к каждому из которых прикреплен список объектов (заданных MAC-адресами интерфейсов), с которыми данный коммутатор логически связан по данному порту. Под логической связью подразумевается способность «увидеть» другой узел сети или порт другого коммутатора через определённый порт. Частным случаем логической связи является физическая связь – прямое соединение коммутатора с узлом или другим коммутатором посредством Ethernet-кабеля. Отметим, что один порт коммутатора может быть связан только с одним объектом.

Коммутаторы поддерживают динамическую таблицу переадресации AFT, хранящую соответствие MAC-адреса интерфейса узла или MAC-адреса порта другого коммутатора порту данного коммутатора. В используемых типах коммутаторов эта таблица хранится в ассоциативно-запоминающем устройстве CAM. Когда кадры поступают на порты коммутатора, MAC-адрес источника запоминается и записывается в таблицу CAM. Порт, на который были получены кадры, и сеть VLAN записываются в таблице вместе с меткой времени. Если MAC-адрес, изученный одним портом коммутатора, был перемещён на другой порт, записывается MAC-адрес и метка времени того порта, который получил кадры последним. Предыдущая запись удаляется. Если MAC-адрес для правильного приёмного порта уже содержится в таблице, то обновляется только метка времени.

Эта информация доступна по протоколу SNMPv2 в MIB-базе BRIDGE-MIB коммутатора. Для определения доменных имён интерфейсов, подключённых к портам коммутаторов сетей Ethernet, в составе MonE создана подпрограмма GetIfConn. Используя MAC-адреса интерфейсов, получаемых при чтении БД TAB_IP_MAC, подпрограмма формирует выходную информацию о доменных именах интерфейсов, подключённых к портам коммутатора. На рис. 1 приведены данные, полученные подпрограммой GetIfConn для некоторого коммутатора B01ETG-1 BBC. Таблицы подключения абонентов формируются для всех коммутаторов сетей Ethernet.

Порт	Параметры подключенного интерфейса или порта коммутатора		
	Название	IP-адрес	MAC-адрес
Gi0/1	b01s01	1.2.1.1	0c:xx:xx:xx:0a:ee
Gi0/2	b01s02	1.2.1.2	0c:xx:xx:xx:5b:5e
Gi0/3	b01s03	1.2.1.3	0c:xx:xx:xx:5e:7c
...			
Gi0/40	b01s40	1.2.1.40	0c:xx:xx:xx:5b:38
Gi0/48	B01ET1 Port 48	-	0c:xx:xx:xx:22:ee
Ex0/3	ET10G-1 Ex0/2	-	0c:xx:xx:xx:5e:9b

Рис. 1. Информация о подключении устройств к портам коммутатора B01ETG1
 Fig.1. Information about connecting devices to the switch ports

Сложность определения топологии сетей Ethernet отмечена рядом авторов, причастных к этому направлению работ [16–19]. В нашем случае эта сложность состоит в том, что таблицы переадресации коммутаторов динамические, хранят запись соответствия MAC-адреса назначения и соответствующего ему порта некоторое ограниченное время, заданное в конфигурации устройства, и на момент исследования таблицы не имеют полной информации о всех доступных сетевых устройствах и их связях. Поэтому получение актуальной информации о топологии сети требует периодического анализа доступных в её топологии устройств. В нашем случае это периодическое использование подпрограмм PingArpChk и GetIfConn для пополнения информации об интерфейсах, подключённых к портам коммутаторов.

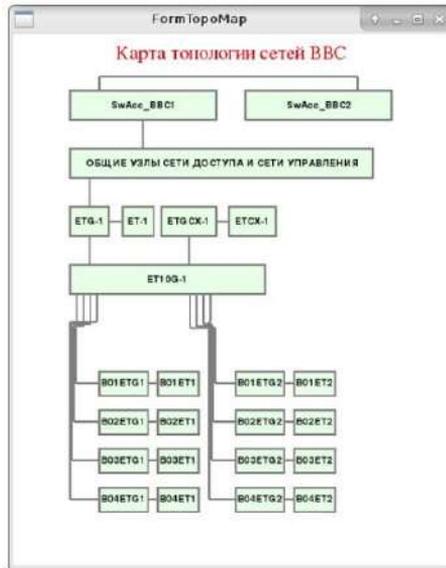


Рис. 2. Карта топологии сетей Ethernet для выбранной ВВС
 Fig.2. Map of the Ethernet network topology for the selected HPC

3. Средство анализа топологии сетей Ethernet

Графический интерфейс программного комплекса MonE реализует функции построения карты топологии сетей, как показано на рис. 2, присваивая элементам карты (изображениям коммутаторов и линиям связи) соответствующие теги. Тегом коммутатора является имя коммутатора, тегом линии связи – идентификаторы связываемых портов (<имя коммутатора>*<имя порта>). При этом существует возможность просмотра развернутой информации по каждому коммутатору, представленному в топологии (рис. 3). В данном случае можно увидеть все устройства, подключенные к портам данного коммутатора.

Кроме того, на карте существует возможность просмотра линий связи между коммутаторами, как показано на рис. 4.



The screenshot shows a window titled 'TabPortsDesign' with a sub-header 'SwAcc_BBC1'. It contains a table with the following data:

Порт	Параметры подключённого интерфейса или порта коммутатора		
	Название	IP-адрес	MAC-адрес
Gi1/0/1	f1.acc	1.1.74.1	0c:xx:xx:xx:9e:5a
Gi1/0/2	f2.acc	1.1.74.2	0c:xx:xx:xx:9c:12
Gi1/0/3	f3.acc	1.1.74.3	0c:xx:xx:xx:96:ee
Gi1/0/4	f4.acc	1.1.74.4	ac:xx:xx:xx:8c:76
Gi1/0/6	a1.acc	1.1.73.17	0c:xx:xx:xx:35:6e
Gi1/0/7	a2.acc	1.1.73.18	0c:xx:xx:xx:a7:f2
Gi1/0/10	serv1.acc	1.1.75.1	0c:xx:xx:xx:9e:b6
Gi1/0/11	serv2.acc	1.1.75.2	ac:xx:xx:xx:8d:6a
Te1/0/2	SwAcc_BBC2 Ex0/38	-	00:xx:xx:xx:5a:e1

Рис. 3. Информация о коммутаторе на карте топологии сети
Fig. 3. Information about the switch on the network topology map



The screenshot shows a window titled 'GetTagsDesign' with a sub-header 'Теги линии связи ET10G-1 — ETG-1'. It contains a table with the following data:

Теги линии связи	Параметры соединяемых интерфейсов	
	Название	MAC-адрес
ET10G-1*Ex0/1_ETG-1*Ex0/3	ET10G-1, порт Ex0/1	00:xx:xx:xx:5e:9a
ETG-1*Ex0/3_ET10G-1*Ex0/1	ETG-1, порт Ex0/3	00:xx:xx:xx:37:47

Рис. 4. Фрагмент вывода карты линий связи между коммутаторами
Fig.4. The fragment map of communication lines for switches

4. Сбор и накопление статистики

Наиболее эффективный метод диагностики коммутируемой сети – запрос информации о поведении сети у самих коммутаторов. В рамках работ по данному направлению реализован SNMPv2-агент, работающий с информационными базами IF-MIB и EtherLike-MIB коммутаторов.

4.1 Выбор переменных MIB для анализа работы и состояния устройств

Для оценки загрузки устройств сетей используются переменные из информационной базы IF-MIB. Статистика загрузки устройств по приёму данных формируется с использованием переменных:

- ifInOctets (ifHCInOctets);
- ifInUcastPkts (ifHCInUcastPkts);
- ifInBroadcastPkts (ifHCInBroadcastPkts);
- ifInMulticastPkts (ifHCInMulticastPkts).

В скобках приведены названия соответствующих переменных, которые представляются 64-разрядными счётчиками (символы «HC» после «if» в названии переменных означают: High Capacity). Переменные, отображающие число ошибок, представляются 32-разрядными счётчиками.

Статистика загрузки устройств по передаче данных формируется аналогичными переменными, но относящимся к выходным данным. Переменные IF-MIB `ifInNUcastPkts` и `ifOutNUcastPkts` не используются, так как значение каждой из них представляет сумму соответствующих значений переменных `ifInBroadcastPkts` и `ifInMulticastPkts`, `ifOutBroadcastPkts` и `ifOutMulticastPkts`. Для анализа статистики сбоев устройств используются следующие переменные IF-MIB: `ifInErrors`, `ifInDiscards`, `ifOutErrors`, `ifOutDiscards`.

Необходимо сказать несколько слов о выборе используемых переменных EtherLike-MIB. Применительно к стандарту Ethernet организация локальных сетей возможна с топологией «общая шина» или «звезда». Обычно логическая топология Ethernet – «общая шина», где среда передачи данных используется всеми узлами в одном сегменте сети. При этом, если несколько узлов в одной среде начнут вместе передавать информацию, то возникнет конфликт при передаче данных, который приведёт к их повреждению и невозможности дальнейшего использования. Чтобы не допустить подобной ситуации, для управления общим доступом узлов в Ethernet используется метод множественного доступа с прослушиванием несущей и обнаружением коллизий (CSMA/CD) [20]. Информационная база EtherLike-MIB включает ряд переменных, с помощью которых фиксируются ошибки, присущие указанной топологии Ethernet: `SingleCollisionFrames`, `MultipleCollisionFrames`, `LateCollisions`, `ExcessiveCollisions`, `CarrierSenseErrors` (здесь и ниже в начале названий переменных EtherLike-MIB мы опускаем наименование анализируемой группы переменных – `dot3Stats`). В рассматриваемой топологии сетей коммутаторы работают в полнодуплексном режиме, т.е. могут посылать и получать данные одновременно, поэтому в рассматриваемых сетях коллизии исключаются. Таким образом, отмеченные выше переменные в список используемых нами не включаются.

Не используется также переменная `SQETestError`, которая определяет число сообщений `SQE TEST ERROR`, генерируемых тестом `SQE`. Тест `SQE` предназначен для обнаружения цепи с коллизиями между трансивером и платой сетевого интерфейса. Поскольку у сетевых плат теперь есть встроенный трансивер, тест `SQE` не требуется. Поэтому мы игнорируем и эту переменную [21].

Кроме того, не используется переменная `DefferedTransmissions`, так как она представляет те же ошибки, что и переменная IF-MIB `ifOutDiscards`.

С учётом сказанного, для анализа статистики сбоев устройств целесообразно использовать следующие переменные (счётчики) EtherLike_MIB:

- `AlignmentErrors`;
- `FCSErrors`;
- `FrameTooLongs`;
- `SymbolErrors`;
- `InternalMacReceiveErrors`;
- `InternalMacTransmitErrors`.

Все приведённые выше переменные представляют накапливаемые значения, начиная с момента инициализации SNMP-агента.

Таким образом, для анализа сети в программе `MonE` используются переменные из информационных баз IF-MIB и EtherLike-MIB, представленные в таблице 1.

Табл. 1. Используемые переменные IF-MIB и EtherLike-MIB и их назначение
Table 1. IF-MIB and EtherLike-MIB variables used and their purpose

№ п/п	Название переменной	Назначение переменной
1	<code>ifHCInOctets</code>	Общее количество принятых портом байт данных, включая служебные

2	ifHCOutOctets	Общее количество переданных портом байт данных, включая служебные
3	ifHCInUcastPkts	Количество пакетов с индивидуальным адресом интерфейса, доставленных портом на верхний системный уровень
4	ifHCOutUcastPkts	Количество пакетов с индивидуальным адресом интерфейса, полученных портом с верхнего системного уровня
5	ifHCInBroadcast-Pkts	Количество пакетов с широковещательным адресом интерфейса, доставленных портом на верхний системный уровень
6	ifHCOutBroadcast-Pkts	Количество пакетов с широковещательным адресом интерфейса, полученных портом с верхнего системного уровня
7	ifHCInMulticast-Pkts	Количество пакетов с групповым адресом интерфейса, доставленных портом на верхний системный уровень
8	ifHCOutMulticast-Pkts	Количество пакетов с групповым адресом интерфейса, полученных портом с верхнего системного уровня
9	ifInErrors	Количество принятых портом пакетов, которые не были переданы на верхний системный уровень из-за обнаружения в них ошибок
10	ifOutErrors	Количество выходных пакетов, которые не были переданы портом из-за наличия в них ошибок
11	ifInDiscards	Количество пакетов, которые были приняты портом, оказались корректными, но не были доставлены на верхний системный уровень, скорее всего из-за переполнения буфера пакетов или же по иной причине
12	ifOutDiscards	Количество отброшенных портом выходных пакетов, в том числе и тех, в которых не были обнаружены ошибки; т.е. количество пакетов, потерянных из-за нехватки мощностей или перегрузок
13	AlignmentErrors	Количество ошибок выравнивания – определяется количеством кадров, полученных портом, длина которых не равна целому числу октет и которые имеют неверную контрольную сумму
14	FCSErrors	Количество кадров, полученных портом, длина которых равна целому числу октет, но которые имеют неверную контрольную сумму
15	FrameTooLongs	Количество кадров, полученных портом, размер которых превышает максимально допустимый для Ethernet
16	SymbolErrors	Для интерфейса, работающего в полнодуплексном режиме – количество случаев, когда средства приёма заняты в течение времени, равному или большему времени приёма кадра минимального размера и в течение которого на канальном уровне была зафиксирована ошибка приёма данных
17	InternalMac-ReceiveErrors	Количество кадров, приём которых не состоялся из-за внутренней ошибки передачи подуровня MAC
18	InternalMac-TransmitErrors	Количество кадров, передача которых портом не выполнялась из-за ошибки, переданной внутренним MAC-подуровнем

4.2 Хранение накопленной статистики

Анализ состояния сетей Ethernet BVC производится ежедневно. При этом данные сохраняются в БД в виде таблиц и записей. Во всех таблицах БД названия записей соответствуют именам коммутаторов, статистика работы которых отображается в этих записях. Обычно для крупных BVC статистика сохраняется 11 – 12 месяцев. В таблицах 2 и 3 дана краткая характеристика информации, содержащейся в таблицах БД.

Табл. 2. Таблицы суточной статистики и характеристика хранимой информации
 Table 2. Tables of daily statistics and characteristics of stored information

Имя таблицы	Характеристика информации
LOAD_ERR	Суточная статистика загрузки и сбоев портов коммутатора, запрашиваемая с использованием переменных IF-MIB. Загрузка портов представлена значениями переменных: ifHCInOctets, ifHCOutOctets, ifHCInUcastPkts, ifHCOutUcastPkts, ifHCInBroadcastPkts, ifHCOutBroadcastPkts, ifHCInMulticastPkts, ifHCOutMulticastPkts; а сбой – значениями переменных: ifInErrors, ifOutErrors, ifInDiscards, ifOutDiscards
ERR_eth	Суточная статистика ошибок и сбоев портов коммутатора, запрашиваемая с использованием переменных EtherLike-MIB: AlignmentErrors, FCSErrors, FrameTooLongs, SymbolErrors, InternalMacReceiveErrors, InternalMacTransmitErrors
SUM_ERR	Суточная статистика ошибок и сбоев всех типов только тех портов коммутатора, которые имеют ошибки и сбои

Табл. 3. Таблицы статистики последнего часа и характеристика содержащейся в них информации
 Table 3. Statistics tables of the last hour and characteristics of the information they contain

Имя таблицы	Характеристика информации
LAST	Последние значения обрабатываемых переменных IF-MIB в коммутаторе
LAST_eth	Последние значения обрабатываемых переменных EtherLike-MIB в коммутаторе
LH_LOAD_ERR	Загрузка и сбой портов коммутатора за последний час, запрашиваемые с использованием переменных IF-MIB
LH_ERR	Статистика ошибок и сбоев всех типов за последний час только тех портов коммутатора, которые имеют ошибки и сбои

4.3 Отображение суточной и текущей статистики

Для представления статистических данных в MonE используется подпрограмма отображения и анализа статистики Map. Окно подпрограммы, представленное на рис. 5, содержит ряд кнопок, позволяющих выбрать требуемую дату, интервал выборки и тип отображаемой информации, а также представляет обобщённую схему (карту) сетей Ethernet VBS, отображающую сбойные связи сетей. Элементы схемы используются для выбора требуемых устройств или линий связи при анализе статистики загрузки и сбоев портов.

При нажатии кнопки «Сбой за сутки» выводится за выбранный интервал времени статистика сбоев портов всех коммутаторов сетей. В левой части окна формируется таблица сбоев, которая отображает суммарное за выбранный интервал количество сбоев каждого типа для «сбойных» портов, как показано на рис. 6, а в правой части – отображаются графики сбоев для всех портов, как показано на рис. 7.

При построении графиков сбоев используется логарифмическая шкала.

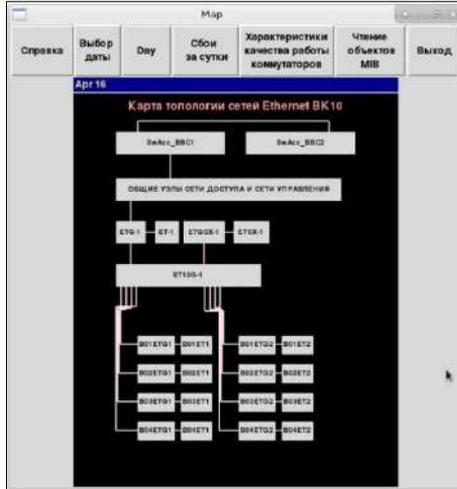


Рис. 5. Окно подпрограммы Map
Fig.5. The Map subroutine window

Сбои портов коммутаторов - Апрель 16

sw	port	if-name	1	2	3	4	5	6	7	8	9	10
B01	ET G1	Ex0/3	ET10G-1			1						
B01	ET G2	Ex0/3	ET10G-1			54						
B02	ET G1	Ex0/3	ET10G-1			34						
B02	ET G2	Ex0/3	ET10G-1			70						
B03	ET G1	Ex0/3	ET10G-1			35						
B03	ET G2	Ex0/3	ET10G-1			24						
B04	ET G1	Ex0/3	ET10G-1			62						
B04	ET G2	Ex0/3	ET10G-1			33						
ET10G-1	Ex0/2	B01ETG1			26892							
ET10G-1	Ex0/3	B01ETG2			10							
ET10G-1	Ex0/0	B03ETG1			4							
ET10G-1	Ex0/3	B04ETG1			30							
ET10G-1	Ex0/0	B04ETG2			2							
ETGCX-1	Fa0/14	ip1_83_4.m	1									
ETGCX-1	FD/47	backmas					29					
SwAcc_BBC1	G11/0/4	f4_acc					30375					
SwAcc_BBC2	Ex0/3	SwAcc_BBC1			1102							

Рис. 6. Левый фрагмент окна отображения суточной статистики портов коммутаторов
Fig.6. The left fragment of the window for displaying daily statistics of switch ports



Рис. 7. Правый фрагмент окна отображения суточной статистики портов коммутаторов
Fig.7. The right fragment of the window for displaying daily statistics of switch ports

На рис. 8 и 9 приведены фрагменты окна отображения загрузки и сбоев портов некоторого выбранного коммутатора ETG-1.

ETG-1		#InOctets	#OutOctets	#InBeastPkts	#OutBeastPkts	E	f	r	o	r	s	1	2	3	4	5	6	7	8	9	10		
Gi0/1	f1.lbe1	427 714 396	1 008 114 051	4 711 094	4 971 475	5 088	2 298 802																
Gi0/2	f2.lbe1	11 365 875	234 121 941	76 456	159 901	4 144	2 299 738																
Gi0/3	f3.lbe1	6 326 638	377 939 120	45 042	200 263	1 026	2 302 858																
Gi0/4	f4.lbe1	9 019 386	243 518 420	98 285	181 683	306	2 303 578																
Gi0/6	a1.lbe1	751 078 824	4 125 737 399	7 739 843	9 509 713	1 812	2 302 072																
Gi0/7	a2.lbe1	1 962 230 297	1 617 638 717	2 202 659	1 081 874	2 303 884	132 931																
Gi0/11	serv1.lbe1	111 963 196	284 232 160	1 294 242	1 303 758	2 239	2 301 645																
Gi0/12	serv2.lbe1	111 943 561	284 544 434	1 295 560	1 305 459	2 754	2 301 139																

Рис. 8. Фрагмент окна, отображающий суточную статистику загрузки и сбоев портов коммутатора

Fig.8. Fragment of the window showing daily statistics of loading and failures of ports of the switch



Рис. 9. Фрагмент окна, отображающий суточную статистику загрузки и сбоев портов коммутатора

Fig.9. Fragment of the window showing daily statistics of loading and failures of ports of the switch

Также можно отобразить загрузку требуемых портов коммутатора в течение заданного интервала времени. Для этого в верхней части области графиков расположены кнопки, позволяющие выбрать нужный час суток.

5. Оценка качества работы сетей на основе статистических данных

Действующие технические нормы на показатели функционирования сетей передачи данных, принятые в 2007 году [22], определяют предельные значения показателей качества: среднюю задержку при передаче пакетов, отклонение от неё, коэффициент потери пакетов, коэффициент ошибок в пакетах, как показано в табл. 4.

Табл. 4. Технические нормы на показатели функционирования сетей передачи данных
Table 4. Technical standards for performance indicators of data transmission networks

Наименование показателя	Тип передаваемого трафика			
	интерактивный ¹	сигнальный	поточковый ²	трафик передачи данных за исключением интерактивного, сигнального и потокового
Средняя задержка передачи пакетов информации (мс)	не более 100	не более 100	не более 400	не более 1000
Отклонение от среднего значения задержки передачи пакетов информации (мс)	не более 50	–	не более 50	–
Коэффициент потери пакетов информации	не более 10^{-3}	не более 10^{-3}	не более 10^{-3}	не более 10^{-3}
Коэффициент ошибок в пакетах информации	не более 10^{-4}	не более 10^{-4}	не более 10^{-4}	не более 10^{-4}
Примечания: ¹ Интерактивный трафик – тип трафика, для которого характерно непосредственное взаимодействие (диалог) пользователей услуги связи или пользовательского (оконечного) оборудования. ² Поточковый трафик – тип трафика, для которого характерен просмотр и (или) прослушивание информации по мере её поступления в пользовательское (оконечное) оборудование.				

5.1 Выбор и определение характеристик качества работы коммутаторов

Важно отметить, что стандарты измерения качества услуг в сетях Ethernet в настоящее время отсутствуют, а существующие рекомендации международных институтов не избавляют от изрядной доли неопределённости при выборе условий и средств измерения [23].

Возникает вопрос: «Как оценить качество работы сетей на основе получаемой статистики?» Например, в статье «Практическая диагностика локальных сетей» [24] приведены пороговые значения ряда характеристик сети, превышение которых свидетельствует о плохой работе сети. При этом значимыми характеристиками для оценки состояния коммутаторов являются следующие.

- Количество ошибочных пакетов E_p (%) – процент пакетов, отброшенных портом из-за

наличия в них ошибки. Для порта коммутатора, подключённого к полнодуплексному сегменту, пороговое значение этой характеристики составляет 0,001%.

- Количество широковещательных пакетов B_p (%) – процент пакетов, адресованных одновременно всем узлам (по отношению к общему числу обработанных портом пакетов). Такие пакеты используются, например, для поиска определённых ресурсов сети, для перевода IP-адресов в MAC-адреса. Большая доля широковещательного трафика снижает производительность сети, так как соответствующие пакеты вынуждены обрабатывать все её узлы. Считается, что пороговое значение этой характеристики составляет 6%. Однако, когда загрузка сети низкая, то даже если доля широковещательного трафика высока, это не ухудшает работу её пользователей. Поэтому считается, что рассматриваемую характеристику следует учитывать лишь в том случае, когда уровень загрузки порта коммутатора не менее 5%.
- Загрузка порта L_p (%) – процентное отношение числа байт, прошедших по сегменту за 1 секунду, к максимально возможной скорости передачи данных по этому сегменту. Для порта коммутатора пороговое значение этой характеристики составляет 50%. Когда для данной характеристики фиксируется плохая оценка, то имеет место несоответствие пропускной способности сети потребностям пользовательских приложений. В этом случае имеет смысл выяснить, какие приложения и какие пользователи создают высокую загрузку портов оборудования.

Приведённые выше пороговые значения характеристик работы портов коммутаторов мы используем для оценки состояния коммутаторов на основе почасовой статистики их работы.

При формировании процента ошибочных пакетов E_p учитываем, что переменные информационной базы MIB2 `ifInErrors` и `ifOutErrors` фиксируют в сумме общее число отброшенных портом ошибочных пакетов, а используемые переменные из EtherLike-MIB представляют число этих же пакетов по типам ошибок.

Исходя из этого, процент отвергнутых пакетов E_p определяем на основе двух характеристик – E_{p_in} и E_{p_out} , присваивая значению E_p большее из этих значений:

$$E_{p_in} = \frac{E_{in} \times 100}{p_in} (\%),$$
$$E_{p_out} = \frac{E_{out} \times 100}{p_out} (\%),$$
$$E_p = \max[E_{p_in}, E_{p_out}],$$

где E_{p_in} – процент ошибочных пакетов, отвергнутых при их приёме в течение часа; E_{in} – значение часового приращения переменной `ifInErrors`; p_in – сумма принятых за час пакетов всех групп адресации; E_{p_out} – процент ошибочных пакетов, отвергнутых при их передаче в течение часа; E_{out} – значение часового приращения переменной `ifOutErrors`; p_out – сумма переданных за час пакетов всех групп адресации.

Пороговое значение этой характеристики принимаем равным 0,001%.

Число отвергнутых портом безошибочных пакетов за часовой промежуток определяется приращениями значений переменных `ifInDiscard` и `ifOutDiscard`.

Процент отвергнутых безошибочных пакетов D_p определяем на основе двух характеристик – D_{p_in} и D_{p_out} , присваивая значению D_p большее из этих значений:

$$D_{p_in} = \frac{D_{in} \times 100}{p_in} (\%),$$
$$D_{p_out} = \frac{D_{out} \times 100}{p_out} (\%),$$
$$D_p = \max[D_{p_in}, D_{p_out}],$$

где D_{p_in} – процент безошибочных пакетов, отвергнутых при их приёме; D_{in} – значение часового приращения переменной `ifInDiscard`; p_{in} – сумма принятых за час пакетов всех групп адресации; D_{p_out} – процент безошибочных пакетов, отвергнутых при их передаче; D_{out} – значение часового приращения переменной `ifOutDiscard`; p_{out} – сумма переданных за час пакетов всех групп адресации.

Пороговое значение D_p принимаем равным 0,01%.

Уровень загрузки порта L_p – процентное отношение числа байт, проходящих через порт в среднем за секунду в течение часового интервала, к максимально возможной скорости передачи данных по этому порту – определяем на основе двух характеристик: L_{p_in} и L_{p_out} , присваивая значению L_p большее из этих значений:

$$L_{p_in} = \frac{InOctets \times 8 \times 100}{S_p \times 1600} (\%),$$
$$L_{p_out} = \frac{OutOctets \times 8 \times 100}{S_p \times 1600} (\%),$$
$$L_p = \max[L_{p_in}, L_{p_out}],$$

где L_{p_in} – уровень часовой загрузки порта приёмом пакетов; $InOctets$ – число принятых байт в течение часа; L_{p_out} – уровень часовой загрузки порта передачей пакетов; $OutOctets$ – число переданных байт в течение часа; S_p – максимальная скорость передачи данных по исследуемому порту (бит/с).

Значения S_p для всех портов коммутаторов Ethernet определены посредством чтения переменной `ifSpeed` из MIB-2.

Пороговое значение L_p принимаем равным 50%.

Процент широковещательных пакетов B_p определяем на основе двух характеристик B_{p_in} и B_{p_out} , присваивая значению B_p большее из этих значений:

$$B_{p_in} = \frac{InBcast \times 100}{p_{in}} (\%),$$
$$B_{p_out} = \frac{OutBcast \times 100}{p_{out}} (\%),$$
$$B_p = \max[B_{p_in}, B_{p_out}],$$

где B_{p_in} – процент широковещательных пакетов, принятых портом в течение часа; $InBcast$ – значение часового приращения переменной `ifInBroadcastPkts`; p_{in} – сумма принятых портом в течение часа пакетов всех групп адресации; B_{p_out} – процент широковещательных пакетов, переданных портом в течение часа; $OutBcast$ – значение часового приращения переменной `ifOutBroadcastPkts`; p_{out} – сумма переданных портом в течение часа пакетов всех групп адресации.

Пороговое значение B_p принимаем равным 6%. Однако, как отмечалось выше, когда загрузка сети низкая, то даже высокая доля широковещательного трафика не ухудшает работу её пользователей. Поэтому считается, что рассматриваемую характеристику следует учитывать лишь в том случае, когда уровень загрузки порта коммутатора не менее 5%.

5.2 Отображение характеристик качества работы коммутаторов

В разработанной программе MonE реализована функция анализа характеристик качества работы коммутаторов за сутки. Как показано на рис. 10, окно представления качества работы конкретного коммутатора состоит из двух частей. В левой части окна представлена таблица, отображающая для каждого порта выбранного коммутатора суммарную за сутки загрузку

каждого типа (8 типов), количество ошибочных пакетов (InErrors и OutErrors), количество отвергнутых безошибочных пакетов (InDiscard и OutDiscard), а также максимальные за сутки значения рассмотренных выше характеристик качества (E_p , D_p , L_p , B_p). В правой части, как показано на рис. 11, отображаются графики указанных в таблице параметров для всех портов коммутатора. Для удобства пользователя при отображении характеристик качества, их названия соответственно заменяются: E – MaxErr, D – MaxDiscd, L – MaxLoad, B – MaxBcast. При построении графиков используется логарифмическая шкала. Имеется возможность выделения всех графиков, относящихся к выбранному порту, и возможность определения значений отображаемых параметров в любой точке графика.

HealthSW															
Характеристики качества работы портов коммутатора SwAcc_BBC1 Apr 16															
port	#-name	#InOctets	#OutOctets	#InUcastPkts	#OutUcastPkts	#InBcastPkts	#OutBcastPkts	#InMcastPkts	#OutMcastPkts	SwInErrPkts	SwInDiscdPkts	MaxErr	MaxDiscd	MaxLoad	MaxBcast
												%	%	%	%
Gi1/0/1	f1.acc	6 237 775 752	247 714 211	5 424 251	2 044 520	123 674	95 501							0.52	11.0
Gi1/0/2	f2.acc	89 127 174	1 061 067 132	762 746	1 285 145	123 674	95 501							0.09	53.0
Gi1/0/3	f2.acc	5 506 384 546	918 840 520	5 445 242	3 497 446	123 674	95 501			45				0.26	43.0
Gi1/0/4	f4.acc	22 994 243 882	385 610 943 330	38 185 028	259 352 043	123 663	95 500			30 375		0.22	53.8	10.0	
Gi1/0/6	a1.acc	47 391	18 837 081	275	21 478	123 671	95 501	5							59.0
Gi1/0/7	a2.acc	237 793 303	324 423 415	1 109 806	2 033 081	123 263	95 501	471						0.01	11.0
Gi1/0/10	serv1.acc	1 028	761 357	14	898	4 998	4 142								50.0
Gi1/0/11	serv2.acc	34 346	18 449 785	391	21 318	123 663	95 501			367		3.11			59.0
Te1/0/2	SwAcc_BBC2	123 329 457 963	46 748 337 509	87 648 078	42 421 855	1 737	117 020	65 844	28 162					2.51	19.0

Рис. 10. Левая часть окна характеристик для коммутатора SwAcc_BBC1 за выбранные сутки
 Fig.10. The left part of the characteristics window for the SwAcc_BBC1 switch for the selected day

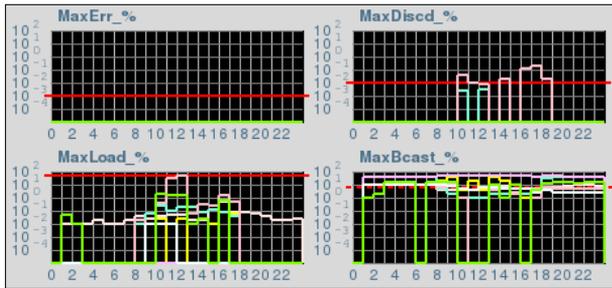


Рис. 11. Правая часть окна характеристик для коммутатора SwAcc_BBC1 за выбранные сутки
 Fig.11. The right part of the characteristics window for the SwAcc_BBC1 switch for the selected day

Для коммутатора SwAcc_BBC1 видно, что пороговые значения превышены на портах Gi1/0/4 и Gi1/0/11, к которым подключены сервер доступа f4.acc и сервисный узел serv2. Видимо, это связано со сбоями в работе портов коммутатора или сетевых адаптеров указанных узлов. Как видно из рис. 11 (график Max Discd %), это происходит в основном в рабочее время, когда нагрузка на сеть максимальна.

6. Заключение

Разработанный и представленный в данной работе специализированный программный комплекс для анализа топологии сетей Ethernet работает на ВВС РФЯЦ-ВНИИТФ с января 2019 года. Применение этого программного комплекса даёт возможность техническим специалистам и администраторам вести оперативное наблюдение за работой сетей Ethernet ВВС, своевременно выявлять критические области в сетях и осуществлять локализацию источников сбоев или ошибок.

Разработанные и входящие в состав программного комплекса программные компоненты предоставляют следующий набор инструментов для сетевой диагностики:

- средства анализа топологии сетей;
- средства анализа загрузки и сбоев портов коммутаторов;
- средства анализа состояния коммутаторов;
- средства отображения значений доступных переменных MIB в коммутаторах.

Программный комплекс позволяет обнаруживать отклонения в работе сети, связанные с плохой работой сетевого оборудования или паразитными нагрузками вследствие неправильно работающего программного обеспечения. Полученная информация позволяет ремонтно-обслуживающему персоналу ВВС устранить ошибки и, тем самым, повысить качество работы сетей Ethernet ВВС.

Список литературы / References

- [1] Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования. Препринт РФЯЦ-ВНИИТФ no. 265, Снежинск, 2020 г., 21 стр. / Ignatyev A.O., Mokshin S.Yu. Base architecture of the mathematical modelling HPC system. Preprint RFNC-VNIITF no 265, Snezhinsk, 2020, 21 p. (in Russian).
- [2] InfiniBand. Available at: <https://www.nvidia.com/ru-ru/networking/products/infiniband/>, accessed 10.05.2023.
- [3] RFC-1901. Introduction to Community-based SNMPv2. Available at: <https://www.ietf.org/rfc/rfc1901.txt>, accessed 10.05.2023.
- [4] RFC-1902. Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1902.txt>, accessed 10.05.2023.
- [5] RFC-1903. Available at: Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1903.txt>, accessed 10.05.2023.
- [6] RFC-1904. Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1904.txt>, accessed 10.05.2023.
- [7] RFC-1905. Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1905.txt>, accessed 10.05.2023.
- [8] RFC-1906. Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1906.txt>, accessed 10.05.2023.
- [9] RFC-1907. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2). Available at: <https://www.ietf.org/rfc/rfc1907.txt>, accessed 10.05.2023.
- [10] RFC-1908. Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework. Available at: <https://www.ietf.org/rfc/rfc1908.txt>, accessed 10.05.2023.
- [11] RFC-1909. An Administrative Infrastructure for SNMPv2. Available at: <https://www.ietf.org/rfc/rfc1909.txt>, accessed 10.05.2023.
- [12] RFC-2863. The Interfaces Group MIB. Available at: <https://www.ietf.org/rfc/rfc2863.txt>, accessed 10.05.2023.
- [13] RFC-1493. Definitions of Managed Objects for Bridges. Available at: <https://www.ietf.org/rfc/rfc1493.txt>, accessed 10.05.2023.
- [14] RFC-3655. Definitions of Managed Objects for the Ethernet-like Interface Types. Available at: <https://www.ietf.org/rfc/rfc3655.txt>, accessed 10.05.2023.
- [15] Python. Available at: <https://www.python.org/>, accessed 10.05.2023.
- [16] Gobjuka H., Breitbart Y. Finding Ethernet-Type Network Topology is Not Easy. Technical Report TR-KSU-CS-2007-03, Kent State University, 2007, 9 p.
- [17] Pandey S., Choi M.-J. et al. IP Network Topology Discovery Using SNMP. In Proc. of the International Conference on Information Networking, 2009, 5 p.
- [18] Воеводин Вад.В, Стефанов К.С. Автоматическое определение и описание сетевой инфраструктуры суперкомпьютера. Вычислительные методы и программирование, том 15, вып. 4, 2014 г., pp. 560-568 / V. Vad. Voevodin, Stefanov K.S. Automatic detection and description of supercomputer network infrastructure. Numerical Methods and Programming, vol. 15, issue 4, 2014, pp. 560-568 (in Russian).
- [19] Поляков В. Определение топологии сети на уровнях 2/3 модели OSI. Блог компании Tibbo / Polyakov V. Determination of network topology at levels 2/3 of the OSI model. Tibbo Blog. Available at: <https://habr.com/ru/companies/tibbo/articles/257911/>, accessed 10.05.2023 (in Russian).

- [20] Sen S., Choudhury R. Multiple Access with Collision Detection (CSMA/CD). In Proc. of the 16th Annual International Conference on Mobile Computing and Networking, 2010, 12 p.
- [21] Конфигурация сети EtherNet/IP. Руководство пользователя / EtherNet/IP Network Configuration. User Manual. Available at: https://www.literature.rockwellautomation.com/idc/groups/literature/documents/um/enet_um001_-ru-p.pdf, accessed 10.05.2023 (in Russian).
- [22] Приказ об утверждении требований к организационно-техническому обеспечению устойчивого функционирования сети связи общего пользования. Мининформсвязь, 27.09.2007, № 113 / Order on approval of the requirements for the organizational and technical support of the sustainable operation of the public communication network. Miniformsvyaz, 09/27/2007, № 113. Available at: <https://www.zakonbase.ru/content/base/111612>, accessed 10.05.2023 (in Russian).
- [23] Орлов С. Мониторинг параметров качества в операторских сетях Ethernet. Журнал сетевых решений/LAN, 2013 г., вып. 12 / Orlov S. Monitoring of quality parameters in operator Ethernet networks. Journal of Network Solutions/LAN, 2013, issue 12. Available at: <https://www.osp.ru/lan/2013/12/13038691>, accessed 10.05.2023 (in Russian).
- [24] Юдицкий С., Кузубов С. Практическая диагностика локальных сетей / Yuditsky S., Kuzubov S. Practical diagnostics of local networks. Available at: <https://old.prolan.ru/company/article/magazine/mpk042004.html>, accessed 10.05.2023 (in Russian).

Информация об авторах / Information about authors

Григорий Павлович ОХРИМЕНКО – старший научный сотрудник. Сфера научных интересов: проектирование топологии коммуникационных систем, разработка программного обеспечения для диагностики сетей.

Grigoriy Pavlovich OKHRIMENKO – Senior Researcher Research interests: communication systems design, network topology design, network diagnostics software development.

Сергей Юрьевич МОКШИИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of Laboratory. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

DOI: 10.15514/ISPRAS-2023-35(2)-10



Approach to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure

E.V. Geraskin, ORCID: 0009-0007-2599-1460 <GeraskinEvgeniy@mail.ru>

N.V. Voinov, ORCID: 0000-0002-0140-1178 <voinov@ics2.ecd.spbstu.ru>

*Peter the Great St.Petersburg Polytechnic University,
29, Polytechnicheskaya, St.Petersburg, 195251, Russia*

Abstract. The article is devoted to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure. The main idea is to develop a software framework to create and perform end-to-end tests written in Python. The approach involves a comprehensive evaluation of the system from the user interface to the database. The testing process is performed in a continuous integration environment, which enables the team to test the system continually as new code is added. The testing process also includes the use of automated tests written in Python. The automated tests allow for faster and more reliable testing and enable the team to test the system across multiple platforms and configurations. The approach also includes the use of virtual environments to simulate the production environment. This enables the team to identify potential issues that may arise in the production environment and to test the system's performance under various conditions.

Keywords: ent-to-end testing; enterprise virtual infrastructure; configuration; software framework; Python.

For citation: Geraskin E.V., Voinov N.V. Approach to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 143-156. DOI: 10.15514/ISPRAS-2023-35(2)-10

Разработка подхода к сквозному тестированию приложения для управления конфигурацией виртуальной инфраструктуры предприятия

Е.В. Гераскин, ORCID: 0009-0007-2599-1460 <GeraskinEvgeniy@mail.ru>

Н.В. Воинов, ORCID: 0000-0002-0140-1178 <voinov@ics2.ecd.spbstu.ru>

*Санкт-Петербургский политехнический университет Петра Великого,
Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29*

Аннотация. Статья посвящена сквозному тестированию приложения для управления конфигурацией виртуальной инфраструктуры предприятия. Основная идея заключается в разработке программной среды для создания и выполнения сквозных тестов, написанных на Python. Подход включает всестороннюю оценку системы от пользовательского интерфейса до базы данных. Процесс тестирования выполняется в среде непрерывной интеграции, что позволяет команде постоянно тестировать систему по мере добавления нового кода. Процесс тестирования также включает использование автоматизированных тестов, написанных на Python. Автоматизированные тесты обеспечивают более быстрое и надежное тестирование и позволяют команде тестировать систему на нескольких платформах и в разных конфигурациях. Подход также включает использование виртуальных сред для имитации производственной среды. Это позволяет команде выявлять потенциальные проблемы, которые могут возникнуть в производственной среде, и тестировать производительность системы в различных условиях.

Ключевые слова: сквозное тестирование; виртуальная инфраструктура предприятия; конфигурация; программная основа; Python

Для цитирования: Гераскин Е.В., Воинов Н.В. Разработка подхода к сквозному тестированию приложения для управления конфигурацией виртуальной инфраструктуры предприятия. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 143-156. DOI: 10.15514/ISPRAS-2023-35(2)-10

1. Introduction

End-to-end testing [1] is a software testing technique that involves evaluating a system from end to end, covering every component and process, from the user interface to the database. The goal of end-to-end testing is to verify the system's functionality, performance, reliability, and security, among other aspects. End-to-end testing ensures that all parts of the system work together as expected and that there are no inconsistencies or errors that might affect the system's performance.

One of the main advantages of end-to-end testing is that it can identify potential issues that might not be detected by other testing techniques, such as unit testing or integration testing. For example, if there is a problem with the interaction between different components of the system, it may not be evident in isolation, but it could have significant consequences for the system's performance as a whole. End-to-end testing can identify such issues and help developers address them before the system is deployed.

Another benefit of end-to-end testing is that it can help reduce the time and effort required to identify and rectify errors. By testing the system as a whole, issues can be located and resolved earlier in the development process, before they become more challenging and expensive to fix. This, in turn, can help reduce the overall cost of development and improve the time-to-market for the product [2].

Moreover, conducting end-to-end testing can contribute to enhancing the product's quality and customer satisfaction. By verifying that the system functions as intended and performs well, end-to-end testing can help ensure that users have a positive experience when using the product. This improves customer loyalty and generates positive feedback.

End-to-end testing is a critical software testing technique that is essential for verifying the functionality, performance, reliability, and security of a system. By testing the system from end to end, potential issues can be identified and addressed early in the development process, leading to a higher-quality product that performs well and satisfies customers' needs. That is why it is necessary to use the end-to-end testing paradigm as an axiom, and build the testing process around it.

Similar to the software development process, end-to-end testing also follows a specific methodology. In this case, methodology refers to the principles, ideas, methods, and concepts that engineers employ while working on a project. There are currently several diverse approaches to end-to-end testing, each with its own starting points, duration of execution, and methods used at each stage. Choosing the right approach can be a challenging task, and it requires an understanding of the unique features and requirements of the system being tested.

This article focuses on the approach to end-to-end testing of a specialized software used for managing the configuration of enterprise virtual infrastructure. Later in text «application» is used to define this software. Detailed architecture of this application is described in section 2.

The approach involves a comprehensive evaluation of the system from the user interface to the database. The testing process is performed in a continuous integration environment, which enables the team to test the system continually as new code is added. The testing process also includes the use of automated tests written in Python. The automated tests allow for faster and more reliable testing and enable the team to test the system across multiple platforms and configurations.

The approach also includes the use of virtual environments to simulate the production environment. This enables the team to identify potential issues that may arise in the production environment and to test the system's performance under various conditions.

2. Features of the application under test and end-to-end testing

Application for managing the configuration of enterprise virtual infrastructure consists of three main components (Fig.1):

- Client-side: a part of the application that handles user requests and interacts with Ansible playbooks via API;
- Configuration management scripts module: Ansible playbooks for setting and applying specified configuration to target system;
- Target infrastructure system (server, virtual machine or PC with already installed operation system, etc.)

There are many approaches to end-to-end testing and some suggest that testing should be done in three stages with each component tested separately. Often, different engineers with different knowledge, skills, and competencies conduct each stage of testing. However, this approach has some disadvantages. Firstly, it can take a considerable amount of time to complete all three stages of testing, especially if the gap between each stage is long. This can lead to delays in the development process and make it more challenging to fix any issues that are identified.

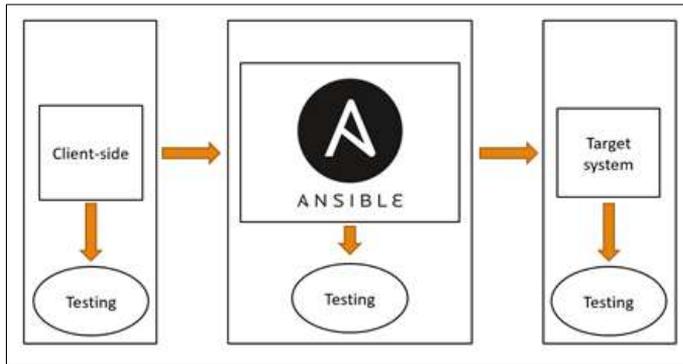


Fig. 1. The main components of the application under test

Another issue with this approach is that it can be challenging to identify the root cause of any issues that arise. If an error is identified in one component, it can be challenging to determine whether the issue is specific to that component or if it is related to another part of the system. This can lead to a significant amount of time spent on troubleshooting and can further delay the development process. Moreover, the three-stage approach to end-to-end testing may not capture all the possible interactions and dependencies between the different components of the system. This can result in issues being missed, which can lead to unexpected behavior when the system is deployed in a production environment. Additionally, this approach can be costly since it requires a significant amount of time and resources to execute and maintain.

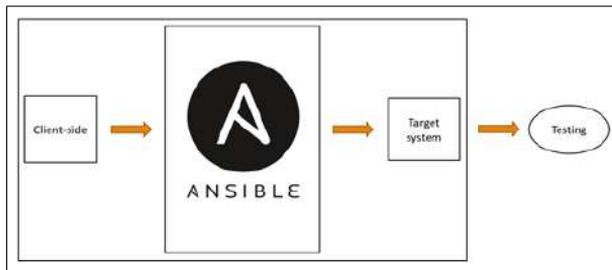


Fig. 2. Approach to end-to-end testing of the considered application

Therefore, for end-to-end testing of the considered application an approach is needed that will not break the system into separate components but will consider the entire system as a whole (Fig.2).

Thus, minimizing the time complexity during the process of end-to-end testing is crucial. Moreover, the entire testing process can be carried out and monitored by a single engineer.

End-to-end testing may encounter several issues, including complexity, time-consuming nature, difficulty in reproducing errors, inconsistency, debugging challenges, high cost, and so on.

Complexity is one of the primary challenges of end-to-end testing, especially when testing complex systems. Testing multiple components of such systems makes the testing process challenging. Tests may also consume a lot of time, making frequent testing difficult [3].

Another problem that may arise during end-to-end testing is difficulty in reproducing errors. End-to-end testing can make it difficult to identify the cause of an error when it arises due to testing the entire system. Furthermore, end-to-end tests can be inconsistent, meaning they may fail repeatedly for various reasons. This can be caused by different factors such as network issues, browser version mismatch, and race conditions.

Debugging end-to-end tests can indeed be a challenging task, especially when a test fails. Identifying the root cause of the failure can be problematic, as the issue may be located in any of the various components of the system. This can make it difficult to isolate the source of the problem and resolve it. As a result, engineers may spend a significant amount of time diagnosing and fixing issues, which can increase the overall development time.

Moreover, end-to-end testing requires a considerable amount of hardware and software resources. This is because the testing process involves the comprehensive evaluation of the system from the user interface to the database, which requires a substantial amount of computational power.

Maintaining end-to-end tests can also be challenging over time, especially when the system being tested evolves and changes. In complex systems, end-to-end testing may not cover all possible scenarios, leading to limited coverage. Tests may also provide insufficient feedback to developers as it may be challenging to pinpoint the exact location of a problem in the system.

Finally, end-to-end testing can be too costly. Conducting end-to-end tests may require the use of a large number of hardware and software resources as well as the need for an adequate number of qualified specialists to write and maintain tests.

Overall, end-to-end testing has its advantages and disadvantages, and its effectiveness depends on several factors, including system complexity, the number of resources available for testing, and the experience of the engineering team. Understanding these issues can help developers and testers create a more effective testing strategy to ensure system quality and reliability.

When developing software and tests, one of the generally accepted methodologies is used:

- TDD (Test Driven Development) [4] is a software development methodology that is based on repeating short development cycles: initially, a test is written that covers the desired change, then program code is written that implements the desired behavior of the system and allows the written test to pass. Then, the written code is refactored with constant checking of the passing of tests.
- TDD (Type Driven Development) [5] is based on types. In this case, data types and type signatures serve as a specification for the program. Types also serve as a form of documentation that is guaranteed to be updated.
- BDD (Behavior Driven Development) [6] involves describing user scenarios in natural language by testers or analysts.
- DDD (Domain Driven Design) [7] is a set of rules that allow for making the right design decisions. This approach significantly speeds up the process of designing software in an unfamiliar domain.
- FDD (Features Driven Development) [8] attempts to combine the most recognized software development methodologies in the industry, based on important functionality (properties) of the developed software for the customer. The main goal of this methodology is to systematically develop real, working software within the set deadlines.

It was decided to use BDD, as it is the only approach that involves obtaining natural language test documentation as output. Additionally, this methodology allows a single engineer (tester) to independently carry out the entire end-to-end testing cycle, from writing test scenarios to creating a

report on the results of testing. This solves one of the main problems - involving multiple engineers with different sets of knowledge and competencies in testing.

When conducting end-to-end testing, specialized tools are usually used for this purpose. Of course, there are more or less universal tools, but it should be understood that the intricacies of end-to-end testing may differ even among software systems operating in the same domain. Therefore, for each system, its own approach and tool for end-to-end testing are usually developed. Moreover, implementation may vary depending on the skills or preferences of engineers, as well as the specifics of the software. This can be a full-fledged application with a graphical interface, a console program, a framework that is embedded in the project with automated tests, and so on.

For example, within the scope of end-to-end testing of V2X (Vehicle-to-Everything) systems [3], a distributed application is used, each node of which emulates a real microcontroller. Moreover, it is necessary to emulate not only the "hardware", but also a specialized message exchange protocol. V2X is a technology that provides communication between vehicles and other objects, such as infrastructure and pedestrians. It should be understood that the criticality of an error in such a system can cost someone's life. Therefore, in such systems, testing tools must be as close as possible to the real environment: using a minimum of mock objects, distributed, well supported and updated. All these factors make end-to-end testing tools very heavy for development, support, and financing, so such tools should only be used when there is an urgent need. Otherwise, resources spent on development and support will be wasted.

Speaking of universal cross-functional testing tools, the most well-known ones are Jaeger and Zipkin [9]. These two very similar tools provide functionality for fairly detailed identification of system failures in a software system. These tools have many advantages: they are easy to set up thanks to detailed documentation and ease of use, they are suitable for almost any software system, provide a wide range of functionality and a graphical interface, which means that there is no need to spend time developing and supporting this functionality.

However, the universality of these tools also has its drawbacks. Jaeger and Zipkin do not provide functionality for executing test scenarios, i.e., testing will still have to be done manually, but the tools will help to identify the location of the error more easily and quickly. The results of the tools' work may be difficult for non-technical people (such as management) to understand, so all reports will also have to be written manually.

Also, both tools are quite "heavy", which may require additional hardware resources for deployment. Jaeger and Zipkin do not support all programming languages, so if the system is written in a specific language, it may simply not be possible to configure the tools.

Finally, these two tools may pose certain risks in terms of information security: if access is configured incorrectly or improperly, various sensitive information may leak. Thus, Jaeger and Zipkin are not without drawbacks, but are still good tools for cross-functional testing, but only as additional control systems. Nevertheless, the functionality does not allow for complete control of cross-functional testing only through these tools.

One of the most popular cross-functional testing tools today is a software tool in the form of a framework for a programming language on which engineers write automated tests. This approach is relevant not only for applications that perform mobile computing [10], but also for other systems from completely different subject areas. The idea of implementing such frameworks is as follows: for a test framework of some programming language (e.g., the Pytest framework for the Python language), a wrapper is written that allows desired actions to be performed with the system under test. Creating a cross-functional testing software framework can offer several advantages:

- Flexible customization. With a custom testing platform, one can adapt their testing to the specific needs of their project, providing greater flexibility and control over the testing process.
- Reusability. A custom framework can be reused in multiple projects or teams, saving time and effort in the long run.

- Integration opportunities. Integrating a custom testing platform with other tools and systems can help simplify and automate the testing process.
- Cost savings. Creating a custom framework may be cheaper than purchasing, learning, and implementing a commercial testing tool or framework.
- Learning opportunities. Creating a custom framework provides opportunities for one's team to learn as they develop and refine their skills in software development and testing.

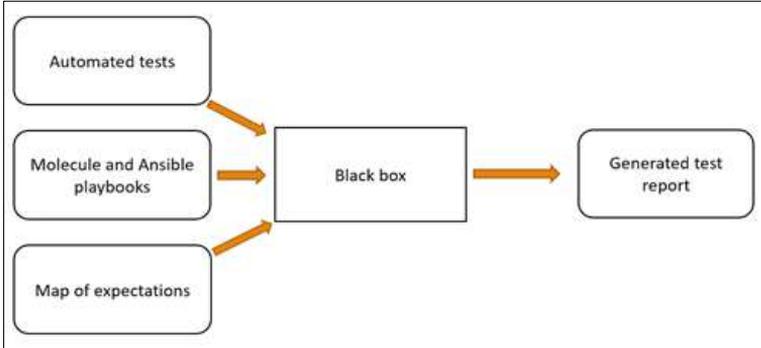


Fig. 3. System-level design

3. The proposed approach

To arrive at the picture shown in Fig.2, an approach to end-to-end testing was developed that can be applied by a single engineer and minimizes downtime between testing of system components.

Behavior Driven Development (BDD) [11] methodology is used for end-to-end testing of both individual components and the system as a whole. BDD is a development approach based on behavior description, where an engineer writes descriptions such as "As a user, when I click the Start button, the menu should be displayed as shown in the picture." Classic development with tests follows. BDD involves engineers describing user scenarios in natural language. Thus, the output will include not only end-to-end test scenarios and their results, but also natural language descriptions that can be used for documentation or reporting. Moreover, a person who is not familiar with the technical implementation of the system can understand the results because all results will be described in natural language.

One of the main advantages of BDD is that this approach focuses on business needs rather than technical implementation details. Tests are written in natural language that everyone can understand, which improves communication and minimizes possible misunderstandings. This approach allows customers and developers to determine what needs to be tested without additional costs for translating technical documentation.

BDD also leads to improved test coverage of the product because this methodology aims to cover all possible application usage scenarios. Based on these tests, potential issues can be easily identified before they become real problems for users.

Furthermore, BDD enables the creation of tests that are easier to maintain in the future. This is because changes in the application require changes in all related tests, so the easier the tests are to maintain and modify, the easier it is to make changes to the application itself.

However, the BDD methodology has its own limitations. BDD can be quite difficult to understand, especially for new developers and testers. It may require them to learn new terminology and testing approaches.

BDD may require additional effort during development to create functional requirements that can be used to write tests. This can take additional time.

Additionally, BDD may require workers to put in significant effort to understand the behavior of the application. This may require a lot of training, practice, and time, which may be unrealistic for some companies at present.

Integration with tools can be challenging, especially with multiple tools at once, to run BDD tests. This may require additional costs to maintain tool settings up-to-date.

However, the main problem with applying BDD in end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure is the presence of components such as configuration management script modules and target infrastructure systems.

In the classical application of BDD, it is suggested to cover end-to-end tests only for the client-side, which is written in commonly used programming languages such as Java, C#, Python, etc. However, it is unclear how to deal with configuration management tools (Ansible/Puppet) and even more unclear how to apply BDD to end-to-end testing of server or virtual machine states. To address this issue, a software tool was developed, which is a Python-based framework that allows the application of BDD to specific components for this approach.

4. Implementation

IT companies develop specialized frameworks (including for testing) to address various tasks within their systems, taking into account the specifics of the product and providing the necessary tools for interacting with it. Since automated testing of the client-side is implemented using the Python framework Pytest, a software tool was developed that is a framework for the Python and allows the application of BDD to specific components for this approach.

Functional requirements for the developed software:

- The software shall prepare the infrastructure on which end-to-end testing will be performed (in this case, creating virtual machines), and after completion, release resources (i.e., delete virtual machines).
- The software shall provide the ability to test the client-side of the application.
- The software shall provide the ability to test configuration management scripts.
- The software shall provide the ability to test the state of the target system after applying the necessary configuration.

The software shall generate a summary report on the results of end-to-end testing in natural language.

The system-level design [12] of the developed software can be described by the diagram shown in Fig. 3.

The developed framework is essentially a versatile black box that can take in automated tests, Ansible playbooks, Molecule scripts, and a map of expectations as inputs, all written in Python. These inputs are then used to execute the end-to-end tests on the system under test (SUT). Testing in this case means, for example, checking the correctness of the scenario for installing security updates on a group of virtual machines.

One of the key advantages of this framework is that it is easily customizable and extensible, allowing users to tailor it to their specific testing requirements. Users can add their own test cases and test scripts, as well as customize the test environment and configuration to suit their needs.

Upon execution, the framework automatically generates a test report in Allure, a flexible and open-source platform for test reporting. The report includes comprehensive descriptions of the test cases in natural language, making it easy for users to understand and interpret the test results.

The software solution developed for this approach is written in Python, with a specially configured Jenkins job as the user interface. The Pytest and Molecule tools are responsible for end-to-end testing of the entire system, on which a module for applying the BDD approach to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure was created.

The developed software solution is highly flexible and versatile framework for Python, which can be easily integrated into any project with automated end-to-end testing requirements. It provides a seamless and well-structured interface for writing and executing end-to-end tests, using a variety of inputs such as autotests, Ansible playbooks, Molecule scripts, and a map of expectations.

To help visualize the inner workings of the end-to-end testing system, a UML sequence diagram has been developed, which captures the interrelation of all components involved in the testing process (Fig. 4). This diagram illustrates the flow of events that take place during the execution of the end-to-end tests, starting with the initialization of the test environment and ending with the generation of the test report.

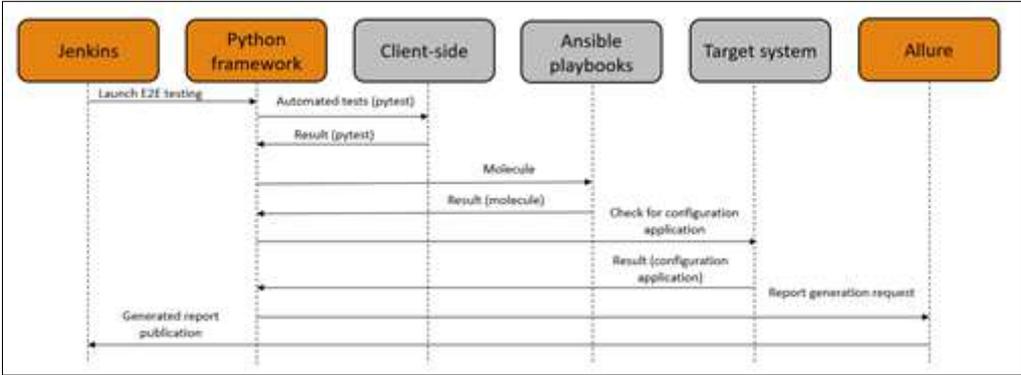


Fig. 4. UML sequence diagram

The diagram shows how each component of the system interacts with the others, providing a clear and concise overview of the entire testing process. It highlights the critical role played by the developed framework in managing the end-to-end testing process, as well as the importance of other components such as the Ansible playbooks and Molecule scripts in setting up the test environment and ensuring the correct behavior of the system under test.

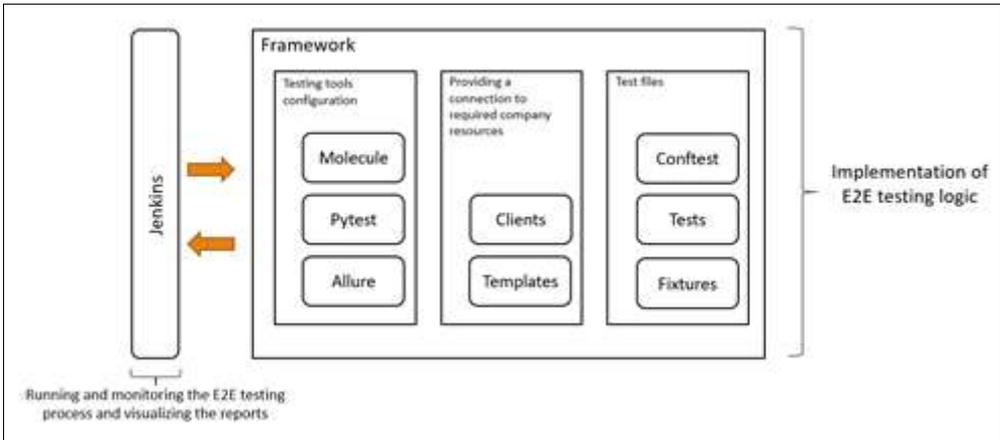


Fig. 5. High-level design of the developed software

The primary modules of the application for managing the configuration of enterprise virtual infrastructure are highlighted in gray, while the modules of the developed software solution are highlighted in orange. Jenkins provides the user interface through which the engineer configures and launches the end-to-end testing process. Subsequently, the Python framework executes the test scenarios sequentially for each module: the client-side (using Pytest), Ansible scripts (using Molecule), and the target system (checking its initial and final states). The results are aggregated into a report; which Allure generates in natural language. The final report is published in Jenkins,

where reports for each of the end-to-end testing process runs are stored. The interrelation of all components of the final end-to-end testing system can be described by a UML sequence diagram.

The architecture is shown in Fig. 5.

The framework comprises three primary components:

- A module for preparing test data and deploying the test infrastructure, which includes fixtures – objects that can be considered a set of conditions required for the test to run. For example, fixtures are often created to generate data before the test starts and return it for use in the test or before the test. This module also handles the preparation of the test infrastructure (creating virtual machines) and the preparation of test data such as IP addresses of virtual machines, login credentials, operating system names, etc.
- The module that implements methods for accessing the necessary services of the company during testing. This module creates conditions for testing that replicate the production environment: to correctly use virtual machines created in the previous module, they shall be properly registered with third-party services developed by other teams. Additionally, this module is responsible for verifying the correctness of the end-to-end testing process with respect to other systems within the company (answering the question “Did our system break another team’s system?”).
- The module for configuring testing tools and reporting. This module implements the configuration of the main testing tools: Pytest for the client-side, Molecule for configuration management, and shell scripts for checking the state of virtual machines. It also generates a summary report of the test results in natural language and loads it into Jenkins for visualization.

The Jenkins job plays a crucial role in implementing the user interface for starting the end-to-end testing process and visualizing the results, as shown in Fig. 6. It serves as a platform for managing the entire testing process, from the initial setup and configuration of the testing environment to the execution of tests and the generation of reports.

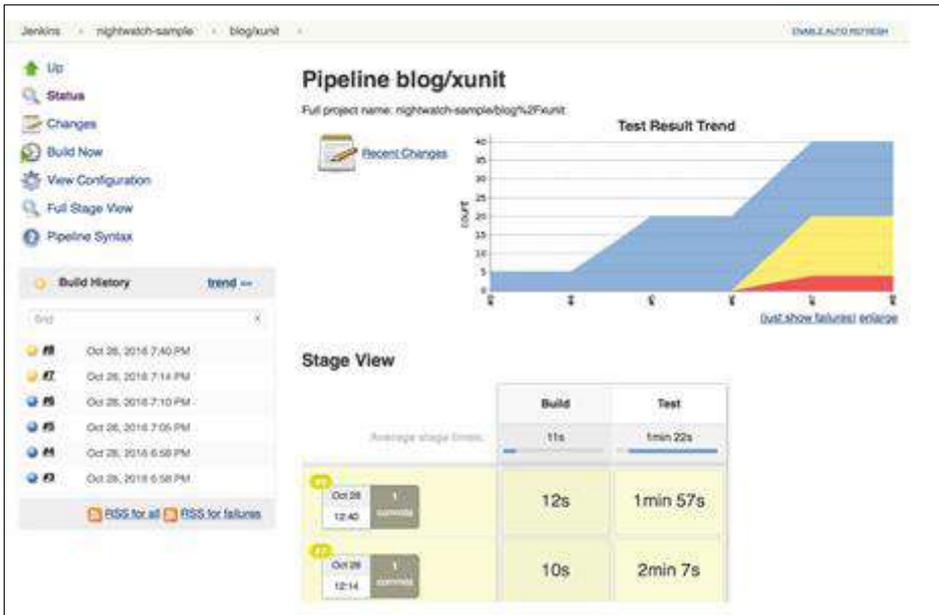


Fig. 6. Jenkins job

Through the Jenkins interface, users can initiate the testing process and monitor its progress in real-time, enabling them to quickly identify and address any issues that may arise. Additionally, Jenkins can be configured to automatically trigger tests based on predefined conditions or events, further streamlining the testing process and reducing the likelihood of errors or oversights.

The visualization of results in the Jenkins interface provides a clear and concise overview of the testing process and its outcomes. Users can easily interpret the results and identify any areas of concern, allowing them to take prompt corrective action and ensure the software meets the necessary quality standards.

Through the Jenkins functionality, it is possible to configure the parameters of end-to-end testing: determine the number of virtual machines, select desired operating systems, configure the system under test, etc.

A summary format displays the last few runs of end-to-end testing, whether they were successful or not, and if not - at what stage an error occurred. Additionally, it is possible to evaluate a graph of the number of successful, skipped, and unsuccessful test scenarios.

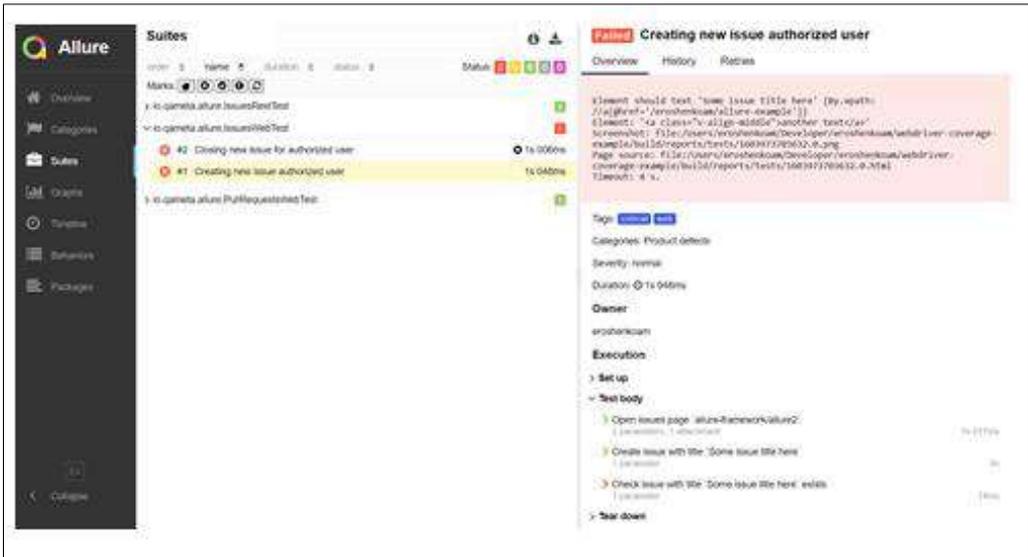


Fig. 7. Test report

The final generated test report can be also accessed through the Jenkins job interface and consists of a set of test scenarios with various input data described in formal language steps, expected results, and the success of executing a particular scenario (Fig. 7).

5. Results

An end-to-end testing approach for the application for managing the configuration of enterprise virtual infrastructure was developed, and a Python framework with a Jenkins user interface and natural language test report generation in Allure was created to implement and apply this approach. The metric chosen to evaluate the usefulness of the new cross-functional testing approach was the time required to execute a single test run on operating system update functionality: checking correct credentials uploading, correct operating system update results and correct deletion of temporary files after update. Ten experiments were conducted using the developed framework, and ten experiments were conducted using the old approach (with step-by-step testing of each module). It should be noted that downtime between stages in the old approach was not included in the metric, i.e., only the time of the testing process was taken into account.

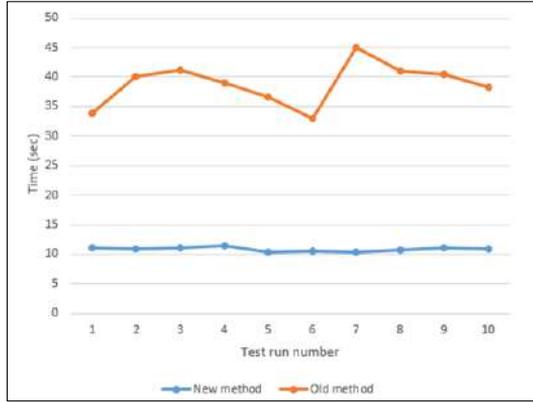


Fig. 8. Results of the first experiment

Initially, an infrastructure consisting of one virtual machine was used. On average, using the developed framework requires 10–11 seconds per test run, while the old approach requires 30 to 45 seconds (Fig. 8).

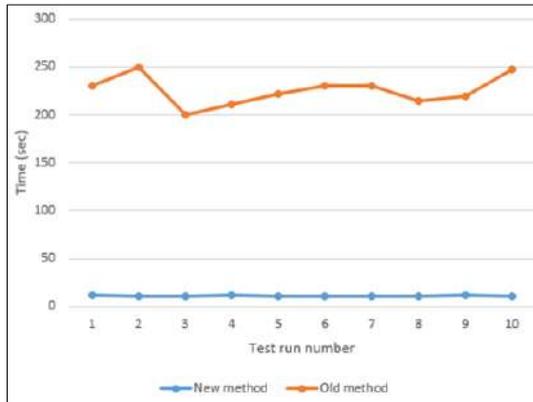


Fig. 9. Results of the second experiment

Next, the size of the infrastructure was increased to ten virtual machines, and the same experiment was conducted. The difference was more significant. The time required for a single test run using the developed framework remained at 10–11 seconds. However, the old approach showed a much longer execution time for a single test run - from 200 to 250 seconds (Fig. 9).

The results obtained from the evaluation of the new approach to end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure were statistically significant and can be explained through the principle of parallel operations. The use of the developed Python framework and its support for parallelism led to a significant reduction in the time required to execute a single test run compared to the old approach with step-by-step testing of each module.

The statistical analysis of the experimental results using a t-test showed a significant difference ($p < 0.01$) between the mean execution times of the old and new approaches. This indicates that the new approach is superior in terms of efficiency, as it enables parallel operations and thereby reduces the time required for end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure.

The principle of parallel operations is widely recognized in computer science and engineering as an effective means of improving the efficiency of various computing tasks. It involves breaking down a large task (testing the whole virtual environment) into smaller subtasks (testing each node) and

executing them concurrently, thereby reducing the overall time required for completion. In the case of end-to-end testing of the application for managing the configuration of enterprise virtual infrastructure, the use of parallelism is particularly beneficial as it allows for the testing of multiple virtual machines simultaneously, resulting in significant time savings.

It is worth noting that with the new approach to end-to-end testing, the entire process is performed and controlled by a single engineer, as opposed to the old approach, which required the involvement of three different engineers with varying levels of knowledge and expertise. This allows for a more streamlined and efficient testing process, as the same engineer is responsible for each stage of the testing process and can quickly identify and resolve issues as they arise.

During the testing process, three engineers are, of course, more efficient than one, since they can confer with each other and find the root cause of the problem faster. But in this case, these are engineers from different teams who perform rather isolated parts of a single process. Therefore, if one of the engineers had a problem, the others could not help him much, because they did not know the subject area of the other teams well. Therefore, the new framework gives the best result in such a situation.

Additionally, the developed software framework for end-to-end testing generates a detailed report in natural language, which provides a comprehensive overview of the testing process and the results obtained. This report is automatically generated by the software, eliminating the need for manual report writing, which was necessary with the old approach. This saves considerable time and effort, allowing engineers to focus on other critical aspects of the development process.

Furthermore, the use of natural language in the generated report makes it easy for stakeholders to understand the testing results and make informed decisions regarding the software development. This can be particularly useful for managers and other non-technical team members who may not have the technical expertise required to interpret traditional testing reports.

6. Conclusion

The proposed approach to end-to-end testing for the application for managing the configuration of enterprise virtual infrastructure offers several advantages over the traditional approach, including a more streamlined testing process and the automatic generation of a comprehensive report in natural language. These benefits can help engineers save time and effort while improving the quality of the software and ensuring that it meets the required standards and specifications.

References

- [1] Tsai W., Bai X. et al. End-to-end integration testing design, In Proc. of the 25th Annual International Computer Software and Applications Conference (COMPSAC), 2001, pp. 166-171.
- [2] Soni M. End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. In Proc. of the IEEE International Conference on Cloud Computing in Emerging Markets (CEEM), 2015, pp. 85-89.
- [3] Wang J., Shao Y. et al. A Survey of Vehicle to Everything (V2X) Testing. *Sensors*, vol. 19, issue 2, 2019, article no. 334, 20 p..
- [4] Beck K. *Test Driven Development By Example*. Addison-Wesley Professional, 2002, 240 p.
- [5] Brady E. *Type-Driven Development with Idris*. Manning Publications, 2017, 480 p.
- [6] Smart J.F. *BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle*. Manning Publications, 2014, 384 p..
- [7] Wesenberg H., Olmheim J. Agile Enterprise Software Development Using Domain-Driven Design. In Proc. of the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications, 2007, pp. 983–993.
- [8] Chowdhury A.F., Huda M.N. Comparison between Adaptive Software Development and Feature Driven Development. In Proc. of the International Conference on Computer Science and Network Technology, 2011, pp. 363-367.

- [9] Martinez Hernandez C., Martinez A. et al. Comparison of End-to-End Testing Tools for Microservices: A Case Study. *Advances in Intelligent Systems and Computing*, vol. 1330, 2021, pp. 407-416.
- [10] Satoh I. A Testing Framework for Mobile Computing Software. *IEEE Transactions on Software Engineering*, vol. 29, issue 12, 2003, pp. 1112-1121.
- [11] Mohanan R. What Is BDD (Behavior Driven Development)? Meaning, Process, and Examples. Available at: <https://www.spiceworks.com/tech/devops/articles/what-is-bdd/>, accessed: May 10, 2023.
- [12] Sobek D.K. II. System-Level Design: *International Journal of Engineering Education*, vol. 22, issue 3, , 2006, pp. 533-539.
- [13] Allure Jenkins Plugin. Available at: <https://plugins.jenkins.io/allure-jenkins-plugin/>, accessed Mar. 28, 2023.

Information about authors / Информация об авторах

Евгений Вадимович ГЕРАСЬКИН – студент 2 курса магистратуры. Научные интересы: тестирование ПО, backend-разработка, разработка информационных систем, автоматизация тестирования.

Evgeny Vadimovich GERASKIN is a 2nd year student of the Master's program. Research interests: software testing, backend development, information systems development, test automation.

Никита Владимирович ВОИНОВ – кандидат технических наук, доцент. Научные интересы: модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования.

Nikita Vladimirovich VOINOV – Candidate of Technical Sciences, Associate Professor. Scientific interests: models, methods and algorithms for designing and analyzing programs and software systems, their equivalent transformations, verification and testing.

DOI: 10.15514/ISPRAS-2023-35(2)-11



Parallel computations in problems of reconstruction of distorted images in spatial-spectral form

B.V. Kostrov, ORCID: 0000-0002-2984-4480 <kostrov.b.v@evm.rsreu.ru>
N.N. Grinchenko, ORCID: 0000-0003-2318-276X <grinchenko_nn@mail.ru>
A.A. Vyugina, ORCID: 0000-0002-0087-5988 <vyugina.an.a@gmail.com>
S.N. Baranova, ORCID: 0009-0005-6557-5192 <baranova.sv.n@gmail.com>

*Ryazan State Radio Engineering University named after V.F. Utkin,
59/1, Gagarina str, Ryazan, 390005, Russia*

Abstract. The object of the study is the image processing process in preparation for data transmission, as well as subsequent recovery. The subject of the study is the application of parallel computing in image processing tasks. The purpose of the article is to study the method of image reconstruction with correction of distortion of the transmission vector based on the input of a value in the trend of its neighbors. The relevance of this topic is determined by the need to effectively perform the operations preceding and following the transmission over the communication channel. During the experimental part, the results were obtained in the form of execution time values with sequential execution and using parallel calculations, which gave the expected increase in speed-up. Since the image consists of unrelated parts, it can be successfully processed by applying data parallelization.

Keywords: image processing; bit plane; parallel computing; complex error; recovery; data transfer; MATLAB

For citation: Kostrov B.V., Grinchenko N.N., Vyugina A.A., Baranova S.N. Parallel computations in problems of reconstruction of distorted images in spatial-spectral form. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 157-168. DOI: 10.15514/ISPRAS-2023-35(2)-11

Параллельные вычисления в задачах восстановления искаженных изображений в пространственно-спектральной форме

Б.В. Костров, ORCID: 0000-0002-2984-4480 <kostrov.b.v@evm.rsreu.ru>
Н.Н. Григченко, ORCID: 0000-0003-2318-276X <grinchenko_nn@mail.ru>
А.А. Вьюгина, ORCID: 0000-0002-0087-5988 <vyugina.an.a@gmail.com>
С.Н. Баранова, ORCID: 0009-0005-6557-5192 <baranova.sv.n@gmail.com>

*Рязанский государственный радиотехнический университет им. В.Ф. Уткина,
Россия, 390005, г.Рязань, ул.Гагарина, 59/1*

Аннотация. Объектом исследования является процесс обработки изображения при подготовке к передаче данных, а также последующее восстановление. Предметом исследования является применение параллельных вычислений в задачах обработки изображений. Целью статьи является исследование метода восстановления изображения с коррекцией искажения вектора передачи на основе ввода значения тренда его соседей. Актуальность данной темы определяется необходимостью эффективного выполнения операций, предшествующих и следующих за передачей по каналу связи. В ходе экспериментальной части были получены результаты в виде значений времени выполнения при последовательном выполнении и использовании параллельных вычислений, что дало ожидаемый

прирост ускорения. Поскольку изображение состоит из несвязанных частей, его можно успешно обрабатывать, применяя распараллеливание данных.

Ключевые слова: обработка изображения; битовая плоскость; параллельные вычисления; сложная ошибка; восстановление; передача данными; MATLAB

Для цитирования: Костров Б.В., Григченко Н.Н., Вьюгина А.А., Баранова С.Н. Параллельные вычисления в задачах восстановления искаженных изображений в пространственно-спектральной форме. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 157-1681. DOI: 10.15514/ISPRAS-2023-35(2)-11

1. Introduction

In the modern information world, all information is collected, stored, and transmitted via communication channels. Every year the amount of information generated by various media increases, that leads to the problem of improve the efficiency of data transmission and processing. Various kinds of interference may occur during data transmission [1]. Therefore, the issue related to the process of forming a data vector, its transmission and subsequent recovery is an integral part in solving the problem of data processing, including graphical information.

After receiving the data by the receiving party, they must be restored to their original form, however, not the entire sequence can be accepted correctly, which leads to the need to calculate and correct distorted values. The calculations performed after receiving graphic data, especially if it is necessary to restore the parts damaged by interference, can take a large amount of time resources, which implies the task not only to correct the distorted pixels, but to perform these operations with sufficient speed. Parallelization can be carried out on different devices: multicore CPUs or video cards with a large number of stream processors.

When performing preprocessing of images for the formation of the transmission vector, as well as recovery after receiving data, methods based on the processing of independent parts of the vector are used. The distribution and independence of data makes it possible to effectively divide the execution into several nodes working in parallel.

The aim of the paper is to study the effectiveness of parallel computing in the tasks of preprocessing and postprocessing images for transmission over a noisy communication channel.

2. Forming transmission vector

The image can be represented in a spatial form, the elements of which are brightness values, for each of which a byte of memory is allocated (1).

$$\mathbf{G}_{N \times N} = \begin{pmatrix} g_{1,1} & \cdots & g_{1,N} \\ \vdots & \ddots & \vdots \\ g_{N,1} & \cdots & g_{N,N} \end{pmatrix} \quad (1)$$

where \mathbf{G} is a matrix in spatial representation, $g_{i,j}$ is the brightness value of one pixel, $N \times N$ is the image size.

Data transmission via communication channels takes place at the physical level, while it is more convenient to represent discrete values of image brightness in a spatial-spectral form, which is formed in accordance with a certain function [2, 3]. In the conducted study, Walsh functions [4] were used as non-trigonometric orthogonal basis functions, which were superimposed on separate blocks of values with a size of 32×32 (2) to decrease the value of the spectral component that reduces the amount of memory needed to store the image in a spatially spectral form.

$$\mathbf{G} = \begin{pmatrix} G_{1,1} & \cdots & G_{1,\frac{N}{n}} \\ \vdots & \ddots & \vdots \\ G_{\frac{N}{n},1} & \cdots & G_{\frac{N}{n},\frac{N}{n}} \end{pmatrix} \quad (2)$$

where \mathbf{G} is a matrix in spatial representation, $G_{i,j}$ is a 32×32 submatrix, N is the number of pixels, n is the number of 32×32 blocks.

Splitting the image into parts allows independent pixel processing. Each of the individual blocks is transformed to a spatially spectral form by one method, which means that the computational process can be parallelized for each submatrix of pixels.

As a result of the transformation, a matrix is obtained representing the image in the spatial-spectral form (3).

$$\mathbf{F}(G)_{N \times N} = \mathbf{G} \times \mathbf{H} = \begin{pmatrix} f_{1,1} & \cdots & f_{1,N} \\ \vdots & \ddots & \vdots \\ f_{N,1} & \cdots & f_{N,N} \end{pmatrix} \quad (3)$$

where \mathbf{F} is the matrix of values of the function of \mathbf{G} , \mathbf{G} is the matrix of image elements, \mathbf{H} is the matrix of basic functions, $f_{i,j}$ is the value of the spectrum [5], $N \times N$ is the size of the spectrum matrix.

To form a data transmission vector, it is necessary to represent the spatial-spectral form of the matrix in binary representation: one value will be stored in an 8-bit representation (4) [6].

$$\mathbf{F}_{(N \times N)_{10}} = \mathbf{F}_{((N * 8) \times N)_2} \quad (4)$$

where \mathbf{F} is the matrix of values of a function of \mathbf{G} , $N \times N$ is the size of the spectrum matrix, $N * 8$ is the number of values in the bit representation.

The number of bit planes (k) takes values from 1 to 8, since one byte of memory is allocated for the spectrum. The formation of bit plane matrices is performed in accordance with expression (5) [7].

$$\mathbf{B}_{N \times N}^k = \begin{pmatrix} f_{1,k} & f_{1,k+8} & \cdots & f_{1,k+8*N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N,k} & f_{N,k+8} & \cdots & f_{N,k+8*N} \end{pmatrix} \quad (5)$$

where \mathbf{B} is the matrix of the bit plane, k is the bit number of the byte whose plane is formed, $k = (1,8)$, $f_{i,j}$ is the value of the corresponding bit (0, 1).

When divided into bit planes, the original spatial-spectral matrix can be represented as sums of digits of the number (6).

$$\mathbf{F}_{N \times N} = 2^7 \begin{pmatrix} f_{1,1}^7 & \cdots & f_{1,8*N-7}^7 \\ \vdots & \ddots & \vdots \\ f_{N,1}^7 & \cdots & f_{N,8*N-7}^7 \end{pmatrix} + 2^6 \begin{pmatrix} f_{1,2}^6 & \cdots & f_{1,8*N-6}^6 \\ \vdots & \ddots & \vdots \\ f_{N,2}^6 & \cdots & f_{N,8*N-6}^6 \end{pmatrix} + \cdots + 2^0 \begin{pmatrix} f_{1,8}^0 & \cdots & f_{1,8*N}^0 \\ \vdots & \ddots & \vdots \\ f_{N,8}^0 & \cdots & f_{N,8*N}^0 \end{pmatrix} \quad (6)$$

where \mathbf{F} is a matrix in the space-spectral representation, $f_{i,j}^k$ is the k digit of the binary representation of the spectrum value.

The image transmission vector is formed by digits, that is, from the sign plane, and then from the highest bit to the lowest bit (7) [8].

$$V_{1 \times N * 8} = (f_{1,1}^1, \dots, f_{N,N}^1, f_{1,1}^2, \dots, f_{N,N}^2, \dots, f_{1,1}^8, \dots, f_{N,N}^8) \quad (7)$$

where V is the transmission vector, N is the number of elements in the bit plane, $f_{i,j}^k$ is the value of the corresponding bit (0, 1), the superscript from 1 to 8 is the number of the bit plane.

The presented sequence of transformation of the source image to the transmission vector contains certain steps: conversion to a spectrum, splitting into bit planes, which can be distributed into separate processing units of 32x32 pixels, since there is no dependence between its component parts in the graphic data. An identical situation occurs on the receiving side: correction of distorted bits, conversion to brightness values.

The scheme of image transmission over the communication channel [9], including the process of its restoration, is shown in fig. 1.

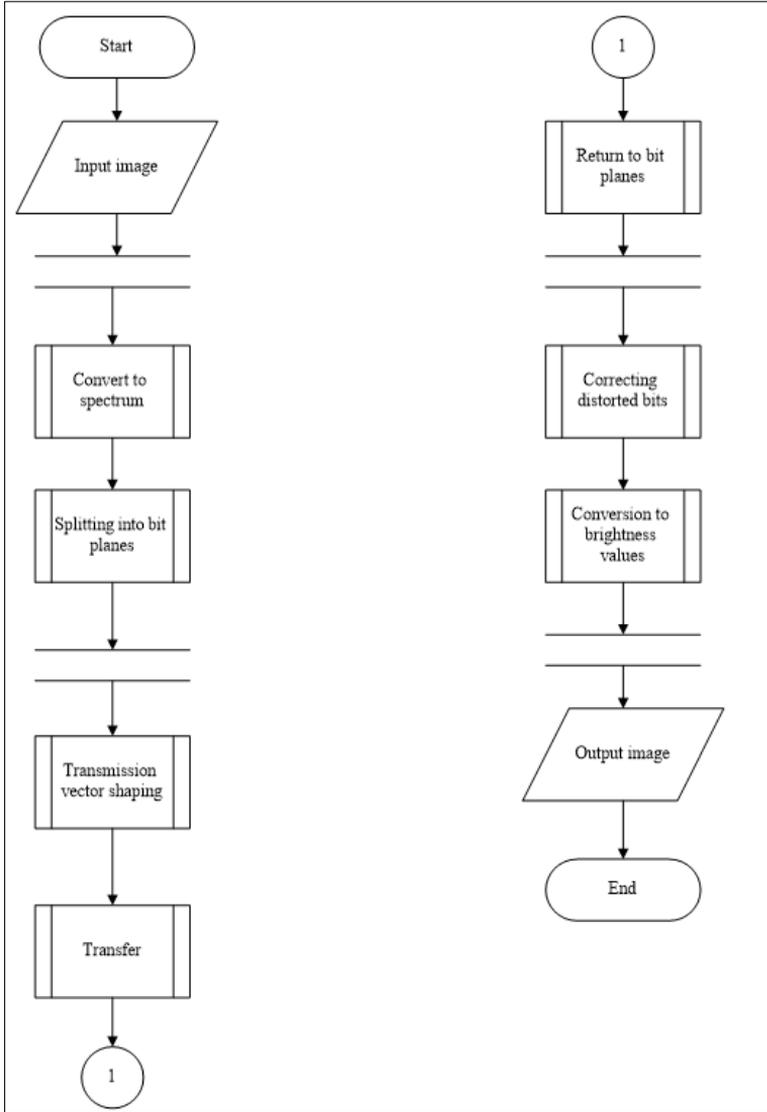


Fig. 1. Image transmission scheme

3. Parallel computing in the MATLAB environment

Modeling of transformations of the source and received data, as well as the noisiest communication channel, was carried out in the Matlab environment. For this purpose, functions corresponding to the stages of the data transfer scheme have been developed (fig. 1).

Matlab supports the use of parallel computing technologies on both CPU and GPU.

Implementation of parallel computing using a multicore processor uses the parallel loop operator `parfor`, which automates the creation of parallel pools and manages file dependencies [10].

Parallel computing on a GPU can be started using the “`gpuArray`” class, which supports automatic launch of standard functions using GPU [10].

The study was conducted on the principle of data separation, that is, each core processes a separate part of the image with the same function.

When running parallel computations, the main thread initializes workflows on nodes that correspond to the execution of a specific thread. When processing an image, parallelization is performed at the data level by dividing the original matrix of elements into separate blocks. When switching to bit planes, separate matrices appear containing bits of a certain digit (6), which are also independent of other matrices, which allows you to work on them in parallel. Fig. 2 shows a scheme of parallel calculations when translating an image to bit planes.

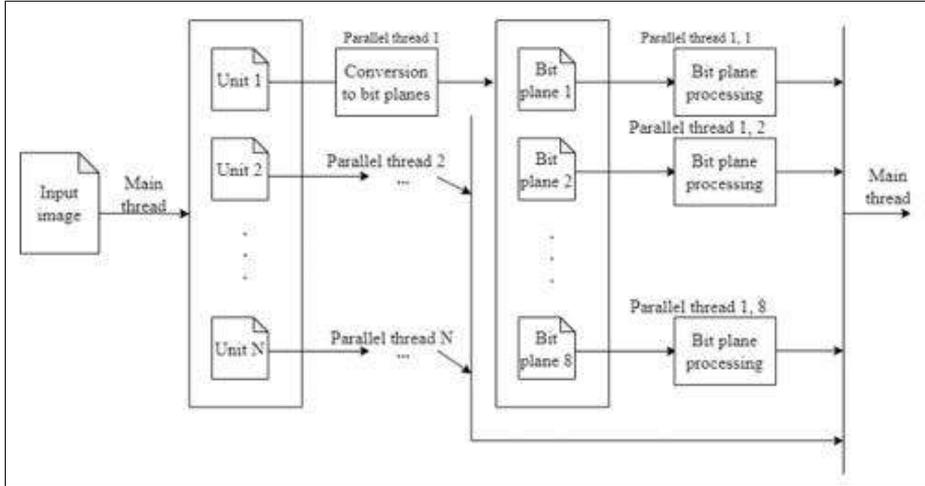


Fig. 2. Diagram of the processing process using parallel computing

The basic characteristic of determining the effectiveness of parallel computing is acceleration (8).

$$S_p = \frac{T_1}{T_p} \quad (8)$$

where S_p is acceleration, T_1 is sequential execution time, T_p is parallel execution time

4. Distortion during data transmission

When transmitting data through a communication channel, errors may occur due to various reasons: signal attenuation, noise, interference. Bit changes are considered distortions in the transmitted vector. There is a dependence of the effect of errors on the bit depth of the image: when an error hits the highest digit, the distortion introduced by the interference will affect the quality of the restored image more strongly than when the same error hits the lowest digit. This dependence is related to the weight of the digit in the binary representation of the number. All types of errors can be divided into 2 groups:

- 1) An error in one bit, which is a change to the opposite value of a random bit in the transmission vector.
- 2) A complex error, which is a change in a continuous sequence of bits. The extreme bits in the sequence with a packet error have a distorted as well as correct values may be contained inside such a sequence [11].

We study complex errors that differ in the nature of the distortion of the sequence of bits:

- 1) The sequence of bits takes the opposite value (error "NOT") (9).

$$V_{1 \times N} = \{V_{(1,(1..i-1))}, \overline{V_{(1,(i..j))}}, V_{(1,(j+1..N))}\} \quad (9)$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error.

- 2) The sequence of bits takes the value "0" (knocked out bits (KOB)) (10).

$$V_{1 \times N} = \{V_{(1,(1..i-1))}, 0_{(1,(i..j))}, V_{(1,(j+1..N))}\} \quad (10)$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error.

3) The sequence of bits takes values alternately "0" and "1" (batch error (BE)) (11).

$$V_{1 \times N} = \{V_{(1,(1..i-1))}, E_{(1,(i..j))}, V_{(1,(j+1..N))}\} \tag{11}$$

where V is the transmission vector, N is the number of bits in the vector, i is the initial bit of the sequence with an error, j is the final bit of the sequence with an error, $E(1 \times M) = \{0..01..10..01..1\}$, where M is the sequence of the error vector.

5. The impact of errors on image recovery

During the experiment, various types of errors were superimposed on the original transmission vector: "NOT" error, knocked out bits and batch error. For the experiment, a set of images was collected, which can be conditionally divided into 3 categories: "forest", "land and water surface" and "urban area" (fig. 3). The images have the same size 256×256 , the brightness of the pixels varies from 10 to 175.

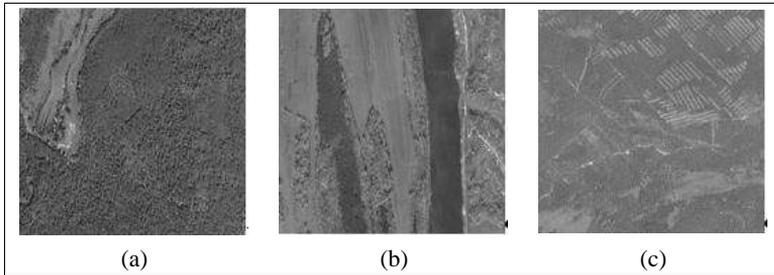


Fig. 3. Test Images: a) category "forest", b) category "land and water surface", c) category "urban area"

In the Matlab environment, the bit representation of a number has an inverse numbering of digits: from the sign digit (7th bit) – the 1st plane to the lowest digit (0th bit) – the 8th plane [12,13].

The formed initial bit planes for the "forest" image are shown in fig. 4.

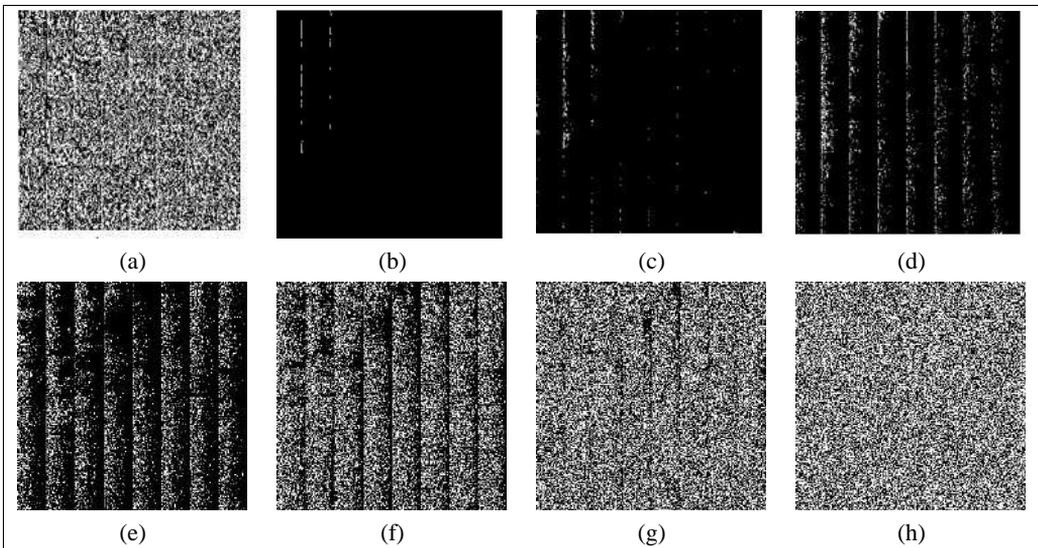


Fig. 4. Bit planes of the "forest" image: a) 1st plane, b) 2nd plane, c) 3rd plane, d) 4th plane, e) 5th plane, f) 6th plane, g) 7th plane, h) 8th plane

To study the effect of errors, depending on the hit in different parts of the vector, 256-bit ranges corresponding to individual bit planes were distorted, that is, during the experiment, the error was superimposed separately on sections from the sign to the lowest digit [14]. Fig. 5 shows 3 types of errors on the 2nd bit plane. The “knocked out bits” error has a minimal effect on the distortion of the plane, since the decimal value of the spatial-spectral representation for the image brightness matrix in most cases does not exceed 64, which in the bit representation gives the value 0 in the plane under consideration.

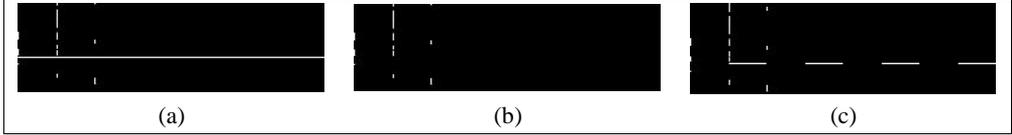


Fig. 5. Types of errors in image 1: a) error “NOT”, b) knocked out bits, c) batch error

When errors appeared in one of the planes, their influence on the resulting image was analyzed, in which the existing distortions were not corrected: Table 1 shows the values of the mean square deviation (MSD) for 3 categories of images and 3 types of error. The value of the MSD was calculated only for the part of the image in which the error was located [15].

Table 1. MSD for an image with an error (without correction)

Image	Type of error	1st bit.pl	2nd bit.pl	3rd bit.pl	4th bit.pl	5th bit.pl	6th bit.pl	7th bit.pl	8th bit.pl
Fig. 2 a)	NOT	4,08	10,6	5,92	3,52	1,99	1,13	0,73	0,59
	KOB	2,25	0,53	1,22	0,94	0,81	0,77	0,62	0,55
	BE	3,02	7,68	4,34	2,53	1,46	0,87	0,63	0,57
Fig. 2 б)	NOT	5,57	12	6,51	3,81	2,27	1,46	1,18	1,09
	KOB	4,2	1,44	1,66	1,52	1,25	1,17	1,1	1,07
	BE	3,73	8,19	4,52	2,68	1,8	1,29	1,12	1,08
Fig. 2 в)	NOT	6,87	11,8	7,16	3,98	2,05	1,12	0,72	0,59
	KOB	4,8	0,53	1,91	1,55	1,25	0,81	0,62	0,55
	BE	4,63	8,66	5,24	2,86	1,53	0,89	0,64	0,56

“NOT” is the “NOT” error, KOB is the knocked-out bits, BE is a batch error

Based on the MSD values presented in table 1, it can be concluded that the knocked-out bits do not greatly affect the restored image, since the original values of the spectrum in binary representation have the value 0 in the higher digits. The error, which is a reset of values to 0, gives a small decrease in the informativeness of the data when it falls into the described range of the vector.

Any type of error that falls on the lower bits (0-2 bit bits) leaves a small distortion that can be considered acceptable. However, at the same time, the result of an error hitting the higher digits has a fairly large impact on the information content of the output image. An example of a “forest” image with a batch error is shown in fig. 6.

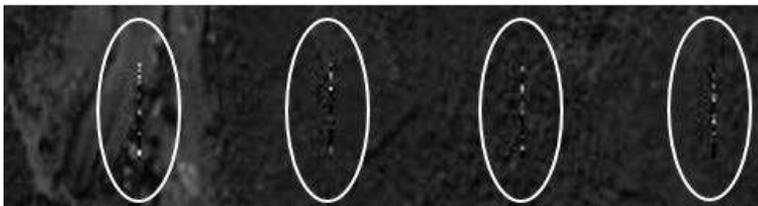


Fig. 6. The “forest” image with a batch error in the 2nd plane

6. Description of the error correction process

After conducting experiments on the imposition of various types of errors, the task was set to correct the received vector in order to reduce the amount of distortion of the transmission vector to improve

the quality of the information content of the restored image. To do this, an algorithm has been developed, which consists of the following steps:

- 1) Transformation of a vector to a matrix form in accordance with bit planes [16].
- 2) Calculation of the average value in decimal representation by adjacent rows of values (above and below) relative to the location of the error bit.
- 3) Calculation of the decimal value of the spectrum in which the error occurred.
- 4) Analysis of the obtained values for whether the value with an error falls into the trend of the values of its neighbors.
- 5) Correction of a bit in the plane to the opposite, if the value of the spectrum with an error does not fit into the trend set by its neighbors.

As a calculation is entering into the trend, the ratio is compiled (12). Fig. 7 shows a graphical representation of the position of the pixels of the image from which the calculation is performed.

$$f_{i,j} / \frac{f_{i-1,j} + f_{i+1,j}}{2} \tag{12}$$

where f is the value of the spectrum in decimal representation, (i, j) are the indices of finding an element in the spectrum matrix.

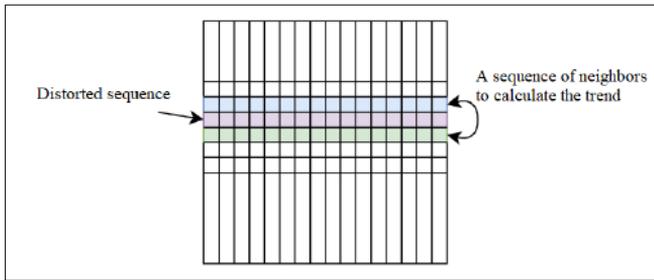


Fig. 7. Graphical representation of the expression (12)

As a result of the conducted research, a number of questions were raised for spatial-spectral matrices: which ranges of values for the ratio are acceptable in order to make a decision about the entry of the desired value into the trend, and what should this ratio strive for. The more accurate the hit in the trend, the closer the value of the spectrum is to the average value of its neighbors, that is, to the value of 1. However, during the experiment, the following problematic situations were identified:

- 1) Table 2 shows the corresponding initial values of the spectrum and their binary unsigned representation.

Table 2. Example 1

Position	Spectrum values	Binary representation of the spectrum
(i-1, j)	710	00001112
(i, j)	2110	00101012
(i+1, j)	1910	00100112

The ratio value is $\frac{21}{13} \approx 1,6$, which means that the real values of the spectrum of the original image can give a sufficiently large deviation from the average value of its neighbors. Accordingly, it is impossible to compare unambiguously with the average value, so it is necessary to check whether the value with the changed bit would be better to enter the trend: $10101012=8510$, the ratio $\frac{85}{13} \approx 6,5$, which is significantly higher than the original value.

Thus, falling into a trend cannot be judged by a constant value, but should be compared by 2 ratios: the initial value to the average value of its neighbors, the changed value to the average value of its neighbors.

- 2) Since the values of the spectrum can be both positive and negative, neighbors can have the same value of the opposite sign (for example, 2 and -2), then their average value will be 0.
 After analyzing the matrices of the spatial-spectral representation, 2 conditions were revealed to determine the trend: a comparison of the relations for the original accepted sequence and the modified one; selecting the value that is closer to 1.

7. Results of the experimental part

In the experiment, the MSD ratio between the input and output images, that is, the brightness values, was calculated. However, during the conversion of the original brightness matrix into a spatial-spectral representation and the restoration from the spectral form to the brightness values, rounding operations are performed, which impose additional distortions. Table 3 shows the MSD values for the spectrum value matrices: initial and after error correction.

Table 3. The value of MSD for the spectrum

Image	Type of error	1st bit.pl	2nd bit.pl	3rd bit.pl	4th bit.pl	5th bit.pl	6th bit.pl	7th bit.pl	8th bit.pl
Fig. 2 a)	NOT	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
	KOB	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
	BE	0,2709	0	0,6963	0,4606	0,3371	0,2787	0,1584	0,08
Fig. 2 б)	NOT	0,7554	0,9545	1,074	0,8539	0,7398	0,699	0,6746	0,6578
	KOB	0,7554	1,3714	1,1815	0,8887	0,75	0,7017	0,6753	0,658
	BE	0,7554	0,9545	1,074	0,8539	0,7398	0,699	0,6746	0,6579
Fig. 2 в)	NOT	0,305	0,6963	0,9211	0,8349	0,5708	0,3168	0,1846	0,0877
	KOB	0,305	0	0,8528	0,8165	0,5641	0,3138	0,1846	0,0877
	BE	0,305	0,6963	0,9211	0,8349	0,5708	0,3168	0,1834	0,087
"NOT" is the "NOT" error, KOB is the knocked-out bits, BE is a batch error									

The results obtained show that the analysis of the studied value for falling into the trend of the values of its neighbors gives a good result in correcting errors. When correcting errors on the lower digits of the numbers corresponding to the seventh and eighth bit planes, the MSD systems are almost identical with the results without restoration, which once again confirms the statement about the small effect of distortion on the original image.

To restore the distorted spectral components, only 3 values are required, which makes it possible to efficiently divide the data into nodes for parallel calculation of the algorithm described in the article. Table 4 contains the time values for post-processing images on a personal computer with an intel Core i5 CPU and an Nvidia GeForce RTX 3060 GPU.

Table 4. Run-time values

Size of the original data set	Sequential calculation time (sec)	Parallel computing time on CPU (sec)	Parallel computing time on GPU (sec)
220 KB	2.79	1.62	0.26
560 KB	5.23	2.72	0.49
1280 KB	14.52	7.68	1.36

8. Conclusion

Restoration by analyzing the input of the spectrum values into the trend of neighboring values in most cases leads the resulting vector to a form whose deviation from the original is quite small. Depending on the characteristics of the image: the range of brightness, sharp boundaries, values of the spatial-spectral representation may arise, complicating the decision to enter the trend, which was taken into account when implementing the approach described in the article. The use of parallel

computing has made it possible to increase the efficiency of processing received data, which is especially important in real-time systems.

As a result of the experiment, an image transmission scheme with a noisy communication channel was constructed, where the processes of preprocessing and restoring the original image were performed on the basis of parallel calculations. The main task was to find a solution to improve the performance of program components by distributing data across individual cores. Testing applications of parallel computing on the CPU and GPU on different sets of images shows a sufficiently large acceleration for systems working with landscape images.

References / Список литературы

- [1] Zolotarev V.V., Ovechkin G. Z-80 Noiseproof Coding. Methods and Algorithms: Handbook. Moscow, Hot Line-Telecom, 2004, 126 pp. (in Russian) / Золотарёв В.В., Овечкин Г. В.З-80 Помехоустойчивое кодирование. Методы и алгоритмы: Справочник. М.: Горячая линия–Телеком, 2004 г., 126 стр.
- [2] Svetlov G.V., Sumenkov N.A. et al. Application of the theory of discrete signals defined on finite intervals for aerospace image processing. *Vestnik VKO «Almaz-Antey»*, issue 3, 2017, pp. 94-102 (in Russian) / Светлов Г.В., Суменков Н.А. и др. Применение теории дискретных сигналов, определенных на конечных интервалах, для обработки аэрокосмических изображений. Вестник Концерна ВКО «Алмаз – Антей», вып. 3, 2017 г., стр. 94-102.
- [3] Bastrychkin A.S., Kostrov B.V. et al. Calculation of image transmission partial spectrum. In Proc. of the 9th Mediterranean Conference on Embedded Computing (MECO), 2020, 3 p.
- [4] Zlobin V.K., Kostrov B.V., Svirina A.G., Spectral analysis of images in finite bases. М., KURS Publishing House, 2016, 172 p. (in Russian) / Злобин В.К., Костров Б.В., Свирина А.Г. Спектральный анализ изображений в конечных базисах. М.: КУРС, 2016 г., 172 стр.
- [5] Ahmed N., Rao K. R. Orthogonal transforms in digital signal processing. Springer, 1975, 276 p.
- [6] Kostrov B.V., Bastrychkin A.S. et al. Protocol of image transmission over communication channel. In Proc. of the All-Russian Scientific and Technical Conference on Intelligent and Information Systems, 2016, pp. 220-225 (in Russian) / Костров Б. В., Бастрычкин А. С. и др. Протокол передачи изображений по каналу связи. Материалы Всероссийской научно-технической конференции «Интеллектуальные и информационные системы», 2016 г., стр. 220–225.
- [7] Grinchenko N.N., Baranova S.N. et al. Mathematical model of the image transmission process based on bit planes. *Vestnik VKO «Almaz-Antey»*, issue 1, 2023, pp. 82–89 (in Russian) / Гринченко Н.Н., Баранова С.Н. Математическая модель процесса передачи изображений на основе битовых плоскостей. Вестник Концерна ВКО «Алмаз – Антей», вып. 1, 2023, pp. 82-89.
- [8] Bastrychkin A.S., Kostrov B.V. et al. Traffic shaping in data transmission systems. In Proc. of the V International Scientific and Technical Forum on Modern Technology in Science and Education (STNO-2022), vol. 4, 2022, pp. 115-120 (in Russian) / Бастрычкин А.С., Костров Б.В. и др. Формирование трафика в системах передачи данных. Сборник трудов V Международного научно-технического форума «Современные технологии в науке и образовании» (СТНО-2022), том 4, 2022 г., стр. 115-120.
- [9] Kostrov B.V., Solomentseva N.I. Communication channel modeling. *Izvestiya Tula State University. Technical Sciences*, issue. 2, 2017, pp. 95-100 (in Russian) / Костров Б.В., Соломенцева Н.И. Известия Тульского государственного университета. Технические науки, 2017 г., 2017 г., стр. 95-100.
- [10] Olenev N.N., Pechenkin R.V., Chernetsov A.M. Parallel programming in MATLAB and its applications Moscow, Dorodnicyn Computing Centre of RAS, 2007, 120 p. (in Russian) / Оленёв Н.Н., Печёнкин Р.В., Чернецов А.М. Параллельное программирование в MATLAB и его приложения. Вычислительный центр им. А.А. Дородницына РАН, 2007 г., 120 стр.
- [11] Kostrov B.V. Information Theory: Textbook, Ryazan State Radio Engineering University, 2006. 124 p. (in Russian) / Костров Б.В. Теория информации. Учебное пособие. Рязань, РГРТУ, 2006 г., 124 стр.
- [12] Gilat A. MATLAB: An Introduction with Applications, 6th edition. John Wiley & Sons, 2017, 414 p.
- [13] Potemkin V.G. Calculations in the MATLAB environment. Moscow: Dialog-MIFI, 2004. 720 p. (in Russian) / Потемкин В. Г. Вычисления в среде MATLAB. М., Диалог-МИФИ, 2004 г., 720 стр.
- [14] Grinchenko N.N., Vyugina A.A., Baranova S.N. et al. The study of bit traffic in the transfer of landscape images. In Proc. of the V International Scientific and Technical Forum on Modern Technology in Science and Education (STNO-2022), vol. 4, 2022, pp. 111-115 (in Russian) / Гринченко Н.Н., Вьюгина А.А., Баранова С.Н. Исследование битового трафика при передаче ландшафтных изображений. Сборник

трудов V Международного научно-технического форума «Современные технологии в науке и образовании» (СТНО-2022), том 4, 2022 г., стр. 111-115.

[15] Shvedov A.S. *Probability Theory and Mathematical Statistics. Introductory Level*. M., HSE Publishing House, 2005. 254 p. (in Russian) / Шведов А.С. *Теория вероятностей и математическая статистика. Вводный уровень*. М., Издательский дом ВШЭ, 2005 г., 254 стр.

[16] Gonzalez R., Woods R. *Digital Image Processing*. Edition 4th. Pearson, 2018, 1022 p.

Information about authors / Информация об авторах

Борис Васильевич КОСТРОВ – доктор технических наук, профессор. Научные интересы: обработка изображений, помехоустойчивое кодирование.

Boris Vasilyevich KOSTROV – Doctor of Technical Sciences, Professor. Research interests: image processing, error-correcting coding.

Наталья Николаевна ГРИНЧЕНКО – кандидат технических наук, доцент. Научные интересы: анализ данных, обработка изображений.

Natalya Nikolaevna GRINCHENKO – Candidate of Technical Sciences, Associate Professor. Research interests: data analysis, image processing.

Ангелина Алексеевна ВЬЮГИНА – аспирантка. Научные интересы: программирование, обработка изображений.

Angelina Alekseevna VYUGINA – post-graduate student. Research interests: programming, image processing.

Светлана Николаевна БАРАНОВА – аспирантка. Научные интересы: анализ данных, обработка изображений.

Svetlana Nikolaevna BARANOVA – post-graduate student. Research interests: data analysis, image processing.

DOI: 10.15514/ISPRAS-2023-35(2)-12



Объектно-ориентированный подход к поддержке сценариев в системах оптического моделирования

М.С. Копылов, ORCID: 0000-0002-9526-0766 <dvaag@hotmail.com>

Н.Б. Дерябин, ORCID: 0000-0003-1248-6047 <dek@gin.keldysh.ru>

Е.Ю. Денисов, ORCID: 0000-0002-0614-9100 <eed@spp.keldysh.ru>

*Институт прикладной математики им. М.В. Келдыша РАН,
125047, Россия, г. Москва, Миусская пл., д. 4.*

Аннотация. В статье рассматриваются особенности поддержки сценариев на языке Python в активно развивающейся системе интерактивной графики. Подобная поддержка является трудоёмкой задачей, которую сложно автоматизировать в общем случае. В качестве решения этой проблемы предлагается подход, позволяющий разработчикам совмещать создание новых компонентов системы с одновременным встраиванием поддержки сценариев без написания избыточного добавочного кода. Результатом работы является дружелюбный пользователю объектно-ориентированный API, описывающий все аспекты взаимодействия системы и сценариев. Используя этот API сценарии могут применяться для автоматизации моделирования, а также для расширения возможностей системы с помощью специальных скриптовых классов. Последнее особо важно, так как оставляет обычным пользователям возможность самостоятельно расширять системы с закрытым исходным кодом.

Ключевые слова: автоматизация моделирования; расширяемость; язык сценариев; графический интерфейс

Для цитирования: Копылов М.С., Дерябин Н.Б., Денисов Е.Ю. Объектно-ориентированный подход к поддержке сценариев в системах оптического моделирования. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 169-180. DOI: 10.15514/ISPRAS-2023-35(2)-12

An object-oriented approach to scenario support in optics CAD systems

M.S. Kopylov, ORCID: 0000-0002-9526-0766 <dvaag@hotmail.com>

N.B. Deryabin, ORCID: 0000-0003-1248-6047 <dek@gin.keldysh.ru>

E.Yu. Denisov, ORCID: 0000-0002-0614-9100 <eed@spp.keldysh.ru>

*Keldysh Institute of the Applied Mathematics of RAS,
4, Miusskaya Sq. Moscow, 125047, Russia.*

Abstract. This article discusses the problems of supporting Python scripts in an actively developing interactive graphics system. Such support is a time-consuming task, which is difficult to automate in the general case. As a solution to this problem, we propose an approach that allows developers to combine the creation of new system components with the simultaneous embedding of scripting support without writing redundant additional code. The result is a user-friendly object-oriented API that describes all aspects of interaction between the system and scripts. Scripts using this API can be used to modeling automation as well as to extend the system with custom extension classes. The latter is especially important as it leaves the ability for ordinary users to extend closed-source systems on their own.

Keywords: modeling automation; extensibility; script language; graphical interface

For citation: Kopylov M.S., Deryabin N.B., Denisov E.Yu. An object-oriented approach to scenario support in optics CAD systems. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 169-180 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-12

1. Введение

Большинство современных систем оптического моделирования являются весьма сложными программными продуктами, состоящими из множества компонентов, модулей и приложений, использующихся для решения различных практических задач. Чтобы максимально использовать возможности таких систем, пользователям часто приходится перемещаться по слоям меню и диалоговых окон, например, для настройки различных параметров моделирования, счёт которых может идти на десятки, а то и сотни. В итоге наступает момент, когда базовых возможностей, предоставляемых графическим интерфейсом пользователя, становится недостаточно для эффективной работы с системой. Наиболее популярным способом преодоления подобной ограниченности является использование сценариев. С их помощью можно добиться следующих преимуществ.

- Сценарии позволяют автоматизировать процессы моделирования, такие как создание моделей, анализ их поведения, просмотр и изменение параметров модели и т.д. Это позволяет сократить время, затрачиваемое на разработку и обновление моделей, и существенно повысить эффективность работы с системами интерактивной графики.
- Используя языки сценариев высокого уровня и соответствующую системную поддержку, можно расширять возможности систем оптического моделирования путём создания пользовательских приложений, классов, подключаемых модулей и настраиваемых элементов графического интерфейса.

Надо отметить, что внедрение и последующая поддержка сценариев требует существенных усилий со стороны разработчиков системы. Поэтому весьма актуальным является вопрос поиска способов, позволяющих минимизировать подобные усилия. В данной работе предлагается опробованный на практике подход, основанный на использовании специального объектно-ориентированного интерфейса, который встраивается в цепочку базовых классов приложения и предоставляет всю необходимую инфраструктуру для дальнейшего расширения системы.

2. Анализ существующих решений

Можно выделить следующие общие черты, присущие многим ныне существующим системам оптического моделирования, в контексте поддержки сценариев.

- Возможность выполнения определённых действий из командной строки, что может использоваться для автоматизации часто повторяющихся задач. Сценарии в этом случае пишутся на языке командного процессора CMD, PowerShell в среде Windows или Bash в среде Linux и содержат последовательности команд, дополненные необходимыми аргументами. Множество всех возможных команд, как правило, описывается в документации к конкретной системе. Подобные сценарии не имеют большой гибкости, однако их использование является для пользователя достаточно лёгкой задачей, не требующей от него какой-либо существенной квалификации.
- Наличие поддержки одного или нескольких универсальных языков высокого уровня, например, Python или VBScript. Данная поддержка подразумевает наличие в системе специального модуля, называемого интерпретатором сценариев. Одновременно с этим имеется определённый программный интерфейс (API), в рамках которого происходит взаимодействие между сценариями и различными компонентами системы. Сценарии на языке высокого уровня являются гораздо более гибкими, но при этом от пользователя требуются, как минимум, базовые знания используемого языка и задействованного API.

- Присутствие специальных инструментальных средств для редактирования и отладки сценариев. Это может быть как простой текстовый редактор, так и продвинутый редактор сценариев, поддерживающий дополнительные функции, например, отладку во время выполнения, использование точек останова, подсветку синтаксиса и т.д.

Рассмотрим особенности поддержки сценариев в некоторых системах интерактивной графики более подробно.

Система Autodesk Maya имеет поддержку сразу нескольких языков сценариев: MEL (Maya Embedded Language) и Python. Сценарии, написанные на языке MEL, используют процедурный MEL API [1], обладающий очень богатым функционалом. Как пример, на данном языке написана большая часть окружения Maya и сопутствующих расширений. Имеется возможность записи действий в сценарий на MEL, т.е. создание макросов. Среди недостатков данного языка можно отметить отсутствие объектной ориентации и использование собственного синтаксиса, что требует наличия у пользователя определённых навыков. Maya Python API [2, 3] свободен от вышеперечисленных недостатков, однако этот API очень сложен для большинства прикладных задач из-за своего весьма низкого уровня. Сценарии можно создавать, запускать и сохранять непосредственно в Autodesk Maya с помощью встроенного редактора сценариев. В равной мере поддерживается выполнение сценариев командной строки.

Весьма популярный пакет программ для создания трёхмерной компьютерной графики Blender также поддерживает автоматизацию и расширение с помощью языка Python. Для взаимодействия сценариев и различных компонентов системы используется Blender Python API [4]. Запуск сценариев можно осуществлять из встроенного редактора сценариев или из командной строки. Среди недостатков можно отметить то, что доступ к объектам модели осуществляется через наборы коллекций, что, на наш взгляд, является далёким от объектно-ориентированного стиля. Также в Blender отсутствует автоматические преобразования внутренних структур в их аналоги, доступные через интерфейс Blender Python API, что требует от разработчиков этого пакета программ явного дублирования усилий и может приводить к пропущенной функциональности [5].

Аналогичные недостатки присутствуют и в системе трёхмерного моделирования FreeCAD, поддерживающей язык Python. Любое расширение внутреннего C++ API этой системы требует ручного создания привязок к Python, что осуществляется с помощью дополнительных заголовочных и xml-файлов [6]. При этом надо отметить, что в некоторых случаях данная проблема частично решается с помощью применения методов автоматической генерации привязок между объектами C++ и объектами того или иного языка сценариев [7, 8, 9]. Отдельно можно выделить библиотеку CLIF [10], позволяющую создавать обёртки для Python с помощью статического анализа C++ кода, что обеспечивает более стабильную и безопасную работу с этим кодом из Python [11].

Система оптического моделирования Rhinoceros 3D включает в себя программный инструмент Rhinoscript, основанный на технологии COM. Этот инструмент используется для создания и запуска сценариев, написанных на языке Microsoft VBScript. Пользователю доступен Rhinoscript API, содержащий более пятисот различных методов, позволяющих легко автоматизировать большинство прикладных задач [12]. Среди недостатков можно отметить отсутствие возможности расширения системы с помощью сценариев. Их запуск из командной строки также не поддерживается, что может затруднять автоматизацию части задач. Дополнительно нужно подчеркнуть, что COM-разработка – это довольно сложное занятие, требующее от разработчиков явного выполнения множества рутинных задач, что усложняет поддержку нового функционала в сценариях по мере развития системы.

Поддержка сценариев в САПР CATIA также основана на технологии COM. Поддерживается запуск сценариев как из графического интерфейса пользователя, так и из интерфейса командной строки. Особенностью этой САПР является то, что она может быть оснащена

внешним интерпретатором сценариев. В этом случае для разработки сценариев пользователи могут использовать любой язык, поддерживающий технологию COM, например, Python, Java или C#. САПР САТИА поставляется с достаточно функциональным редактором сценариев и имеет большой и хорошо документированный Automation API [13]. Недостаток, связанный с высокой сложностью COM разработки, актуален и для данной системы.

3. Описание подхода

В этом разделе описаны основные возможности нашего подхода и детали его реализации.

3.1 Концепция целевого интерфейса

Разрабатываемая нами система оптического моделирования реализует строгую модульную архитектуру, в которой выделяются две главные группы компонентов: модули вычислительного ядра и модули графического интерфейса пользователя [14]. Важно отметить тот факт, что весь функционал системы сосредоточен во множестве компонентов ядра, и это позволяет использовать его в отрыве от других модулей. Эта особенность делает возможным использование различных клиентских сторон пользовательского интерфейса (frontend), например, консольных или сетевых без глубокой переделки всей системы. В сильно упрощённом виде функциональная схема системы, используемая нами, показана на рис. 1.

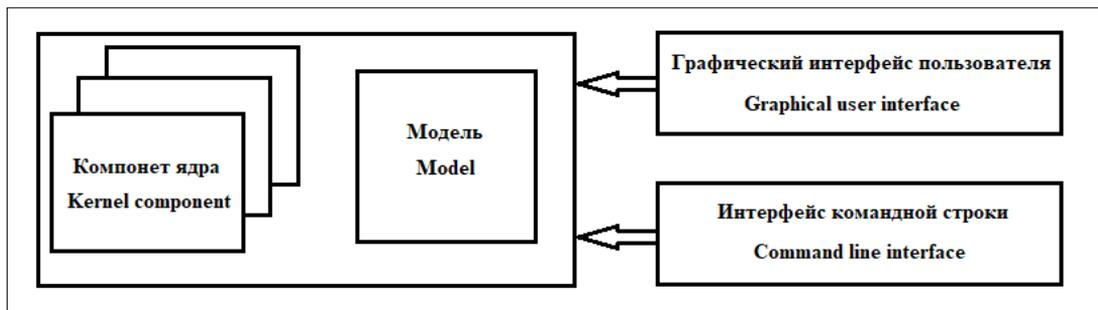


Рис. 1. Функциональная схема системы
Fig. 1. Functional scheme of the system

Для взаимодействия как между собой, так и с загруженной моделью (сценой) все компоненты нашей системы используют так называемый унифицированный целевой интерфейс. Этот интерфейс строится из требований к графическому интерфейсу, т.е. полнота, непротиворечивость и достаточность. Он реализуется двумя дружественными базовыми C++ классами: Entity и EntityType. Данный интерфейс предоставляет набор операций, позволяющий работать с объектами системы в терминах свойств и действий.

Под свойствами подразумеваются внутренние атрибуты объекта. Это могут быть как элементарные типы данных, например, скаляры или массивы, так и ссылки на другие объекты системы. Целевой интерфейс предоставляет специальный метод для определения свойства объекта, с помощью которого задаётся имя свойства и тип его значений. Дополнительно целевой интерфейс предоставляет виртуальные методы для чтения и установки значения свойств SetProperty() и GetProperty().

Под действиями подразумеваются виртуальные функции (методы) класса, которые могут иметь параметры и возвращать результат ExecProc() и ExecFunc(). Параметрами и результатами этих функций могут быть как простые объекты, например, элементарные типы данных, так и ссылки на любые объекты системы, реализующие целевой интерфейс. Также целевой интерфейс предоставляет специальный метод для определения действий,

позволяющий задать имя действия, типы параметров, тип результата и метод класса, непосредственно исполняющий каждое конкретное действие.

Все классы производные от `Entity` реализуют функционал объектов системы, а классы производные от `EntityType` реализуют тип объекта. Экземпляров объектов может быть сколь угодно много, однако каждый тип объекта существует в системе в единственном числе и имеет глобальную точку входа (паттерн *singleton*). Классы типов используются для создания новых экземпляров объектов с помощью механизма конструкторов.

3.2 Взаимодействие с Python API

Возможность работы со свойствами и действиями существующих объектов системы исключительно через интерфейс базового класса `Entity` обеспечивает их совместимость как между собой, так и с будущими (ещё не разработанными) объектами и классами. Поэтому поддержка сценариев на языке высокого уровня Python была добавлена в нашу систему с учётом этой ключевой особенности. Интерпретатор языка сценариев Python был встроен в нашу систему с помощью пакета `CPython` [15], дополненного специальным модулем, реализующим унифицированный целевой интерфейс. По сути, интерпретатор сценариев стал ещё одной клиентской стороной нашей системы, позволяющей выполнять сценарии как из графического интерфейса пользователя, так и из командной строки. Более подробную информацию об интеграции данного языка можно найти в [16].

Благодаря подобной реализации, при расширении системы новыми классами разработчикам более не требуется писать какой-либо дополнительный код для доступа к этим классам из сценариев. Новые классы становятся доступными из сценариев автоматически при условии, что они будут наследовать унифицированный целевой интерфейс.

На листинге 1 приведён пример кода, который определяет новый тип целого числа, хранящего только положительные значения (основной язык C++).

```
PosIntType::PosIntType()
    : EntityType("PosInt")
    {
    Def(new Property(this, "value", &PosInt::m_value, TIntVal,
        READ_ONLY));
    Def(new XAction(this, "Set", &PosInt::Set,
        Arg("value", TIntVal));
    }
```

Листинг 1. Пример определения нового типа

Listing 1. An example of a new type definition

Новый тип наследуется от базового `EntityType`. В его конструкторе определяется свойство `value`, которое будет использоваться для чтения значения, и действие `Set` для его установки. Для создания объекта данного типа определяется функция *конструктор* (основной язык C++), листинг 2.

```
Entity *PosIntType::ConstructObject() const
{
    return new PosInt();
}
```

Листинг 2. Функция конструктор

Listing 2. Constructor function

Свойства и действия целевого интерфейса явно преобразуются в конструкции языка Python, поэтому сценарии могут использовать обычную точечную нотацию для доступа к наборам свойств и действий, как показано на листинге 3 (основной язык Python):

```
x = PosInt() # вызов конструктора объекта типа PosInt
x.Set(10) # установка нового значения через вызов действия Set
```

```
print(x.value) # чтение значения через обращение к свойству value
```

Листинг 3. Точечная нотация

Listing 3. Dot notation

Такой подход позволяет разрабатывать сценарии с использованием устоявшегося синтаксиса языка Python. В нашем случае нет необходимости использовать какие-либо дополнения или отдельные синтаксические конструкции для взаимодействия с объектами системы. От пользователя требуются лишь базовые знания языка Python.

Другим преимуществом данного подхода является то, что получившийся в итоге интерфейс программирования сценариев (Python API) полностью совпадает с унифицированным целевым интерфейсом системы с точки зрения иерархии объектов, их свойств и действий. На рис. 2 показан пример такой иерархии классов, определяющий различные типы узлов модели (сцены).

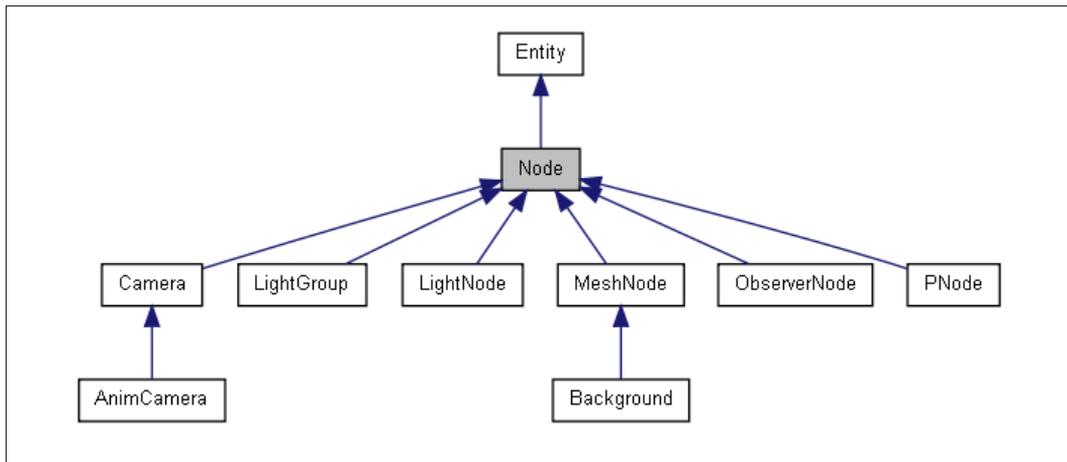


Рис. 2. Иерархия классов узлов модели

Fig. 2. Hierarchy of node classes of the model

Отдельно нужно отметить один важный момент, связанный с тем, что целевой интерфейс из-за использования языка сценариев теперь выходит непосредственно на конечного пользователя. От этого интерфейса требуется быть удобным для человека, т.е. по возможности понятным, естественным и непротиворечивым. Сюда можно включить использование разработчиками понятных имён свойств, действий и типов, использование свойств там, где это возможно, вместо избыточных Get/Set методов и т.д.

3.3. Параметрические объекты

Автоматизация моделирования с помощью сценариев, хранящихся вне самой модели, например, в виде файлов на диске, может быть затруднена, так как изменение параметров модели часто требует изменение самого сценария. Данный недостаток, присущий многим другим системам интерактивной графики, был решён нами с помощью использования специальных параметрических объектов, которые, с одной стороны, являются частями модели, а с другой стороны, содержат в себе сценарий на языке Python, представляющий из себя класс расширения.

С технической точки зрения параметрические объекты ничем не отличаются от других объектов системы, реализующих унифицированный целевой интерфейс. Однако подобные объекты имеют среди своих свойств специальное скрытое поле, содержащее программный код класса расширения. В отличие от обычных сценариев, являющихся набором последовательно выполняющихся подпрограмм, сценарии классов расширений не имеют явной точки входа, с которой начинается их выполнение. Вместо этого для взаимодействия с

остальной системой используются функции обратного вызова, которые вызываются ядром комплекса оптического моделирования. Классы расширений позволяют задавать, а затем многократно изменять параметры сценарного объекта без изменения кода самого класса расширения.

Классы расширений полностью поддерживают парадигму объектно-ориентированного программирования, т.е. объектную ориентацию, наследование и полиморфизм, тем самым предоставляя пользователю удобную возможность расширения конечной системы новым функционалом.

На листинге 4 приведён пример кода, который определяет новый класс расширения, описывающий тип геометрического объекта в форме параллелепипеда (основной язык Python):

```
class Box(PMesh):
    # Definition of parameters
    org = Param("Origin", VectType(Size))
    size = Param("Size", VectType(Size(0.001)), (300, 200, 100))
    # Creation method
    def Init(self):
        # Create parts
        self.AddPart("Side")
    # Evaluation method
    def Eval(self):
        # Calculate box boundaries
        x1, y1, z1 = self.org
        length, width, height = self.size
        x1 -= length / 2; y1 -= width / 2; z1 -= height / 2
        x2 = x1 + length; y2 = y1 + width; z2 = z1 + height
        # Reconstruct box geometry
        self.ClearGeometry()
        self.AddVerts(0, [[x1, x2]], [[y1, y2]], [[z1, z2]])
        #...
        self.AddVerts(5, [[x1]], [[y1, y2]], [[z1, z2]])
    # Notification handler
    def OnNotify(self, object, reason):
        pass
```

Листинг 4. Определение нового класса расширения

Listing 4. Defining a new extension class

Структура класса расширения состоит из его имени и последующих определений методов и параметров. Параметры задаются через вызов специального метода Param(). Ими могут быть любые типы, реализующие целевой интерфейс. Дополнительно параметру можно задать значение по умолчанию. Специальный метод Init() исполняет роль конструктора и вызывается однократно при создании нового экземпляра объекта. Метод Eval() автоматически вызывается ядром системы при изменении параметров класса. Назначением этого метода является обновление представления объекта в соответствии с новыми значениями параметров.

С точки зрения целевого интерфейса, параметры класса расширения являются свойствами, а методы, соответственно, являются действиями. Это означает, что доступ к ним возможен из любой части системы, например, из графического интерфейса или из сценариев, что демонстрируется на рис. 3.

Так как параметрические объекты являются частями модели (сцены), должна обеспечиваться их корректная работа при загрузке модели в более старую или более новую версию комплекса

Аргументами вызова `NotifyChange()` являются причина уведомления, которая задаётся в каждом конкретном классе в виде констант перечисления, и любые дополнительные параметры, которые могут быть полезны при обработке уведомления в методе `OnChange()` объекта владельца.

Данная функциональность целевого интерфейса была воплощена и в классах расширений. Для этого в каждом экземпляре параметрического объекта имеется специальное скрытое поле, хранящее список ссылок на отслеживаемые объекты. Добавление конкретного объекта в этот список осуществляется с помощью метода `EnableNotify()` получающего объект в качестве аргумента. Как правило, данный метод вызывается в конструкторе класса расширения, но явных ограничений здесь нет, подписаться на приём уведомлений можно в любой момент времени и из любого места сценария. Обработчик уведомлений реализуется в методе `OnNotify()`, который является непосредственной надстройкой над методом `OnChange()` целевого интерфейса.

На листинге 5 приведён пример кода класса расширения, реализующего обработчик, отслеживающий изменение расположения интересующего узла сцены (основной язык Python):

```
# Notification handler
def OnNotify(self, obj, reason):
    # Check the object that issued the notification
    s = GetScene()
    if ((s != None) and (obj == s.GetNode("Box"))):
        # Check reason
        if ((reason is not None) and (reason ==
            NodeNotifyReason.TRANSFORM_CHANGED)):
            # Position of 'Box' node was changed, do some stuff
            ...
```

Листинг 5. Пример кода класса расширения, реализующего обработчик
Listing 5. Example code for an extension class that implements a handler

Параметрические объекты, использующие механизм уведомлений, позволяют отказаться от необходимости опроса состояния отслеживаемых объектов, что существенно повышает эффективность процессов моделирования и экономит системные ресурсы.

4. Результаты

Описанная технология была успешно внедрена в разрабатываемый нами программный комплекс оптического моделирования и реалистичной компьютерной графики Lumisert. Одним из главных преимуществ данной технологии стала возможность расширения функциональности системы без непосредственного участия разработчиков, т.е. силами конечных пользователей, что позволяет выполнять точную настройку и подгонку комплекса под их нужды, особенно в случае необходимости добавления моделирования вычислений тех или иных стандартов и метрик. Это весьма важно в свете того, что различные архитектурно-проектировочные или оптико-дизайнерские компании, как правило, используют различные стандарты освещённости или яркости, поэтому использование сценариев и классов расширений решает сразу несколько проблем.

- Экономическая эффективность. Доработка кода программного комплекса для каждого конкретного заказчика может быть весьма дорогостоящей. Благодаря возможности добавлять расчеты с помощью сценариев, пользователи могут настраивать комплекс в соответствии со своими конкретными потребностями, не требуя внесения изменений в его исходный код. Это позволяет им сэкономить как время, так и деньги.
- Гибкость. Использование сценариев обеспечивает большую гибкость при добавлении новых функций и возможностей. Это связано с тем, что сценарии можно легко изменять

и обновлять, а от пользователей не требуются продвинутые навыки и знания в области программирования.

- **Безопасность.** Изменение кода программного комплекса для каждого заказчика не только дорого, но и небезопасно. Существует высокая вероятность внесения ошибок, которые могут вызвать серьезные проблемы. Сценарии и классы расширений обеспечивают более безопасный подход к настройке.
- **Повышенная масштабируемость.** Сценарии облегчают интеграцию пользовательских объектов и функций в специальные библиотеки, которые впоследствии можно легко расширять по мере возрастания потребностей.

К настоящему времени для нашего комплекса был создан набор из ~200 сценарных расширений, большая часть которых представляет собой классы объектов сцены пользовательских типов, таких как узлы, геометрии, микрогеометрии, а также функции. Оставшаяся часть является различными видами симуляторов.

4.1. Примеры расширений

Одним из практических примеров расширения функциональности нашего программного комплекса является сценарный симулятор определения уровня дискомфорта, вызванного бликами от источников света (Glare Rating). Данный симулятор позволяет моделировать естественное освещение пространства с использованием различных параметров, задаваемых пользователем, и последующим сохранением результатов в файловой системе.

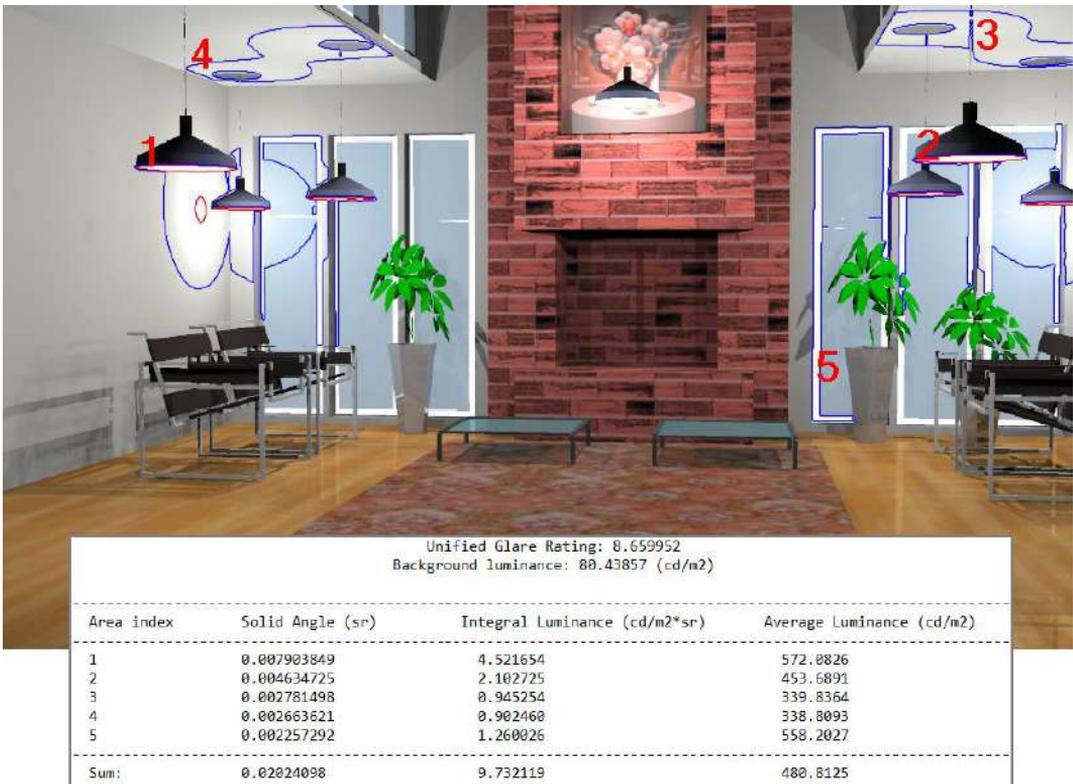


Рис. 5. Симулятор для анализа бликов
Fig. 5. Simulator for glare rating analysis

Основными параметрами симулятора являются: ориентация сцены, положение наблюдателя, яркость источников освещения. Множество дополнительных параметров позволяет

управлять различными характеристиками моделирования, от выбора метода расчета – BRT (Backward Ray Tracing) или PT (Path Tracing) до формата имен генерируемых файлов с результатами.

Результатом работы симулятора является изображение, в котором области яркости для заранее заданного интервала выделяются изолиниями, а общее значение рейтинга бликов (Unified Glare Rating) и детальная информация по каждой области сохраняется в файл отчёта, как это показано на рис. 5.

Другой пример представляет собой сценарное расширение, позволяющее задать геометрию в виде криволинейной поверхности с синусоидальным профилем в сечении, например, если требуется моделирование поверхностей с такими формами. Для того чтобы добавить подобный объект в модель, пользователю достаточно перетащить его методом drag and drop из библиотеки объектов в сцену, затем настроить необходимые параметры, такие как размеры требуемой поверхности, амплитуда и период кривой, шаг разбиения треугольной сетки, образующей криволинейную поверхность, и т.д., что демонстрируется на рис. 6.

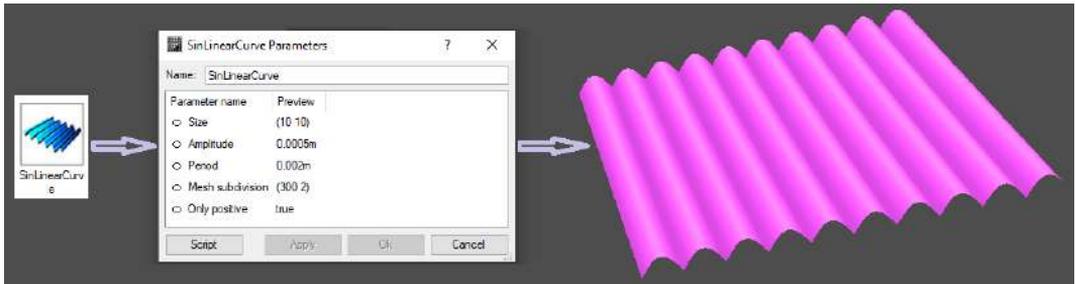


Рис. 6. Параметры сценарного объекта и его отображение в сцене
Fig. 6. Scenario based object parameters and its appearance in the scene

5. Заключение

Описанный подход к поддержке сценариев широко используется в разрабатываемом нами программном комплексе оптического моделирования. Он позволил разработчикам уменьшить количество кода, над которым ведётся единовременная работа в рамках жизненного цикла комплекса, благодаря унификации внутреннего целевого программного интерфейса объектов системы с программным интерфейсом сценариев. Удалось снизить вероятность возникновения ошибок и расхождений функциональности, а также увеличить скорость разработки. Пользователи комплекса, в свою очередь, получили в распоряжение удобный интерфейс сценариев, позволяющий автоматизировать различные прикладные задачи, а также создавать свои собственные эффективные расширения с минимальными затратами времени.

Список литературы / References

- [1] PyMEL for Maya, Available at: <https://help.autodesk.com/cloudhelp/2018/JPN/Maya-Tech-Docs/PyMel/index.html>, accessed 13.02.2023.
- [2] Mechtley A., Trowbridge R. Maya Python for Games and Film: A Complete Reference for the Maya Python API. CRC Press, 2011, 381 p.
- [3] Collado G.E.G. Modeling and Python Scripting in Maya for the Animation Short Style. Bachelor's Thesis. Department of Computer Engineering, California Polytechnic State University, 2016, 22 p.
- [4] Blender 3.4 Python API Documentation, Available at: <https://docs.blender.org/api/current/index.html>, accessed 13.02.2023.
- [5] Anders M. Blender 2.49 Scripting, Packt Publishing, 2010, 282 p.
- [6] Machado F., Malpica N., Borromeo S. Parametric CAD modeling for open source scientific hardware: Comparing OpenSCAD and FreeCAD Python scripts. PLoS ONE, vol. 14, issue 12, 2019, article id e0225795, 30 p.

- [7] Goyal M., Osborne I., Pivarski J. The Awkward World of Python and C++. arXiv preprint arXiv:2303.02205, 2023, 6 p.
- [8] Buse F., Bellmann T. General Purpose Lua Interpreter for Modelica. In Proc. of the 14th Modelica Conference, 2021, pp. 425-431.
- [9] Standish R. C++ Reflection for Python Binding. *Overload*, issue 152, 2019, pp. 11-18.
- [10] C++ Language Interface Foundation (CLIF), Available at: <https://github.com/google/clif>, accessed 30.03.2023.
- [11] Can D., Martinez V.R. et al. Pykaldi: A Python Wrapper for Kaldi. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5889-5893.
- [12] RhinoScript Guides. Available at: <https://developer.rhino3d.com/guides/rhinoscript/>, accessed 13.02.2023.
- [13] Li Y. Research of Integration Technology between CATIA and TOOLMANAGER Based on CAA, *International Journal of Advanced Network, Monitoring and Controls*, vol. 1, issue 1, 2016, pp. 130-135.
- [14] Барладян Б.Х., Волобой А.Г. и др. Интеграция реалистичной графики в системы автоматизированного проектирования и управления жизненным циклом изделия. *Программирование*, том 44, вып. 4, 2018 г., стр. 26-35 / Barladian B.K., Voloboy A.G. et al. Integration of realistic computer graphics into computer-aided design and product lifecycle management systems, *Programming and Computer Software*, vol. 44, issue 4, 2018, pp. 225-232.
- [15] Extending and Embedding the Python Interpreter. Available at: <http://docs.python.org/3/extending/index.html>, accessed 13.02.2023
- [16] Дерябин Н.Б., Жданов Д.Д., Соколов В.Г. Внедрение языка сценариев в программные комплексы оптического моделирования. *Программирование*, том 43, вып. 1, 2017 г., стр. 40-53 / Deryabin N.B., Zhdanov D.D., Sokolov V.G. Embedding the Script Language into Optical Simulation Software // *Programming and Computer Software*, vol. 43, issue 1, 2017, pp. 13-23.
- [17] Галактионов В.А., Дерябин Н.Б., Денисов Е.Ю. Объектно-ориентированный подход к реализации систем компьютерной графики. Информационно-измерительные и управляющие системы, вып. 6, 2009 г., стр. 96-108 / Galaktionov V.A., Deryabin N.B., Denisov E.Yu. Object-oriented approach for computer graphics systems implementation. *Information-measuring and Control Systems*, issue 6, 2017, pp. 96-108 (in Russian).

Информация об авторах / Information about authors

Михаил Сергеевич КОПЫЛОВ – младший научный сотрудник. Сфера научных интересов: языки программирования, компьютерная графика, вычислительная оптика, трассировка лучей

Mikhail Sergeevich KOPYLOV, junior researcher. Research interests: programming languages, computer graphics, computing optics, ray tracing techniques.

Николай Борисович ДЕРЯБИН – научный сотрудник. Сфера научных интересов: архитектура систем оптического моделирования, компьютерная графика, вычислительная оптика.

Nikolai Borisovich DERYABIN, researcher. Research interests: architecture of optical simulation software systems, computer graphics, computing optics.

Евгений Юрьевич ДЕНИСОВ – научный сотрудник. Сфера научных интересов: компьютерная графика, создание программных систем оптического моделирования, автоматическое тестирование, параллельные и распределённые вычисления.

Evgeniy Yuryevich DENISOV, researcher. Research interests: computer graphics, elaboration of optical simulation software systems, automatic testing, concurrent and distributed computing.



Об использовании открытых сторонних библиотек при программной реализации вихревых методов вычислительной гидродинамики

^{1,2} И.К. Марчевский, ORCID: 0000-0003-4899-4828 <iliamarchevsky@mail.ru>

¹ Ю.А. Измайлова, ORCID: 0000-0003-1137-3235 <yulia.izmailova@mail.ru>

¹ М.А. Ерофеева, ORCID: 0009-0000-1061-6974 <mariya.a.erofeeva@gmail.com>

¹ Д.Ю. Кобзарь, ORCID: 0009-0006-0746-6497 <cobzardash@yandex.ru>

¹ Московский государственный технический университет имени Н.Э. Баумана,
105005, Россия, г. Москва, ул. 2-я Бауманская, д. 5, к. 1

² Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

Аннотация. Обсуждается наиболее общая структура вычислительного алгоритма, реализующего бессеточные лагранжевы методы вычислительной гидродинамики. Затронуты не только основные, но и «вспомогательные», но оттого не менее важные процедуры, реализациям которых часто практически не уделяется внимания. Последнее может приводить к значительному дисбалансу и снижению эффективности кодов, в которых «основные» вычислительные операции существенно оптимизированы. Обсуждаются авторские коды VM2D и VM3D, развитие которых на первом («поисковом») этапе шло главным образом по пути выбора и реализации необходимых математических моделей, а достижение приемлемой эффективности обеспечивалось «экстенсивным» путем – привлечением значительных вычислительных ресурсов (в частности, видеокарт). Предпринята попытка сделать заключение о целесообразности использования существующих сторонних библиотек для выполнения операций вычислительной геометрии, решения задач на графах и т.п.

Ключевые слова: бессеточные лагранжевы методы вычислительной гидродинамики; вихревые методы; код VM2D; код VM3D; алгоритмы вычислительной геометрии; свободные библиотеки.

Для цитирования: Марчевский И.К., Измайлова Ю.А., Ерофеева М.А., Кобзарь Д.Ю. Об использовании открытых сторонних библиотек при программной реализации вихревых методов вычислительной гидродинамики. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 181-200. DOI: 10.15514/ISPRAS-2023-35(2)-13

On open third-party libraries usage in implementation of vortex particle methods of computational fluid dynamics

^{1,2} I.K. Marchevsky, ORCID: 0000-0003-4899-4828 <iliamarchevsky@mail.ru>

¹ Yu.A. Izmailova, ORCID: 0000-0003-1137-3235 <yulia.izmailova@mail.ru>

¹ M.A. Erofeeva, ORCID: 0009-0000-1061-6974 <mariya.a.erofeeva@gmail.com>

¹ D.Yu. Kobzar, ORCID: 0009-0006-0746-6497 <cobzardash@yandex.ru>

¹ Bauman Moscow State Technical University,
105005, Moscow, 2-nd Baumanskaya st., 5.

² Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

Abstract. The most general structure of a computational algorithm that implements meshless Lagrangian methods of computational fluid dynamics is discussed. Not only the main ones are touched upon, but also “auxiliary”, but therefore no less important procedures, which implementation is often practically ignored. The latter can lead to a significant imbalance and decrease in the efficiency of codes in which the “basic” computational operations are significantly optimized. The author's in-house codes VM2D and VM3D are discussed, the development of which at the first (“exploratory”) stage proceeded mainly along the path of choosing and implementing the necessary mathematical models, and the achievement of acceptable efficiency was ensured by an “extensive” way – involving significant computing resources (in particular, graphical accelerators). An attempt was made to make a conclusion about the expediency of using existing third-party libraries to perform computational geometry operations, solve problems on graphs, etc..

Keywords: meshless Lagrangian CFD methods; vortex particle methods; VM2D code; VM3D code; computational geometry algorithms; open source libraries

For citation: Marchevsky I.K., Izmailova Yu.A., Erofeeva M.A., Kobzar D.Yu. On open third-party libraries usage in implementation of vortex particle methods of computational fluid dynamics. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 181-200 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-13

1. Введение

Задачи вычислительной гидро- и газовой динамики традиционно и вполне заслуженно относят к наиболее вычислительно сложным, прежде всего по причине нелинейности уравнений и определяющих соотношений, а также необходимости воспроизведения процессов, протекающих на различных пространственных и временных масштабах – от малых и даже подсеточных (если говорить об использовании традиционных численных методов) до больших, соизмеримых со всей расчетной областью.

В то же время в инженерной практике приходится сталкиваться с целым классом задач моделирования, в которых течение среды как таковое не представляет самостоятельного интереса, однако его необходимо учитывать, чтобы правильно определять величины аэрогидродинамических нагрузок, действующих на находящиеся в потоке тела. Приближенные методы, появившиеся еще в «докомпьютерную» эру, такие как расчет по стационарным коэффициентам или использование коэффициентов присоединенных масс (и те, и другие могут быть определены экспериментально или в результате моделирования сравнительно простого режима обтекания – стационарного или потенциального соответственно), вполне пригодны и не потеряли актуальности, но лишь для узкого диапазона условий: (квази)стационарного обтекания неподвижных или медленно движущихся тел в практически постоянном потоке, или, наоборот, вибрирующих в неподвижной среде тел. В существенно же нестационарных условиях обтекания, сопровождаемого интенсивным вихреобразованием, отрывами потока и т.п., указанные методики неприменимы.

Простого и универсального решения у данной проблемы нет; в то же время, если ограничиться рассмотрением сравнительно простых моделей среды – однофазной,

однородной, несжимаемой, нетеплопроводной – то подобные течения возможно моделировать с использованием т.н. вихревых методов вычислительной гидродинамики. Последние, по существу, восходят к классическим работам Н.Е. Жуковского, связавшего многие наблюдаемые гидродинамические эффекты с вихревой природой течения. Не вдаваясь в историю их развития (см. обстоятельные обзоры [1–3]), отметим лишь следующее обстоятельство: после этапа интенсивного развития в 60-х годах XX века – в начале «компьютерной эры» (а мощности ЭВМ того времени позволяли реализовать лишь их, для более привычных на сегодня сеточных методов вычислительных ресурсов было явно недостаточно) – наступил этап некоторого «застоя», во время которого, наоборот, бурное развитие получили сеточные методы. Их очевидными преимуществами являются универсальность, возможность моделирования в широком диапазоне свойств среды в частности и постановок задач в целом, сравнительная простота перехода к решению междисциплинарных задач и др. Вихревые методы при этом заняли свою, довольно узкую нишу, и продолжили планомерно развиваться, превратившись на сегодняшний день в довольно мощный инструмент инженерного анализа.

2. Структура вычислительного алгоритма в вихревых методах

Особенностью вихревых методов в частности и лагранжевых методов частиц в целом, существенно отличающей их от классических сеточных, является структура «вычислительного конвейера». В алгоритмах сеточных методов на каждом шаге расчета по времени решается некоторая типовая задача: строится разностный аналог оператора задачи (как правило, дифференциального), представляющий собой систему алгебраических уравнений, которая, в свою очередь, решается тем или иным итерационным методом; при этом могут активно использоваться различные схемы интерполяции, производятся решения некоторых вспомогательных задач и т.п. При решении междисциплинарных задач в большинстве случаев используют схемы расщепления, рассматривая таким образом на каждом «подшаге» задачу только для одной неизвестной величины, однако это не нарушает общей логики – выполнения сходных операций для всех узлов или ячеек сетки и решения разреженных систем уравнений.

В вихревых методах на каждом шаге расчета приходится решать достаточно большое количество совершенно разнотипных задач, конкретный перечень которых зависит от рассматриваемой модификации. Если попытаться представить в наиболее общем виде алгоритм, выполняемый на каждом временном шаге, то можно выделить следующие наиболее трудозатратные блоки операций:

- 1) моделирование процесса генерации завихренности на обтекаемых поверхностях;
- 2) вычисление скоростей движения вихревых частиц;
- 3) восстановление поля скоростей и поля давления; расчет гидродинамических нагрузок, действующих на обтекаемые тела;
- 4) моделирование эволюции завихренности в области течения и движения обтекаемых поверхностей.

Примерная схема вычислительного алгоритма представлена на рис. 1.

При этом следует иметь в виду, что каждый из отмеченных пунктов включает в себя целый перечень подзадач, часто совершенно различных по своим «характеристикам»; например, к первому блоку относятся такие процедуры, как

- построение системы линейных алгебраических уравнений – дискретного аналога граничного (вообще говоря, сингулярного или гиперсингулярного) интегрального уравнения, где нетривиальным является не только шаг формирования матрицы, аппроксимирующей оператор, но и расчет правой части;
- решение возникающей линейной системы с полностью заполненной матрицей.

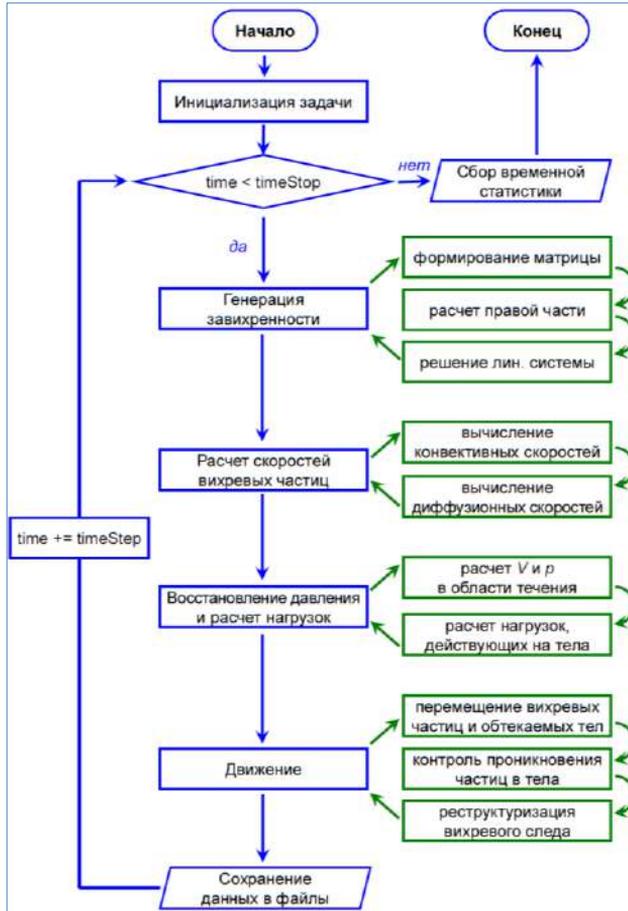


Рис. 1. Общая схема алгоритма вихревых методов
 Fig. 1. The general scheme of the algorithm of vortex methods

Второй блок включает в себя операции вычисления скоростей движения вихревых частиц, как конвективных, т.е. фактически скоростей среды в точках расположения вихревых частиц, так и диффузионных, выражающих влияние вязкости среды. Задача расчета конвективных скоростей вихревых частиц аналогична задаче N тел. Отметим, что известно несколько подходов к моделированию вязких эффектов в вихревых методах; в качестве альтернативы упомянутому методу диффузионных скоростей могут быть использованы метод случайных блужданий (Random walk), метод обмена интенсивностями (Particle strength exchange), метод деформирования ядра вихря (Core spreading) и некоторые другие; данные методы далее рассматривать не будем; общее представление о них можно получить из обзора [3].

Для расчета гидродинамических нагрузок, выполняемого в рамках третьего блока, могут применяться различные алгоритмы; наиболее удобным представляется использование обобщений интегралов Бернулли и Коши – Лагранжа [4] и большого количества их модификаций, адаптированных для решения задач различных типов [5–7]. Возможно как восстановление поля давления с последующим его интегрированием по поверхности тела, так и определение «интегральных» величин нагрузок – главного вектора сил и главного момента. Соответствующие вычислительные алгоритмы либо сравнительно просты, либо аналогичны тем, что используются при решении задачи типа N тел. Операция восстановления поля скоростей среды в заданных точках по существу идентична расчету конвективных скоростей вихревых частиц.

Четвертый блок операций предполагает реализацию таких алгоритмов как

- расчет перемещений вихревых частиц и обтекаемых поверхностей, включая «сопряжение» представления завихренности в виде вихревого слоя на обтекаемой поверхности и ее же представления в виде вихревых структур в области течения;
- контроль проникновения завихренности внутрь обтекаемых поверхностей;
- реструктуризация вихревого следа.

В абсолютном большинстве публикаций по вихревым методам авторы уделяют внимание лишь отдельным вопросам, чаще всего – реализации алгоритмов моделирования движения вихревых частиц, а также схемам дискретизации граничного интегрального уравнения. Не подвергая сомнению тот факт, что данные вопросы являются наиболее сложными и, как следствие, интересными для исследования, отметим, что в области решения указанных проблем к настоящему времени накоплен значительный опыт и достигнут существенный прогресс: в литературе известны приближенные быстрые методы, имеющие квазилинейную вычислительную сложность, позволяющие эффективно рассчитывать взаимное влияние вихревых частиц, а также разработаны весьма точные, робастные, и при этом обладающие невысокой вычислительной сложностью схемы для численного решения ГИУ (расшифровки этой и остальных используемых в статье аббревиатур, исключая названия организаций и названия конкретных кодов, если последние не имеют непосредственной расшифровки, для удобства сведены в таблицу в конце статьи). Тем не менее, перечисленные алгоритмы во многих случаях пока находятся на разных стадиях методических исследований и «отработки», и далеко не всегда могут быть включены в программные реализации.

3. Программные реализации вихревых методов

Несмотря на то, что вихревые методы имеют по крайней мере 60-летнюю историю развития, как в нашей стране, так и за рубежом, их современные модификации практически не реализованы в программных комплексах, которые были бы доступными широкому кругу исследователей. Известны реализации так называемых панельных методов, которые можно рассматривать лишь как самые простые версии вихревых методов, однако они не раскрывают истинный потенциал современных математических моделей и алгоритмов. Подобное положение дел, безусловно, негативно сказывается на популярности вихревых методов среди исследователей и расчетчиков.

Программные реализации вихревых методов стали появляться лишь в последнее время, начиная примерно с 2017-2018 года, некоторые из них поддерживаются и развиваются, при этом практически все они представлены с открытым исходным кодом, что вписывается в актуальную тенденцию отказа от использования коммерческого программного обеспечения для моделирования в механике сплошной среды и перехода к свободно распространяемому ПО с открытыми исходными кодами.

Одной из немногих доступных реализаций вихревых методов применительно к моделированию плоских течений является опубликованный в 2018 году программный комплекс `vvflow`, доступный по ссылке <https://packagecloud.io/vvflow/stable>, в котором реализован метод вязких вихревых доменов, разработанный проф. Г.Я. Дынниковой [5, 8–10]. Возможности программного комплекса позволяют, в том числе решать сопряженные задачи в гидроупругой постановке для систем недеформируемых профилей с упругими связями между ними. Из технологий параллельных вычислений в `vvflow` использована лишь `OpenMP`, что позволяет проводить расчеты только на системах с общей памятью.

В 2018 году появились пакеты с открытым кодом `Omega2D` и `Omega3D`, (доступные соответственно из репозитория <http://github.com/Applied-Scientific-Research/Omega2D> и <http://github.com/Applied-Scientific-Research/Omega3D>), реализующие методы вихревых частиц для моделирования плоских и пространственных течений и описанные в [11]. Данные коды позиционируются разработчиками как платформы для разработки и реализации

различных модификаций вихревых методов; существенное внимание уделено вопросам визуализации. На данный момент в указанных пакетах нет возможности моделирования обтекания подвижных профилей/тел, не вполне ясен вопрос о точности вычисления действующих на обтекаемые поверхности гидродинамических нагрузок (по умолчанию их вычисление вообще не производится). Кроме того, при расчете взаимного влияния вихревых частиц используется только «прямая» реализация закона Био – Савара (т. е. алгоритм квадратичной вычислительной сложности), ускорение вычислений может быть достигнуто лишь за счет подключения внешних библиотек, позволяющих выполнять часть вычислительной работы на графических ускорителях.

В 2019 году появился код FLOWVPM, являющийся частью более крупного проекта FLOWUnsteady (<http://github.com/byuflowlab/flowunsteady>), свободно доступный с исходным кодом, написанным на языке Julia. Пакет позволяет моделировать пространственные течения и обтекание тел, в том числе подвижных, сложной геометрической формы; описание реализованных в нем моделей и методов приведено в [12]. Технологии распараллеливания вычислений в доступной версии кода в явном виде не используются (однако средства языка Julia позволяют это делать неявно); для расчета скоростей вихревых частиц использован быстрый метод мультиполей. В качестве вихревых частиц используются вихревые сгустки (vortex blobs), таким образом, поле завихренности в общем случае является несоленоидальным.

В диссертациях [13, 14] описаны программные пакеты, реализующие метод вязких вихревых доменов для моделирования плоских течений, однако недоступные для широкого круга потенциальных пользователей. Известно также о разработанном проф. G. Morgenthal (ФРГ) и активно используемом в его научной школе пакете VXflow (<https://www.igorkav.com/category/vxflow/>), созданном для решения задач промышленной аэродинамики зданий и сооружений в двумерной и квази-трехмерной постановке; в данном пакете реализован метод случайных блужданий [15] и оригинальная модификация быстрого метода расчета скоростей вихревых частиц [16]; основные модели и алгоритмы, положенные в основу кода, описаны в [17]. Алгоритмы распараллеливания вычислений на кластерных системах и возможности выполнения расчетов на графических картах в VXflow не предусмотрены.

По публикациям известно также о существовании кодов, реализующих вихревые методы (метод дискретных вихрей в различных модификациях, метод замкнутых вихревых рамок), в ВВИА им. Н.Е. Жуковского, МГУ им. М.В. Ломоносова, НИИ Парашютостроения, ЦАГИ им. Н.Е. Жуковского, Санкт-Петербургском государственном морском техническом университете, Институте машиноведения им. А.А. Благонравова РАН, Институте математики им. С.Л. Соболева СО РАН и некоторых других российских организациях; из зарубежных разработок отметим оригинальные методы моделирования двумерных вязких течений, описанные и реализованные в коде, разработанном итальянскими авторами [18]; давно и успешно развиваются методы моделирования пространственных течений в Японии в научной школе проф. K. Kamemoto [19].

Значительный вклад как в развитие вихревых методов, особенно применительно к моделированию течений в пространственной постановке, так и в решение проблем их эффективной программной реализации внесли G.-H. Cottet (Франция), P.D. Koumoutsakos (Греция, Швейцария), G. Winkelmanns (Бельгия) и др. Тем не менее, о разработанных ими кодах известно лишь по публикациям, для свободного доступа они недоступны. Известно также о создании программ авторами J.S. Uhlman, J.S. Marshal (США) и другими.

Коллективом авторов [20, 21] разработан пакет VM2D, свободно доступный по ссылке <https://github.com/vortexmethods/VM2D> с исходным кодом на языке C++. Он предназначен для решения широкого круга плоских задач, в том числе в сопряженной гидроупругой постановке. В отличие от остальных кодов в нем реализованы усовершенствованные схемы повышенной точности для моделирования генерации завихренности [22], имеется удобный

интерфейс для решения большого числа однотипных задач на кластерных системах. Код распараллелен с применением технологий OpenMP, MPI и CUDA, включая возможность их совместного использования. Однако многие из реализованных в нем алгоритмов являются «тривиальными», т.е. реализуют наиболее очевидные, но, к сожалению, далеко не самые эффективные способы выполнения соответствующих операций.

Аналогичным образом устроен и разрабатываемый авторами код VM3D [23], реализующий оригинальную модификацию вихревого метода для моделирования пространственных течений. Отметим, что для решения пространственных задач в VM3D реализован оригинальный алгоритм построения дискретного аналога граничного интегрального уравнения (также записываемого не так, как в большинстве известных реализаций), в основе которого – схема Галеркина. Для вычисления возникающих при этом двойных сингулярных интегралов по треугольным панелям можно вместо авторского алгоритма [24] использовать несколько менее эффективный, но весьма универсальный метод Тейлора – Даффи, достаточно подробно описанный в [25]; свободная и открытая реализация данного алгоритма имеется в составе кода `scuff-em`, предназначенного для решения задач электростатики и электродинамики и доступного из репозитория <https://github.com/HomerReid/scuff-em> под лицензией GPL-3 (модуль `TaylorDuffy`; пример вычисления интегралов имеется в документации к коду).

Далее рассмотрим пути оптимизации обсуждавшихся выше вычислительных алгоритмов, в основе которых лежит использование сторонних библиотек. При этом будем говорить лишь о высокоуровневых, или прикладных библиотеках, не «спускаясь» на уровень, к примеру, библиотек типа OpenMP или Intel TBB, которые, без сомнений, могут быть крайне эффективны при выполнении отдельных частных операций сравнительно «низкого» уровня (параллельное исполнение витков циклов, параллельная сортировка, операции редукции и конкурентного доступа к данным и т.п.). Не будем также обсуждать использование «сервисных» библиотек, обеспечивающих, к примеру, инициализацию задач и сохранение результатов расчетов. Укажем лишь, что для чтения исходных данных, параметров расчетов и разнообразных настроек, обычно хранимых в текстовых файлах, удобно использовать какой-либо универсальный парсер, например, `toml11`, доступный из репозитория <https://github.com/ToruNiina/toml11> (лицензия MIT). При сохранении результатов расчетов, в том числе с целью визуализации целесообразно пользоваться возможностями открытой библиотеки VTK (<https://www.vtk.org>, лицензия BSD) и универсального постпроцессора Paraview (<https://www.paraview.org>, лицензия BSD)

4. Операции линейной алгебры

Несмотря на относительную простоту выполняемых в вихревых методах операций линейной алгебры (к примеру, решение линейной системы с заполненной хорошо обусловленной матрицей или обращение такой матрицы), сравнительно большая размерность задачи, достигающая, особенно при моделировании пространственного обтекания тел, десятков тысяч, требует эффективного выполнения соответствующих процедур.

4.1 Выполнение расчетов на CPU. Библиотека Eigen

При проведении вычислений на центральном процессоре представляется эффективным использование библиотеки Eigen. Библиотека развивается с 2006 года сообществом TuxFamily и Inria, последние версии доступны на сайте <https://eigen.tuxfamily.org> и распространяются под свободной лицензией MPL 2.0. Особенность библиотеки – отсутствие необходимости ее отдельной компиляции: исходный код представляет собой набор заголовочных файлов, непосредственно включаемых директивами `#include` в необходимые модули пользовательского ПО.

К ее преимуществам следует отнести простоту интеграции в код, отсутствие зависимостей с другими библиотеками, большое количество реализованных алгоритмов, близкую к естественной математической нотации записи выражений. Важное значение имеет наличие качественно подготовленной обстоятельной документации. Имеется возможность выполнения некоторых операций в параллельном OpenMP-режиме на ЭВМ с общей памятью: к таким относятся умножение матриц, LU-разложение с частичным выбором ведущего элемента (PartialPivLU), итерационное решение линейных систем методами сопряженных и бисопряженных градиентов (ConjugateGradients, BiCGStab), реализация метода наименьших квадратов с использованием метода сопряженных градиентов (LeastSquaresConjugateGradient). При наличии установленных на компьютере пользователя реализаций BLAS/LAPACK библиотека Eigen может выступать в качестве интерфейса к ним; то же относится к Intel MKL.

Высокая производительность Eigen достигается благодаря логике построения библиотеки на основе т.н. «ленивых» шаблонов: это означает (не погружаясь в подробности), что на выходе получается абстрактное синтаксическое дерево правой части исходного выражения в виде шаблонных объектов, известных на стадии компиляции. Это дерево превращается в реальный исполняемый код только при исполнении оператора присваивания. Обратной стороной такого подхода является «плохопредсказуемое» поведение при написании кода, в котором используются операции неявного приведения типов или автоматического вывода типа. К примеру, исполнение кода

```
Eigen::MatrixXd A({ { 1.0, 2.0 } }, { 3.0, 4.0 } });  
Eigen::MatrixXd B({ { 1.0, 0.0 } }, { 0.0, 1.0 } });  
auto C = A * B;  
Eigen::MatrixXd R1 = C;  
B(0,0) = B(1,1) = 2.0;  
Eigen::MatrixXd R2 = C;
```

приведет к тому, что матрицы R1 и R2 будут отличаться в 2 раза, поскольку в данном случае переменная C имеет тип не матрицы, а некоторого «ленивого» шаблонного выражения, т.е. при ее определении никакого перемножения матриц в реальности не происходит. Решить эту проблему можно либо явным указанием типа переменной C (Eigen::MatrixXd), либо модификацией строки:

```
auto C = (A * B).eval();
```

смысл которой понятен из сокращенного названия операции «evaluate».

Аналогичное, на первый взгляд некорректное поведение, может наблюдаться при использовании тернарного оператора, а также при обработке выражений, в которых один и тот же объект встречается и слева, и справа от знака присваивания.

Таким образом, особенности внутреннего устройства библиотеки линейной алгебры Eigen, получить первоначальное представление о которых можно из документации (https://eigen.tuxfamily.org/dox/UserManual_UnderstandingEigen.html), необходимо учитывать при написании кода, чтобы обеспечить наиболее полное и эффективное использование ее возможностей.

4.2 Выполнение расчетов на GPU. Библиотеки CUDA

Возможности использования библиотеки Eigen при выполнении расчетов на графических картах, используя технологию Nvidia CUDA, ограничены работой с матрицами и векторами фиксированной размерности – 1...4-мерными. В этой связи использование Eigen для исполнения основной массы необходимых операций оказывается невозможным. Заменой ей может служить библиотека cuBLAS или cuSPARSE (для работы с заполненными и разреженными матрицами соответственно); обе входят в состав Nvidia Toolkit. Удобство их использования заключается в отсутствии необходимости писать код, компилируемый для

устройства компилятором `nvcc`, т.е. все используемые функции являются `host`-функциями. Также указанные библиотеки снабжены подробной документацией.

Недостаток использования `cuBLAS/cuSPARSE` с точки зрения пользователя, не владеющего «низкоуровневой» технологией работы с GPU, – ненаглядная и не интуитивно-понятная нотация при написании кода. В отличие от `Eigen`, библиотеки для работы с CUDA не предоставляют пользователю собственную «экосистему» типов и структур данных, а используются лишь для выполнения вычислительно трудоемких операций. При этом следует иметь в виду, что выполнение действий на устройстве (видеокарте) требует накладных расходов – копирования на устройство исходных данных, а затем копирования результата в обратном направлении. Говоря об операциях BLAS уровня 3 (решение линейных систем, умножение и обращение заполненных матриц), необходимо помнить, что их сложность пропорциональна кубу размерности матриц, а объем обрабатываемых данных – ее квадрату, поэтому выполнение таких операций на устройстве оправдано лишь для достаточно «больших» задач. К примеру, для матриц, имеющих размер порядка нескольких десятков тысяч, выигрыш от решения линейной системы на устройстве, в зависимости от соотношения производительностей конкретного процессора и видеокарты, может достигать нескольких десятков раз.

Исполнение на устройстве операций BLAS уровня 2 едва ли может быть оправданным, исключая случаи, когда матрица уже загружена в память устройства.

4.3 Решение задачи типа N тел

«Прямой» алгоритм решения задачи N тел реализуется исключительно просто (без разницы, для CPU или GPU) и практически идеально масштабируется. Главный его недостаток – квадратичная вычислительная сложность, что при количестве вихревых частиц порядка сотен тысяч или миллионов приводит к неприемлемым затратам времени.

Эффективное решение данной проблемы – в использовании приближенных быстрых алгоритмов, все многообразие которых можно разделить на два класса: методы, в основе которых построение иерархической пространственной структуры дерева (`treecodes`), и методы, предполагающие выполнение быстрого преобразования Фурье (FFT). К первым относятся метод Барнса – Хата, быстрый метод мультиполей (FMM) и многочисленные их вариации; именно эти методы, судя по публикациям, завоевали наибольшую популярность.

Большой интерес представляет достаточно обстоятельное методическое исследование, проведенное для данной задачи анонимным автором [26], полученные там результаты весьма наглядны и отражают основные тенденции: в частности, главным выводом является то, что наибольшую производительность обеспечивает CUDA-реализация алгоритма метода Барнса – Хата, при этом следует уделять самое пристальное внимание программной реализации с учетом специфики выполнения вычислений и организации памяти на GPU. Там же [26] приведена ссылка на репозиторий <https://github.com/drons/nbody> (свободное ПО, лицензия не указана), где представлены реализации всех описанных алгоритмов. Представленный код требует библиотек Qt, Boost, OpenCL, OpenGL, CUDA Toolkit.

К сожалению, даже тот алгоритм, что признан автором оптимальным, реализован в указанной библиотеке не самым эффективным образом (по крайней мере, если требуется обеспечивать достаточно высокую точность расчетов); это становится ясно из сопоставления указанной реализации с кодом ECL-BH автора M.Burcher, доступным на его персональной странице <http://www.cs.txstate.edu/~burtscher/research/ECL-BH/> (лицензия BSD) и детально описанным в [27]: при идентичных исходных данных и одинаковой точности время вычислений на одной и той же видеокарте отличается примерно на порядок (в 10 раз). Следует отметить, что указанный код ECL-BH не лишен недостатков: на производительных видеокартах типа Nvidia Titan V, Tesla V100, Tesla A100, имеющих порядка сотни мультипроцессоров, наблюдается «гонка данных», которая, впрочем, может быть устранена после внимательного

изучения и анализа структуры кода путем добавления в необходимых местах квалификатора `volatile`. Код ECL-BH включен в проект Lonestar (доступен из репозитория <https://github.com/chexuhao/Lonestaromp>, лицензия MIT), а также используется в ряде “in-house”-разработок, в которых приходится решать гравитационную задачу N тел. При этом тот факт, что основная характеристика каждого тела – его масса – это скалярная положительная величина, существенно используется в данном коде, и для его обобщения даже на задачи взаимодействия точечных зарядов (положительного и отрицательного знаков), и тем более на задачи взаимодействия вихревых частиц, требуется нетривиальная модификация, приводящая к некоторому снижению производительности. Сравнительную простоту модификации кода обеспечивает его весьма ясная, хотя и несколько громоздкая структура, и отсутствие зависимостей от других библиотек, не считая стандартных.

Другим представителем семейства `treecodes` является семейство быстрых методов мультиполей. По методам мультиполей в различных модификациях имеется огромное количество публикаций, среди которых отметим монографию [28] и серию работ [29, 30] международного коллектива, в состав которого входят L. Barba (США), R. Yokota (Япония), Н. Гумеров (Россия, США), посвященных большей частью именно проблемам использования быстрого метода мультиполей совместно с методом вихревых частиц. Как следует из указанных, а также других их публикаций, данными авторами разработаны соответствующие коды, допускающие в том числе проведение расчетов с использованием графических ускорителей. Именно эти коды в свободном доступе отсутствуют, однако на в репозитории <https://github.com/barbagroup/gemsfmm> присутствует код `gemsfmm` тех же авторов, в котором реализован быстрый метод мультиполей для решения плоской и пространственной задачи N тел, причем как для CPU, так и для GPU.

В основе метода Барнса – Хата, быстрого метода мультиполей и их многочисленных вариаций – иерархическое разбиение пространства и построение дерева. Для FMM требуется `quad/oct`-дерево, для алгоритма Барнса – Хата подходит также и k -d дерево. Наименее тривиальной является эффективная реализация процедуры построения дерева на GPU. Одна из реализаций имеется в вышеупомянутом коде ECL-BH, однако более эффективным, по крайней мере для реализаций на GPU, оказывается алгоритм, основанный на линейном упорядочивании расположенных на плоскости или в пространстве частиц при помощи фрактальной кривой Мортонa [31]. Его реализация в виде кода `lbvh`, опирающегося на библиотеку `thrust`, доступна из репозитория <https://github.com/ToruNiina/lbvh> (лицензия MIT). При этом отметим, что для сортировки частиц по их кодам Мортонa можно использовать более эффективную функцию поразрядной сортировки `DeviceRadixSort::SortPairs` из библиотеки CUB, входящей с определенного момента в состав CUDA Toolkit (https://nvlabs.github.io/cub/structcub_1_1_device_radix_sort.html). Эффективный алгоритм вычисления префиксных сумм описан в [32]; соответствующий код доступен в репозитории <https://github.com/TVycas/CUDA-Parallel-Prefix-Sum>.

Алгоритмы приближенного решения задачи N тел, основанные на быстром преобразовании Фурье, распространены значительно меньше; применительно к вихревым методам наиболее известна работа [16], положенная в основу упоминавшегося выше кода `vxflow`. Наиболее трудоемкая операция там – это собственно двумерное быстрое дискретное преобразование Фурье, выполняемое для двух перемножаемых матриц, и обратное преобразование, выполняемое для матрицы-результата. Для его оптимальной реализации на CPU представляется целесообразным использование обсуждавшейся выше библиотеки `Eigen` или широко известной библиотеки `FFTW` (лицензия GPL-2.0). Для GPU практически безальтернативным вариантом является использование библиотеки `cuFFT` (составная часть CUDA Toolkit).

5. Формирование вихревых петель

При решении двумерных задач, в которых рассчитывается плоское обтекание профилей или систем профилей, после решения граничного интегрального уравнения определяется величина интенсивности вихревого слоя на обтекаемых контурах, затем завихренность, содержащаяся в этом слое, «стягивается» в точечные вихри, которые «пополняют» вихревой след. Данная процедура с вычислительной точки зрения является крайне простой.

В трехмерных задачах при расчете пространственного обтекания тел реализовать аналогичный алгоритм нельзя, поскольку требуется обеспечить соленидальность представления завихренности в области течения; для этого достаточно представлять завихренность в виде замкнутых структур (петель). Чтобы определить положения вновь генерируемых на поверхности тела петель следует решить обратную задачу восстановления потенциала двойного слоя на поверхности по известному значению его градиента (интенсивность вихревого слоя есть развернутый на $\pi/2$ вокруг внешней нормали поверхностный градиент потенциала двойного слоя). С учетом того, что задача решается на триангулированной поверхности, число ячеек которой практически точно в два раза больше числа вершин (как следует из формулы Эйлера, $F = 2V - 4$), и можно считать, что в центре каждой ячейки известно значение поверхностного градиента функции, задача определения ее узловых значений сводится к поиску псевдорешения системы с разреженной матрицей, что, в свою очередь, эквивалентно процедуре метода наименьших квадратов, рис. 2.

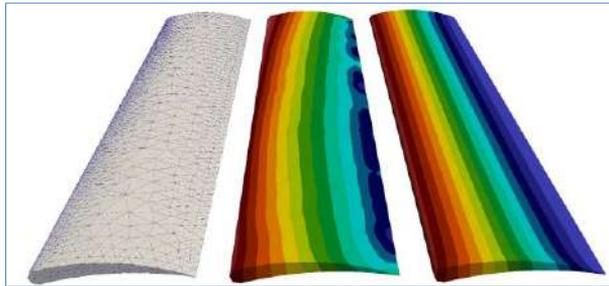


Рис. 2. Поверхностная сетка на модели крыла (слева) и плотность потенциала двойного слоя, восстановленная путем решения гиперсингулярного ГИУ (в центре) и при помощи процедуры метода наименьших квадратов после определения интенсивности вихревого слоя (справа)

Fig. 2. The surface mesh on the wing model (left) and the double layer potential density reconstructed by solving the hypersingular integral equation (center) and using the least squares procedure that follows the reconstruction of the vortex sheet intensity (right)

Для реализации метода наименьших квадратов могут быть использованы уже обсуждавшиеся библиотеки Eigen (для CPU) и cuSPARSE (для GPU).

После восстановления потенциала по его линиям уровня задаются положения вихревых петель. Каждая петля задается узлами, при этом движение петли определяется движением ее узлов. При этом после выполнения шага расчета отдельные узлы петли сближаются, другие удаляются друг от друга. Неравномерность дискретизации петель ведет к их неестественному «выгибанию» и закладывает источник вычислительной неустойчивости. Чтобы этого избежать, на каждом шаге расчета нужно производить редискретизацию вихревых петель, т.е. заново расставлять узлы вдоль петли. При этом до этой процедуры целесообразно исключать из рассмотрения или объединять сблизившиеся узлы (рис. 3).

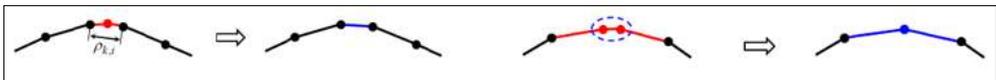


Рис. 3. Схема исключения и объединения сблизившихся узлов вихревой петли
Fig. 3. Scheme of exclusion and merging of closely placed nodes of the vortex loop

Собственно редискретизацию нужно производить так, чтобы насколько это возможно сохранять длину петли. Это можно обеспечить, если по текущему положению узлов восстановить кривую в виде сплайна, а затем разбить полученный сплайн на дуги равной длины. Хотя построение и работа со сплайнами – довольно простая операция, для этих целей удобно использовать существующие библиотеки, например, библиотеку spline, доступную из репозитория <https://github.com/ttk592/spline> и опубликованную под лицензией GPL-2.0. Отметим, что в указанной библиотеке нет возможности построения циклического сплайна (с периодическими граничными условиями, рис. 4, что позволяет избежать появления угловой точки), однако такую возможность можно сравнительно легко добавить. Библиотека исключительно легковесна (один заголовочный файл), зависимости отсутствуют.

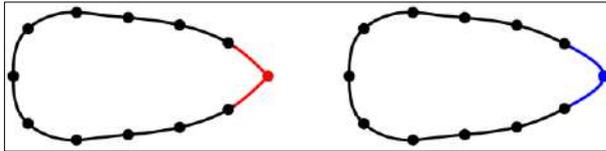


Рис. 4. Форма вихревой петли, восстановленная классическим (слева) и циклическим (справа) сплайном
Fig. 4. Vortex loop smooth shape reconstruction with classical (left) and periodic (right) splines

6. Алгоритмы вычислительной геометрии

В трехмерном случае возникает необходимость решения ряда задач вычислительной геометрии, чаще всего простых и понятных по постановке, но нетривиальных в смысле использования эффективных алгоритмов. Одна из них возникает, когда фрагменты вышеупомянутых вихревых петель располагаются вблизи обтекаемой поверхности и после их перемещения на шаге расчета вследствие неизбежных погрешностей моделирования оказываются внутри тела. В этом случае представляется наиболее естественным отыскать кратчайший путь на поверхности между точками пересечения петли с ней и «проложить» вихревую петлю по этому пути. Здесь же с очевидностью возникает и вторая задача: установить факт пересечения вихревых петель с триангулированной поверхностью. Отметим, что сходная задача актуальна и в двумерном случае: требуется идентифицировать попадания точечных вихревых частиц внутрь контура обтекаемого профиля.

Эти и многие другие задачи могут быть эффективно решены с использованием библиотеки алгоритмов вычислительной геометрии CGAL, написанной на C++ и обеспечивающей удобный доступ к эффективным реализациям множества геометрических и смежных алгоритмов. Библиотека CGAL свободно доступна вместе с исходными кодами по лицензии LGPL-v3; она развивается с 1996 года как крупная совместная разработка Университета г. Утрехт, ETH (г. Цюрих), Свободного университета Берлина, Университета Мартина Лютера (г. Галле), Университета Иоганна Кеплера (г. Линц), Тель-Авивского университета, а также Института Макса Планка и INRIA. Следует отметить, что библиотека весьма громоздка, основана на Boost, Qt и Eigen; требует отдельной компиляции. Фактически она состоит из множества тематически объединенных пакетов, каждый из которых отвечает за структуры данных и базовые алгебраические, геометрические, логические операции или алгоритмы, включающие построение, перестроение и оптимизацию поверхностных и объемных сеток, реконструкцию поверхностей, а также решение отдельных задач.

В контексте обсуждаемых вопросов представляют интерес алгоритмы поиска кратчайшего пути на триангулированной поверхности. Зачастую сложность данной задачи недооценивают, ошибочно полагая, что для ее решения могут быть применены многие «стандартные» хорошо известные алгоритмы на графах: алгоритмы Дейкстры, Флойда – Уоршелла, Беллмана – Форда и др. Это верно лишь отчасти: только в случае, если в качестве решения допустим путь на поверхности, проходящий по ее ребрам. Такой путь обычно оказывается весьма «угловатым», что может стать причиной развития вычислительной

неустойчивости в вихревых методах, поэтому требуется его некоторым образом сгладить. Ясно, что никакой путь, лежащий на триангулированной поверхности (если только она не является плоскостью) не может быть гладким в математическом смысле, поэтому в данном можно предложить следующее нестрогое, но интуитивно понятное определение «достаточно гладкого» пути: если сделать развертку для тех граней, по которым проходит путь, то углы между соответствующей ему ломаной не должны превышать (по крайней мере, по порядку величины) двугранных углы между этими же гранями на исходной поверхности. Наиболее гладким в указанном смысле будет кратчайший путь, поиск которого может быть осуществлен с использованием модулей *The Heat Method* и *Triangulated Surface Mesh Shortest Paths* библиотеки *CGAL*. Удобство и простота использования библиотеки в некоторой мере нивелируются не слишком высокой производительностью: во всех алгоритмах такого типа наиболее эффективно могут быть решены задачи в постановке «найти путь от фиксированной точки на поверхности до множества других точек», причем увеличение, даже значительное, числа «финальных» точек мало сказывается на росте вычислительной сложности. В данном же случае требуется решать задачу для множества различных пар точек «начало – конец маршрута» (рис. 5).

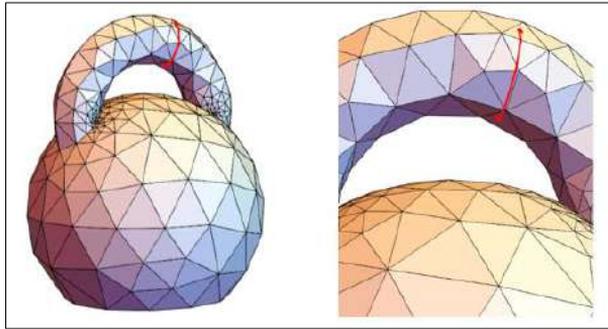


Рис. 5. Пример поиска кратчайшего пути между двумя точками на поверхности
Fig. 5. Example of the shortest path from one fixed point to another on the surface

Необходимо отметить, что на практике можно ограничиться поиском пути, достаточно гладкого в указанном выше смысле, но не обязательно самого короткого из возможных. Алгоритм такого типа [33], особенно если он обеспечивает хорошую экономию времени вычислений, представлялся бы весьма полезным, при этом найденный путь может быть как весьма близким к кратчайшему, так и значительно отличающимся от него (рис. 6).

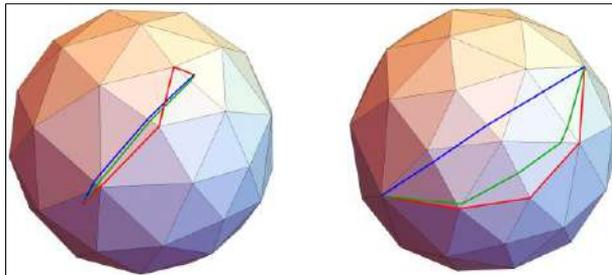


Рис. 6. Кратчайший путь между точками на поверхности (синяя линия), путь найденный по алгоритму Дейкстры (красная линия) и результат «локальной оптимизации» пути [33] (зеленая линия) для двух пар случайным образом выбранных точек на триангулированной сфере

Fig. 6. The shortest path between points on the surface (blue line), the path found by Dijkstra's algorithm (red line) and the result of "local optimization" of the path (green line) for two pairs of randomly selected points on a triangulated sphere

Если говорить о задаче идентификации точек, проникших внутрь триангулированной поверхности или многоугольного контура, которую обычно можно переформулировать как

задачу поиска точек пересечения системы отрезков с множеством треугольников или второй системой отрезков, то она должна быть решена максимально точно, ошибки в ее решении, как показывает опыт, ведут к развитию неустойчивости в алгоритмах вихревых методов. Для решения этой задачи можно использовать такие модули библиотеки CGAL как 2D Intersection of Curves и 3D Fast Intersection and Distance Computation, а также Intersecting Sequences of dD Iso-oriented Boxes. В основе реализованных там алгоритмов – построение иерархического AABB-дерева и его обход. Временные затраты на решение задачи могут быть снижены за счет оптимальной организации процесса, правильного выбора того множества геометрических объектов, по которым это дерево строится: треугольные панели или пересекающие их отрезки, и др. Пример вихревого следа, моделируемого системой вихревых петель за крылом конечного размаха, представленным триангулированной поверхностью, показан на рис. 7. В процессе моделирования на каждом шаге расчета наблюдаются десятки случаев проникновения фрагментов петель внутрь поверхности, которые должны быть идентифицированы, и произведено перестроение петель по кратчайшим путям на поверхности.

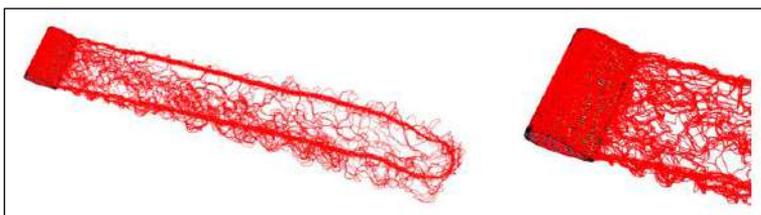


Рис. 7. Формирование вихревого следа за крылом конечного размаха
Fig. 7. Vortex wake formation after the wing of finite span

Отметим, что для плоских задач данный алгоритм поиска пересечений отрезков, вообще говоря, уступает по эффективности т.н. алгоритму Балабана [34], позволяющему отыскивать точки пересечения множества отрезков на плоскости, который является неулучшаемым в смысле вычислительной сложности. Реализацию этого алгоритма на C++, выполненную автором, можно найти в репозитории <https://github.com/ivvaan/balaban-segments-intersections>. Тем не менее, особенности постановки – наличие двух систем отрезков и необходимость искать пересечения отрезков первой группы с отрезками второй группы – делают задачу несколько отличающейся от той, для которой алгоритм Балабана является оптимальным.

Отметим, что имеющиеся в CGAL алгоритмы построения AABB и k -d деревьев и их обхода представляет интерес и в контексте выполнения иных процедур вихревых методов: в частности, редискретизации вихревого следа, когда требуется отыскивать близко расположенные вихревые частицы с целью их возможного объединения (в 2D задачах) или выполнения перезамыкания вихревых колец (в 3D задачах). Для этих же целей, а также при вычислении диффузионной скорости вихревых частиц может быть полезен алгоритм поиска ближайших соседей. Перечисленные алгоритмы объединены в CGAL в группу Spatial Searching and Sorting Algorithms.

7. Моделирование динамики обтекаемых тел

При решении сопряженных задач гидроупругости, когда обтекаемые поверхности являются подвижными и/или деформируемыми, вычислительная сложность процедуры расчета движения или деформации тел, как правило, существенно ниже таковой применительно к решению гидродинамической подзадачи. Тем не менее, сама механическая система может быть нетривиальной, и для моделирования ее динамики под действием гидродинамических нагрузок, в зависимости от конкретной постановки задачи, можно с успехом использовать ряд известных библиотек.

К примеру, если обтекаемые тела или профили являются жесткими (недеформируемыми) и лишь перемещаются при наложенных на них связях, может быть эффективна свободная библиотека Project Chrono, разрабатываемая совместными усилиями Университета Висконсина (Мадисон, США) и Пармского университета с 2008 года. Исходный код доступен в репозитории <https://github.com/projectchrono/chrono> под лицензией BSD. Библиотека «тяжеловесна», требует отдельной компиляции со многими зависимостями (определяются конкретным перечнем компилируемых модулей), поддерживает расчеты на CPU и GPU. Интерфейс библиотеки сравнительно несложный, изучение большого количества прилагающихся примеров позволяет быстро освоить базовые принципы работы с ней.

Если же под действием гидродинамических (или иных) нагрузок обтекаемое тело деформируется, то наиболее перспективным видится использование библиотек конечноэлементного анализа, из числа которых можно выделить Deal.II – ведущую с 2000 года совместную разработку университетов Колорадо, Клемсона, Техаса (все – США), Гейдельбергского университета (ФРГ), а также национальной лаборатории Oak Ridge. Исходный код проекта размещен в репозитории <https://github.com/dealii/dealii> и доступен под лицензией LGPL-v2. Библиотека также весьма «тяжеловесная», имеет много зависимостей и требует отдельной компиляции. Из ее особенностей – большой набор учебных примеров, снабженных чрезвычайно подробными объяснениями и комментариями. Библиотека поддерживает распараллеливание вычислений посредством MPI и выполнение расчетов на GPU.

8. Заключение

Несмотря на существенные трудности при решении задач моделирования гидро- и аэродинамических процессов, протекающих в нестационарных условиях, использование вихревых методов вычислительной гидродинамики позволяет добиться удовлетворительных результатов, отвечающих требованиям инженерной практики, рассматривая при этом достаточно простые модели среды и затрачивая сравнительно скромные вычислительные ресурсы.

В работе применительно к реализациям алгоритмов вихревых методов дана оценка целесообразности и перспектив использования сторонних библиотек линейной алгебры, вычислительной геометрии, параллельных вычислений и т.п. Их использование помогает упростить и оптимизировать алгоритм, однако в ряде случаев может отрицательно влиять на производительность. Иными словами, к использованию каждой из «стандартных» (библиотечных) функций следует подходить достаточно ответственно, тщательно анализируя те постановки задач, для которых соответствующие алгоритмы и библиотеки оптимизированы, и сопоставляя их с контекстом, возникающим в вихревых методах.

Исходя из вышеизложенного, становится ясным одно из важных направлений развития упомянутых в работе и будущих реализаций вихревых методов. Обоснованное использование возможностей открытых библиотек и их сочетаний может позволить не только оптимизировать имеющиеся реализации, в том числе за счет использования возможностей современных ЭВМ различных архитектур, но и расширить их возможности.

Табл. 1. Расшифровки использованных в тексте аббревиатур

Table 1. Descriptions of abbreviations used in the text

Аббревиатура	Расшифровка	Перевод на русский язык с пояснением
ГИУ	Граничное Интегральное Уравнение	–
ПО	Программное Обеспечение	–

ЭВМ	Электронная Вычислительная Машина	–
AABB-дерево	(Axis Aligned Bounding Box)-дерево	Пространственное дерево, ячейки которого – параллелепипеды с гранями, параллельными координатным плоскостям
BiCGStab	BiConjugate Gradient Stabilized method	Метод бисопряженных градиентов со стабилизацией (итерационный метод решения линейных систем)
BLAS	Basic Linear Algebra Subroutines	Базовые подпрограммы линейной алгебры (стандарт де-факто для интерфейса библиотек, выполняющих основные операции линейной алгебры)
BSD	Berkeley Software Distribution	Программная лицензия университета Беркли (для распространения ПО с открыты кодом)
CGAL	Computational Geometry Algorithms Library	Библиотека алгоритмов вычислительной геометрии
CPU	Central Processing Unit	Центральный процессор
CUDA	Compute Unified Device Architecture	Архитектура унифицированного вычислительного устройства (программно-аппаратная архитектура параллельных вычислений с их выполнением на графических процессорах фирмы Nvidia)
Deal.II	Differential equations analysis library (2-nd generation)	Библиотека конечно-элементного анализа (библиотека для численного решения дифференциальных уравнений в частных производных методом конечных элементов)
FFT	Fast Fourier Transform	Быстрое преобразование Фурье
FFTW	Fastest Fourier Transform in the West	Самое быстрое преобразование Фурье на Западе (одна из наиболее известных и эффективных бесплатных программных реализаций быстрого преобразования Фурье)
FMM	Fast Multipole Method	Быстрый метод мультиполей
GMRES	Generalized Minimal RESidual method	Обобщенный метод минимальных невязок (итерационный метод решения линейных систем)
GPL	(GNU) General Public License	Универсальная общественная лицензия GNU (лицензия на свободное программное обеспечение, созданная в рамках проекта GNU)
GPU	Graphics Processing Unit	Графический процессор (графический ускоритель, видеокарта)
Intel MKL	Intel Math Kernel Library	Библиотека математического ядра Intel (библиотека оптимизированных математических алгоритмов, сопровождающая компиляторы Intel)
Intel TBB	Intel Threading Building Blocks	Кроссплатформенная библиотека шаблонов C++, разработанная Intel для параллельного программирования
LAPACK	Linear Algebra PACKage	Библиотека методов решения основных задач линейной алгебры (с открытым исходным кодом)
LGPL	(GNU) Lesser General Public License	Ослабленная универсальная общественная лицензия GNU (лицензия на свободное программное обеспечение, созданная в рамках проекта GNU)
OpenCL	Open Computing Language	Фреймворк для параллельного программирования с выполнением вычислений на центральных и графических процессорах, а также FPGA

OpenGL	Open Graphics Library	Фреймворк для разработки приложений, использующих двумерную и трёхмерную компьютерную графику
OpenMP	Open Multi-Processing	Стандарт для распараллеливания программ на системах с общей памятью
MIT (license)	Massachusetts Institute of Technology (license)	Лицензия Массачусетского технологического университета (разрешительная лицензия на открытое и свободное программное обеспечение)
MPI	Message Passing Interface	Интерфейс передачи сообщений (стандарт для распараллеливания программ на системах с распределенной памятью)
MPL	Mozilla Public License	Общественная лицензия Mozilla Foundation (лицензия на открытое и свободное программное обеспечение)
VTK	Visualization Toolkit	Инструментарий визуализации (открытая кроссплатформенная библиотека для геометрического моделирования, обработки изображений и визуализации)

Список литературы / References

- [1] Leonard A. Vortex methods for flow simulation. *Journal of Computational Physics*, vol. 37, issue 3, 1980, pp. 289-335.
- [2] Sarpkaya T. Computational methods with vortices – The 1988 Freeman Scholar Lecture. *Journal of Fluids Engineering*, vol. 111, issue 1, 1989, pp. 5–52.
- [3] Mimeau C., Mortazavi I. A review of vortex methods and their applications: from creation to recent advances. *Fluids*, vol. 6, issue 2, 2021, article no. 68, 49 p.
- [4] Dynnikova G.Ya. An analog of the Bernoulli and Cauchy – Lagrange integrals for a time-dependent vortex flow of an ideal incompressible fluid. *Fluid Dynamics*, vol. 35, issue 1, 2000, pp. 24-32.
- [5] Андронов П.Р., Гувернюк С.В., Дынникова Г.Я. Вихревые методы расчета нестационарных гидродинамических нагрузок. М., Изд-во МГУ, 2006 г. 184 стр. / Andronov P.R., Guvernuyuk S.V., Dynnikova G.Y. Vortex methods for non-stationary hydrodynamic loads estimation. Moscow, Moscow State University, 2006, 184 p. (in Russian).
- [6] Dynnikova G.Ya, Andronov P.R. Expressions of force and moment exerted on a body in a viscous flow via the flux of vorticity generated on its surface. *European Journal of Mechanics - B/Fluids*, vol. 72, 2018, pp. 293–300.
- [7] Dynnikova G.Ya. General expression of aerodynamic force under different boundary conditions (slip, partial slip, no-slip). *Physics of Fluids*, vol. 33, issue 6, 2021, article no. 063104.
- [8] Дынникова Г.Я. Лагранжев подход к решению нестационарных уравнений Навье – Стокса. Доклады Академии наук, том 399, вып. 1, 2004 г., стр. 42–46. / Dynnikova G.Ya. The Lagrangian approach to solving the time-dependent Navier-Stokes equations. *Doklady Physics*, vol. 49, issue 11, 2004, pp. 648–652.
- [9] Гувернюк С.В., Дынникова Г.Я. Моделирование обтекания колеблющегося профиля методом вязких вихревых доменов. Известия РАН. Механика жидкости и газа, вып. 1, 2007, стр. 3–14 / Guvernuyuk S.V., Dynnikova G.Ya. Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. *Fluid Dynamics*, vol. 42, issue 1, 2007, pp. 1-11.
- [10] Дынникова Г.Я. Вихревые методы исследования нестационарных течений вязкой несжимаемой жидкости. Диссертация на соискание ученой степени доктора физико-математических наук. М., 2011 г., 269 стр. / Dynnikova G.Ya. Vortex methods for studying unsteady flows of a viscous incompressible fluid. Thesis for the degree of Doctor of Physical and Mathematical Sciences. Moscow, 2011. 269 p. (in Russian).
- [11] Stock M.J., Gharakhani A. Open-source accelerated vortex particle methods for unsteady flow simulation. In Proc. of the ASME 2020 Fluids Engineering Division Summer Meeting, 2020, paper no: FEDSM2020-20486, 10 p.
- [12] Alvarez E.J., Ning A. High-fidelity modeling of multirotor aerodynamic interactions for aircraft design. *AIAA Journal*, vol. 58, issue 10, 2020, pp. 4385-4400.
- [13] Гирча А.И. Реализация вихревых методов и алгоритмов моделирования процессов нестационарной гидродинамики на основе эффективного комплекса программ. Диссертация на соискание ученой

- степени кандидата физико-математических наук. М., 2009 г., 172 с. / Gircha A.I. Implementation of vortex methods and algorithms for simulation the processes of unsteady hydrodynamics based on an effective software package. Thesis for the degree of candidate of physical and mathematical sciences, Moscow, 2009. 172 p. (in Russian).
- [14] Григоренко Д.А. Комплекс программ для реализации семейства вихревых методов и его применение. Диссертация на соискание ученой степени кандидата физико-математических наук. М., 2008 г., 149 с. / Grigorenko D.A. A software package for the implementation of a family of vortex methods and its application. Thesis for the degree of candidate of physical and mathematical sciences, Moscow, 2008. 149 p. (in Russian).
- [15] Chorin A.J. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*, vol. 57, issue 4, 1973, pp. 785-796..
- [16] Morgenthal G., Walther J.H. An immersed interface method for the Vortex-In-Cell algorithm. *Computers and Structures*, vol. 85, issues 11-14, 2007, pp. 712-726.
- [17] Morgenthal G. Aerodynamic analysis of structures using high resolution vortex particle methods. Ph.D. thesis. University of Cambridge, 2002. 209 p.
- [18] Giannopoulou O., Colagrossi A. et al. Chorin's approaches revisited: Vortex particle method vs finite volume method. *Engineering Analysis with Boundary Elements*, vol. 106, 2019, pp. 371-388.
- [19] Kamemoto K., Tsutahara M., eds. *Vortex methods: selected papers of the First International Conference on Vortex Methods*. Wspc, 2000. 220 p.
- [20] Marchevsky I., Sokol K., Ryatina E., Izmailova Y. The VM2D Open Source Code for Two-Dimensional Incompressible Flow Simulation by Using Fully Lagrangian Vortex Particle Methods. *Axioms*, 12(3):248, 2023. DOI: 10.3390/axioms12030248.
- [21] Kuzmina K., Marchevsky I. et al. On the scope of Lagrangian vortex methods for two-dimensional flow simulations and the POD technique application for data storing and analyzing. *Entropy*, vol. 23, issue 1, article no. 118, 2021, 38 p.
- [22] Марчевский И.К., Сокол К.С., Измайлова Ю.А. Т-схемы для математического моделирования генерации завихренности на гладких профилях в вихревых методах. Вестник МГТУ им. Н.Э. Баумана. Естественные науки, вып. 6, 2022 г., стр. 33–59 / Marchevskii I.K., Sokol K.S., Izmailova Yu.A. T-schemes for mathematical modelling of vorticity generation on smooths airfoils in vortex particle methods. *Herald of the Bauman Moscow State Technical University, Series Natural Sciences*, issue. 6, 2022, pp. 33–59 (in Russian).
- [23] Dergachev S.A., Marchevsky I.K., Shcheglov G.A. Flow simulation around 3D bodies by using Lagrangian vortex loops method with boundary condition satisfaction with respect to tangential velocity components // *Aerospace Science and Technology*. 2019. Vol. 94. Art. 105374. DOI: 10.1016/j.ast.2019.105374.
- [24] Марчевский И.К., Серафимова С.Р. Аналитическое и полуаналитическое вычисление интегралов от логарифмического и ньютоновского потенциала и их градиентов по прямолинейным отрезкам и треугольным панелям. Вычислительные методы и программирование, том. 23, вып. 2. 2022 г., 137-152. / Marchevsky I.K., Serafimova S.R. Analytic and semi-analytic integration of logarithmic and Newtonian potentials and their gradients over line segments and rectilinear panels. *Numerical Methods and Programming*, vol. 23, issue 2. 2022, pp. 137-152 (in Russian).
- [25] Reid M.T.H., White J.K., Johnson S.G. Generalized Taylor–Duffy Method for Efficient Evaluation of Galerkin Integrals in Boundary-Element Method Computations. *IEEE Transactions on Antennas and Propagation*, vol. 63, issue 1, 2015, pp. 195-209.
- [26] Задача N тел или как взорвать галактику не выходя из кухни / N-body problem or how to blow up a galaxy without leaving the kitchen. Available at: <https://habr.com/ru/articles/437014/>, accessed 9.05.2023 (in Russian).
- [27] Burtscher M., Pingali K. Chapter 6 – An efficient CUDA implementation of the tree-based Barnes Hut n-body algorithm. In *GPU Computing Gems Emerald Edition; Applications of GPU Computing Series*. Morgan Kaufmann Publishers, 2011. pp. 75-92.
- [28] Gumerov N.A., Duraiswami R. *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*. Elsevier Science, 2005. 426 p.
- [29] Hu Q., Gumerov N.A. et al. Scalable fast multipole accelerated vortex methods. In *Proc. of the International Parallel and Distributed Processing Symposium*, 2014, pp. 966-975.
- [30] Yokota R., Barba L.A. Comparing the treecode with FMM on GPUs for vortex particle simulations of a leapfrogging vortex ring. *Computers and Fluids*, vol. 45, issue 1, 2011, pp. 155-161.

- [31] Karras T. Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In Proc. of the Fourth ACM SIGGRAPH Eurographics conference on High-Performance Graphics, 2012, pp. 33–37.
- [32] Harris M., Sengupta S., Owens J.D. Parallel Prefix Sum (Scan) with CUDA. Chapter 39. In *Gpu gems 3*, ed. by Hubert Nguyen. Addison-Wesley Professional, 2007. Available at: <https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-39-parallel-prefix-sum-scan-cuda>, accessed 9.05.2023.
- [33] Gumirova A.I., Marchevsky I.K., Safronov Yu. The algorithm of the path length optimization on the polyhedron surface. In Proc. of the 49th International Summer School-Conference on Advanced Problems in Mechanics, 2021, Springer, 2023. In press.
- [34] Балабан И.Ю. Алгоритмы поиска пересечений множества отрезков. Препринты ИПМ им. М.В.Келдыша, 1994 г., no. 45. 25 стр. / Balaban I.Yu. Algorithms for finding intersections of a set of segments. KIAM preprints, 1994, no. 45. 25 p. (in Russian).

Информация об авторах / Information about authors

Илья Константинович МАРЧЕВСКИЙ – доктор физико-математических наук, доцент, профессор кафедры. «Прикладная математика» МГТУ им. Н.Э. Баумана, старший научный сотрудник Института системного программирования им. В.П. Иванникова РАН. Сфера научных интересов: вычислительная гидродинамика, вихревые методы, теория устойчивости, численные методы, высокопроизводительные вычисления, элементарная математика.

Ilya Konstantinovich MARCHEVSKY – Doctor of Science (Phys.-Math.), Associate professor, Professor of Applied Mathematics department, Bauman Moscow State Technical University; Senior Researcher in Ivannikov Institute for System Programming of the RAS. Research interests: computational fluid dynamics, vortex particle methods, theory of stability, numerical methods, high performance computing, elementary mathematics.

Юлия Андреевна ИЗМАЙЛОВА – студентка магистратуры кафедры «Прикладная математика». Сфера научных интересов: вихревые методы, высокопроизводительные вычисления.

Yulia Andreevna IZMAILOVA – master's student of Applied Mathematics department. Research interests: vortex particle methods, high performance computing.

Мария Александровна ЕРОФЕЕВА – студентка кафедры «Прикладная математика». Сфера научных интересов: вихревые методы, теория устойчивости, интегральные уравнения.

Maria Aleksandrovna EROFEEVA – student of Applied Mathematics department. Research interests: vortex particle methods, theory of stability, and integral equations.

Дарья Юрьевна КОБЗАРЬ – студентка кафедры «Прикладная математика». Сфера научных интересов: вихревые методы, дифференциальная геометрия, теоретическая механика.

Daria Yurievna KOBZAR – student of the Applied Mathematics department. Research interests: vortex particle methods, differential geometry, theoretical mechanics.

DOI: 10.15514/ISPRAS-2023-35(2)-14



О проблемах при моделировании плоских течений вязкой жидкости при повышенных значениях числа Рейнольдса вихревыми методами в программном комплексе VM2D

*И.А. Коробова, ORCID: 0000-0002-8701-723X <i-soldatova@bk.ru>
Е.П. Рятина, ORCID: 0000-0002-7533-3796 <evgeniya.ryatina@yandex.ru>
А.А. Хорошева, ORCID: 0009-0009-0876-8667 <khorohevaann@gmail.com>*

*Московский государственный технический университет имени Н.Э. Баумана,
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1*

Аннотация. Вихревые методы вычислительной гидродинамики – эффективный в инженерной практике метод определения гидродинамических нагрузок, действующих на помещенные в поток тела. Их использование позволяет производить решение задач гидроупругости в сопряженной постановке со сравнительно малыми затратами вычислительных ресурсов. Во многих приложениях рассматривается поперечное обтекание элементов конструкций, имеющих значительное удлинение, что позволяет с допустимой точностью использовать метод плоских сечений, переходя к рассмотрению плоских задач. Современные модификации вихревых методов позволяют моделировать течения вязкой несжимаемой среды. На основе метода вязких вихревых доменов в 2017-2022 гг. в МГТУ им. Н.Э. Баумана и ИСП им. В.П. Иванникова РАН создан программный комплекс VM2D. Данный код позволяет с достаточной точностью рассчитывать обтекание профилей при малых значениях числа Рейнольдса, тогда как для повышенных чисел Рейнольдса верные результаты наблюдаются лишь для профилей с острыми кромками и угловыми точками, и только на режимах, когда наиболее интенсивный отрыв происходит именно с указанных точек. Причина погрешности результатов для других режимов видится в некорректном моделировании отрыва с гладкой поверхности профиля при высоких числах Рейнольдса, что, в свою очередь, является следствием неправильного моделирования эволюции завихренности в окрестности точек (зон) отрыва. В работе приведены некоторые результаты моделирования обтекания различных профилей при различных значениях числа Рейнольдса и выдвинута гипотеза, объясняющая причину рассогласования результатов расчетов с данными экспериментов. Показано, что спектр кинетической энергии турбулентности соответствует «двумерной турбулентности».

Ключевые слова: плоское течение, вихревые методы, аэродинамические характеристики, спектр кинетической энергии турбулентности

Для цитирования: Коробова И.А., Рятина Е.П., Хорошева А.А. О моделировании плоских течений вязкой жидкости вихревыми методами в программном комплексе VM2D. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 201-214. DOI: 10.15514/ISPRAS-2023-35(2)-14

On Problems in 2D Viscous Flows Simulation at Higher Values of the Reynolds Numbers by Vortex Methods Using the VM2D Code

I.A. Korobova, ORCID: 0000-0002-8701-723X <*i.soldatova@bk.ru*>
E.P. Ryatina, ORCID: 0000-0002-7533-3796 <*evgeniya.ryatina@yandex.ru*>
A.A. Khorosheva, ORCID: 0009-0009-0876-8667 <*khoroshevaann@gmail.com*>

*Bauman Moscow State Technical University,
5/1, ul. Baumanskaya 2-ya, Moscow 105005, Russia*

Abstract. Vortex methods of computational fluid dynamics are an efficient tool in engineering practice for estimating hydrodynamic loads acting on bodies placed in a flow. Their usage allows for solving of coupled hydroelastic problems with relatively small computational cost. In many applications, the cross flow around structural elements with large elongation is considered, that allows one to use the flat cross-sections method providing the acceptable accuracy. Thus, flat flows simulation around airfoils is required. Modern modifications of vortex particle methods make it possible to simulate flows of a viscous incompressible medium. Based on the method of viscous vortex domains in 2017-2022 the VM2D code have been developed in Bauman University and Ivannikov Institute for System Programming. This code allows for flow simulating around airfoils with acceptable accuracy at low Reynolds numbers, while for higher Reynolds numbers, correct results are observed only for airfoils with sharp edges and corner points, and only in regimes where the most intensive flow separation takes place at these points. The reason for the error in the results for other regimes is seen in incorrect modeling of the flow separation on smooth airfoil surface line at high Reynolds numbers, which, in turn, is a consequence of incorrect modeling of vorticity evolution in the vicinity of separation points (zones). Some results of flow simulations around different airfoils at different values of the Reynolds number are presented and a hypothesis explaining the reason for the discrepancy between numerical results and experimental data is proposed. It is shown that the kinetic energy spectrum of turbulence corresponds to “two-dimensional turbulence”.

Keywords: 2D flow; vortex methods; aerodynamic characteristics; spectrum of kinetic energy of turbulence

For citation: Korobova I.A., Ryatina E.P., Khorosheva A.A. On Problems in 2D Viscous Flows Simulation at Higher Values of the Reynolds Numbers by Vortex Methods Using the VM2D Code. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 201-214 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-14

1. Введение

Одной из наиболее важных в аэрогидродинамике является задача о моделировании взаимодействия элементов различных конструкций с потоком. В инженерных приложениях течение среды как таковое обычно самостоятельного интереса не представляет, а требуется рассчитать действующие на обтекаемое тело нагрузки, величина и характер действия которых определяют поведение подвижной или деформируемой конструкции. Подобные задачи возникают при расчете элементов строительных конструкций, как правило протяженных, большепролетных мостов, разнообразных тросовых конструкций, проводов воздушных линий электропередачи, подводных трубопроводов и шлангов, элементов конструкций летательных аппаратов, трубок теплообменников энергетических установок и т.п.

В настоящее время существует множество различных подходов к решению задач вычислительной гидродинамики. Наиболее распространенными являются методы, относящиеся к классу сеточных, такие как метод конечных разностей, метод контрольных объемов, метод конечных элементов и всевозможные их модификации. Стоит отметить, что в рамках использования для моделирования таких методов имеется возможность учитывать, и весьма точно, действие множества факторов, характерных для исследуемых физических процессов, что делает их область применения весьма обширной. При этом данные методы являются вычислительно трудоемкими, особенно если речь идет о моделировании нестационарного обтекания подвижных/деформируемых тел.

Другой класс – лагранжевы бессеточные методы – включает в себя вихревые методы [1–3], область применения которых ограничена несжимаемыми течениями, но во многих технических приложениях это упрощение представляется вполне допустимым. В вихревых методах в качестве первичной расчетной величины рассматривают завихренность; по известному полю завихренности можно восстановить поле скоростей в любой точке среды, используя обобщенный закон Био – Савара [4], и поле давления, пользуясь аналогами интегралов Бернулли и Коши – Лагранжа [5]. Генерация завихренности происходит на поверхности обтекаемых тел, обеспечивая тем самым выполнение граничного условия прилипания. Следует отметить, что в вихревых методах при расчете внешнего обтекания нет необходимости искусственно ограничивать область течения, т.к. граничное условие затухания возмущений на бесконечности выполняется автоматически и точно.

На сегодняшний день вихревые методы активно развиваются и существуют модификации, которые позволяют рассматривать трехмерные и двумерные течения. Соответствующие алгоритмы являются несоизмеримыми по сложности, однако при решении многих практических задач тела, взаимодействующие с потоком, имеют значительное удлинение, и поэтому вместо расчета пространственного обтекания можно рассмотреть одну или несколько задач о плоском обтекании отдельных сечений (метод плоских сечений [6]). Таким образом, алгоритмы расчета плоских задач по моделированию обтекания профилей до сих пор остаются актуальными; главным их «конкурентным преимуществом» является низкая вычислительная сложность, а следовательно – возможность сравнительно быстрого выполнения расчетов.

Тем не менее, в силу специфических особенностей вихревых методов на сегодня существует лишь небольшое количество программных реализаций вихревых методов, свободно доступных расчетчикам (в идеале – с открытым исходным кодом). Одним из них является программный комплекс VM2D [7], свободно доступный из репозитория github по ссылке <https://github.com/vortexmethods/VM2D>, в основе которого лежит метод вязких вихревых доменов [2, 8] и некоторые авторские модификации, в основном связанные с использованием T -схем повышенной точности [9] для решения граничных интегральных уравнений, а также некоторых других алгоритмов. Код VM2D можно использовать для моделирования плоских течений; он позволяет решать широкий класс задач, в том числе в сопряженной постановке, с использованием возможностей современных многопроцессорных ЭВМ различных архитектур (возможны расчеты на CPU и GPU). Целью данной работы является рассмотрение области применимости существующих алгоритмов вихревых методов на основе опыта использования программного комплекса VM2D и определения некоторых направлений будущего развития указанного кода в частности и вихревых методов моделирования плоских течений в целом.

2. Численное моделирование вязких течений в программном комплексе VM2D

Не ставя перед собой задачу полно и всесторонне рассмотреть классы и постановки задач, которые могут быть решены с использованием кода VM2D (получить представление об этом можно из [7, 10]), обратимся к вопросу об учете влияния вязкости при моделировании плоских течений.

В исторической перспективе современные версии вихревых методов для решения двумерных задач восходят к методу дискретных вихрей [11, 12] (в зарубежных публикациях используются как термин «discrete vortex method» [13], так и «lumped vortex method» [14]), который, появившись еще в 1950-х годах, по-видимому стал одним из первых методов вычислительной гидродинамики. В основе указанного метода – численное моделирование течения идеальной (невязкой) жидкости, описываемого уравнениями Эйлера; в такой постановке можно моделировать потенциальные или, по крайней мере, безвихревые течения.

Стоит отметить, что возможно воспроизведение и намного более сложных нестационарных отрывных режимов обтекания профилей – это особенно актуально для тонких профилей (пластинок), крыловых профилей и, в целом, для профилей с острыми кромками и угловыми точками. Именно на концах пластины (одном или обоих, рис. 1) или в угловых точках профиля задается положение схода вихревой пелены; сама сходящая вихревая пелена, формирующая вихревой след, рассматривается как тонкая поверхность разрыва касательной компоненты скорости.

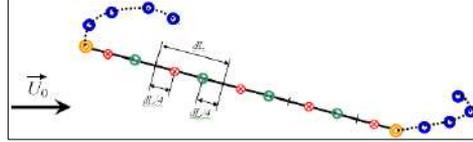


Рис. 1. Расчетная схема моделирования обтекания тонкой пластинки методом дискретных вихрей со сходом вихревых пелен с обоих концов. Показано разбиение пластинки на панели; положения контрольных точек (красным), присоединенных вихрей (зеленым), свободных вихрей (оранжевым) и вихревые следы – тонкие поверхности разрыва, моделируемые точечными вихрями (синим)

Fig. 1. Numerical scheme for flow simulation around a thin plate using the method of discrete vortices with vortex sheets shedding from both ends. Plate splitting into panel is shown as well as positions of control points (red), attached vortices (green), free vortices (orange) and vortex sheets that are modelled as thin discontinuity surfaces, represented by point vortices

Подобная модель хоть и является довольно грубой, но позволяет решить ряд представляющих практический интерес задач. Принципиальный ее недостаток – невозможность решать задачи по расчету обтекания гладких профилей, на которых положение точки отрыва потока априори задать невозможно. Также нет возможности моделировать течения, характеризующиеся низкими числами Рейнольдса, во многом определяемые силами вязкости.

Развитие вихревых методов моделирования плоских течений в значительной мере пошло именно по пути разработки алгоритмов решения уравнений Навье – Стокса с учетом влияния вязкости. На сегодня известны метод случайных блужданий (random walk), разработанный в 1970-х годах [15], метод обмена интенсивностями (particle strength exchange, PSE), предложенный в 1989 году [16], метод распределения ядра вихря (core spreading), впервые предложенный в 1973 г. [17] и позже развитый в 1996 году [18], метод диффузионной скорости, известный в нескольких модификациях, и некоторые другие. Именно к последнему относится метод вязких вихревых доменов [2, 8], реализованный в коде VM2D и обладающий, как представляется, рядом преимуществ перед остальными методами.

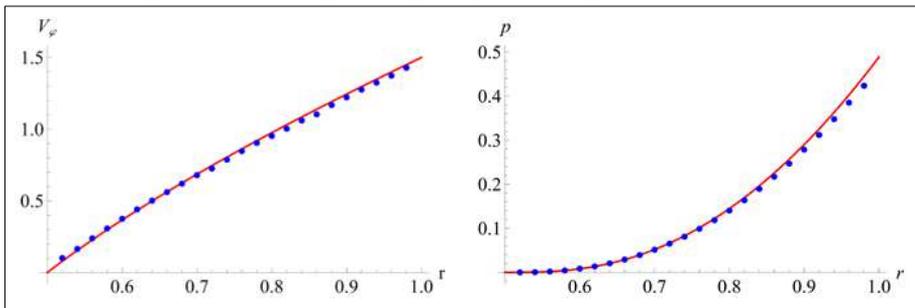


Рис. 2. Профили окружной компоненты скорости и давления в течении Куэтта между соосными цилиндрами (внутренний радиусом 0.5 неподвижен, внешний радиусом 1 вращается); точки – расчет, сплошная линия – точное решение в зависимости от расстояния до оси вращения

Fig. 2. Velocity and pressure profiles for Couette flow between co-rotating cylinders (internal cylinder of radius 0.5 is immovable, external one with radius 1 is rotating); dots indicate numerical solution; solid line represents exact solution against the distance to the axis

Методические эксперименты по выполнению расчетов при помощи кода VM2D показывают, что течения с преобладанием вязких эффектов удается моделировать довольно качественно: профили скоростей в течении Пуазейля в канале, в течении Куэтта между двумя соосными вращающимися цилиндрами (рис. 2), в пограничном слое на тонкой пластине в задаче Блазиуса (рис. 3) воспроизводятся весьма точно, причем даже при сравнительно малом количестве вихревых частиц.

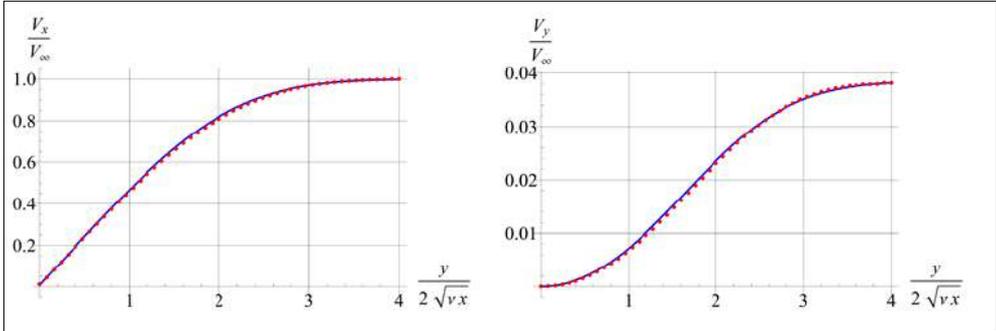


Рис. 3. Профили горизонтальной V_x и вертикальной V_y компонент скорости в пограничном слое на тонкой пластине (задача Блазиуса); представлена зависимость от автомодельной переменной (ν – кинематическая вязкость, V_∞ – скорость набегающего потока); точки – расчет, сплошная линия – точное решение.

Fig. 3. Horizontal V_x and vertical V_y dimensionless velocity profiles in the cross section on a thin plate (Blasius problem); the dependency against the self-similar variable is shown (ν is kinematic viscosity, V_∞ is incident flow velocity); dots indicate numerical solution; solid line represents exact solution.

Намного сложнее обстоит дело с моделированием обтекания профилей при умеренных и повышенных значениях числа Рейнольдса, когда, с одной стороны, влияние сил вязкости сравнительно мало (а вдали от границы профиля, в глубине области течения – в большинстве случаев пренебрежимо!), но при этом его требуется учитывать корректно, чтобы правильно моделировать процессы, происходящие в пристеночном слое, и таким образом правильно воспроизводить отрыв потока от профиля.

3. Моделирование нестационарного обтекания кругового цилиндра

С учетом того, что моделирование нестационарных и переходных режимов обтекания является той областью, где применение для расчетов вихревых методов может быть наиболее эффективно, рассмотрим тестовую задачу о моделировании обтекания цилиндра, мгновенно приведенного в движение с постоянной скоростью в неподвижной среде.

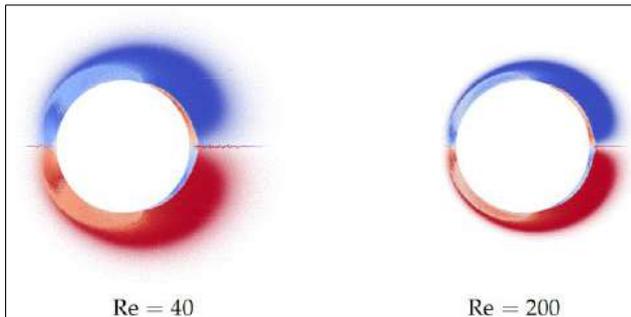


Рис. 4. Положения вихревых частиц за мгновенно приведенным в движение круговым цилиндром $Re = 40$ и $Re = 200$ в момент времени $t_* = 0.5$

Fig. 4. Vortex particles positions in vortex wakes after the impulsively started circular cylinder at $Re = 40$ and $Re = 200$ at time moment $t_* = 0.5$

Расчет нестационарной силы лобового сопротивления, действующей на цилиндр при различных значениях числа Рейнольдса, является «стандартной» тестовой задачей, которая рассматривалась многими исследователями. На рис. 4 показаны вихревые следы в начальной фазе движения цилиндра, рассчитанные при помощи кода VM2D при идентичных постановках задачи, различающихся только величинами коэффициента вязкости среды, соответствующими значениями числа Рейнольдса $Re = 40$ и $Re = 200$. В качестве масштаба времени выбрано отношение диаметра цилиндра к скорости набегающего потока.

Зависимости суммарной силы сопротивления, обусловленной распределением давления и вязким трением, от безразмерного времени для рассмотренных случаев показаны на рис. 5 в сравнении с результатами [19–21]. Видно, что результаты моделирования хорошо согласуются с результатами, полученными другими исследователями.

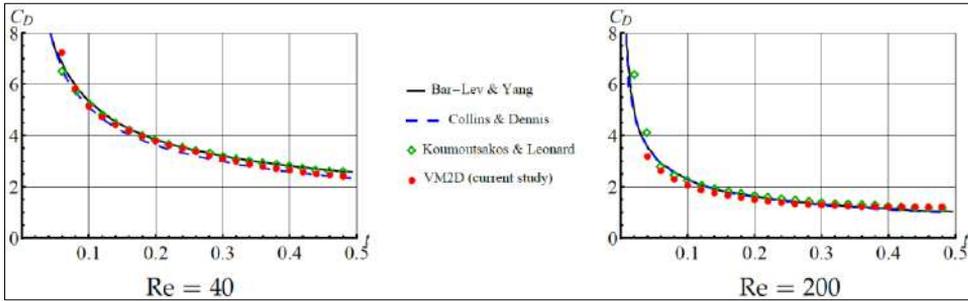


Рис. 5. Нестационарный коэффициент лобового сопротивления

для мгновенно приведенного в движение кругового цилиндра при $Re = 40$ и $Re = 200$

Fig. 5. Unsteady drag force coefficient acting on the impulsively started circular cylinder at $Re = 40$ and $Re = 200$

Аналогичное моделирование было выполнено для скорости движения цилиндра, соответствующей более высокому значению числа Рейнольдса $Re = 3000$. Конечное безразмерное время моделирования теперь было выбрано равным $t_* = 5.0$. Вихревой след в моменты времени $t = 1.0$, $t = 2.0$, $t = 3.0$, $t = 4.0$, $t = 5.0$ показаны на рис. 6.

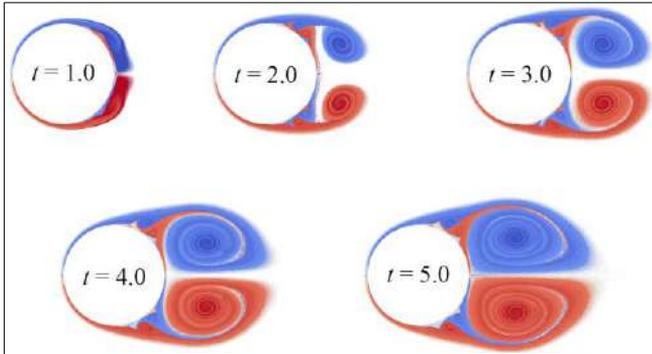


Рис. 6. Положения вихревых частиц за мгновенно приведенным в движение круговым цилиндром при $Re = 3000$ в разные моменты времени

Fig. 6. Vortex particles positions in vortex wakes after the impulsively started circular cylinder at $Re = 3000$ at different time moments

Эта задача оказывается значительно более сложной. Чтобы обеспечить согласованность результатов моделирования с приведенными в литературе [21–27] данными для $Re = 3000$, потребовалось рассмотрение весьма тщательной дискретизации профиля, малого шага по времени, достаточно большого количества вихревых частиц (порядка полумиллиона) в области течения (порядка полумиллиона). Столь подробное разрешение в совокупности с использованием T -схем решения граничного интегрального уравнения позволили избежать

возникновения осцилляций действующих сил, характерных для многих реализаций вихревых методов. На рис. 7 показаны результаты расчета нестационарной зависимости силы лобового сопротивления от времени, полученные при использовании кода VM2D в сравнении с результатами других авторов.

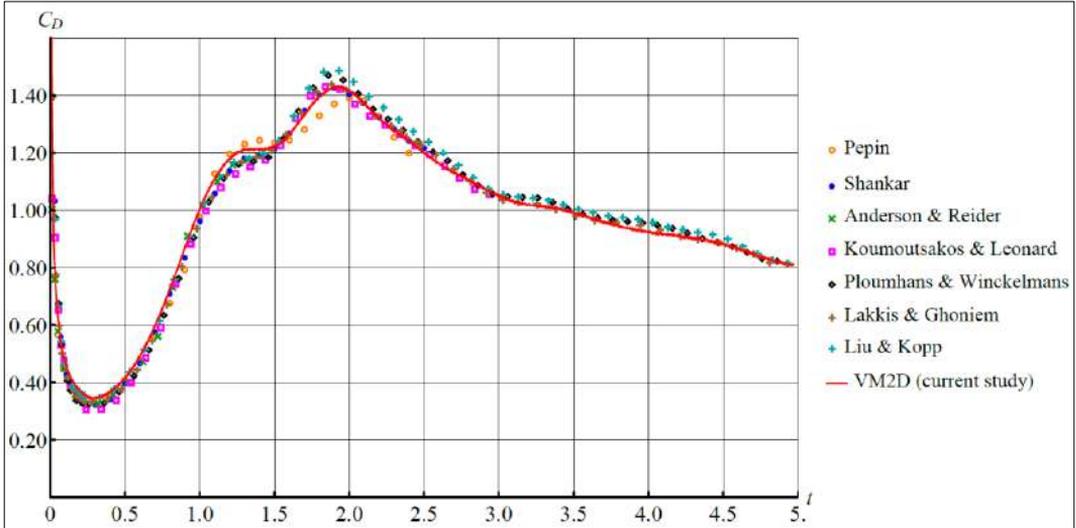


Рис. 7. Нестационарный коэффициент лобового сопротивления для мгновенно приведенного в движение кругового цилиндра при $Re = 3000$

Fig. 7. Unsteady drag force coefficient acting the impulsively started circular cylinder at $Re = 3000$

Попытки моделирования течений вокруг кругового профиля, характеризующихся большими значениями числа Рейнольдса, приводят к нежелательному результату: с использованием VM2D при моделировании квазистационарного режима обтекания результат оказывается качественно правильным (образуется вихревая дорожка Кармана с поочередным срывом вихрей), но количественно неверным.

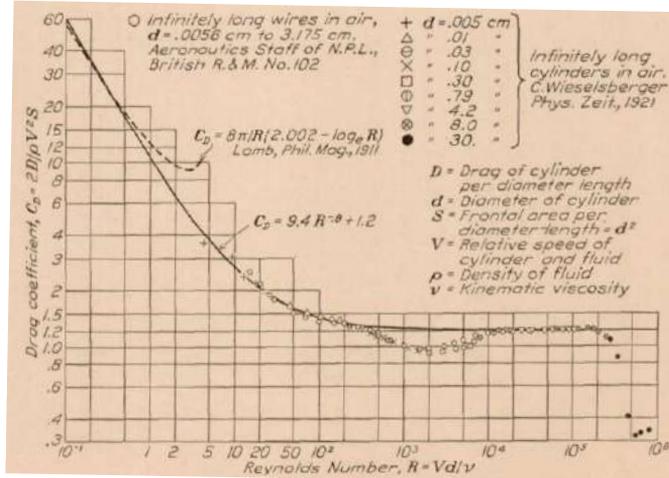


Рис. 8. Стационарный коэффициент силы лобового сопротивления, действующей на круговой цилиндр, в зависимости от числа Рейнольдса [28]

Fig. 8. Stationary drag coefficient acting on a circular cylinder in dependence to Reynolds number [28].

К примеру, не удастся воспроизвести известный эффект «стабилизации» коэффициента лобового сопротивления, в соответствии с которым стационарный (осредненный)

коэффициент лобового сопротивления цилиндра остается близким к 1.2 в широком диапазоне чисел Рейнольдса, а в области $Re \approx 1000$ даже опускается ниже единичного значения (рис. 8). При проведении расчетов же вихревыми методами коэффициент сопротивления оказывается существенно завышенным, близким к 1.5...1.6. Кроме того, из рис. 8 видно, что значение $Re = 2 \cdot 10^5$ соответствует так называемому «кризису обтекания», после которого наблюдается резкое снижение коэффициента лобового сопротивления. Качественный эффект «кризиса сопротивления» наблюдается и в расчетах вихревым методом (на это также указано в работе [29]), однако он наблюдается раньше, примерно при $Re = 10^5$.

4. Моделирование обтекания крыловых профилей

Рассмотрим задачу моделирования обтекания крыловых профилей при средних значениях числа Рейнольдса. На рис. 9 показан вихревой след за симметричным крыловым профилем NASA-0012, установленным под углом атаки $\alpha = 6^\circ$ в установившемся режиме обтекания при $Re = 10^4$ и $Re = 10^5$ [10]. Параметры расчета были выбраны аналогичными таковым для рассмотренного в предыдущем разделе случая обтекания цилиндра.

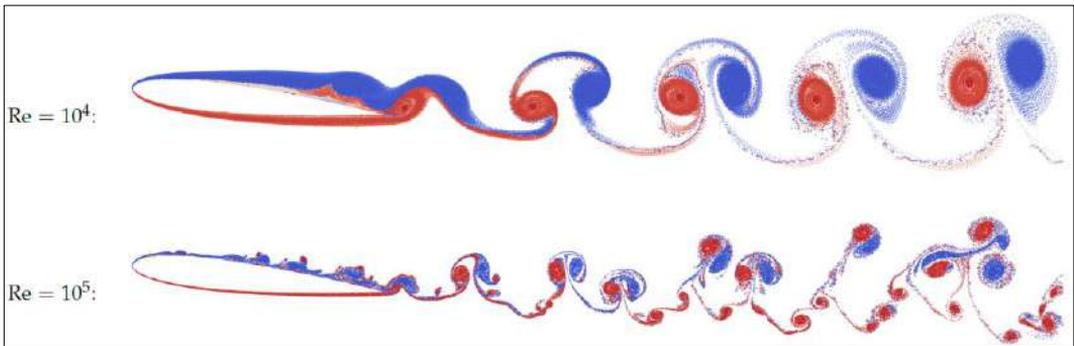


Рис. 9. Вихревые среды за профилем NASA-0012 для угла атаки $\alpha = 6^\circ$ при $Re = 10^4$ и $Re = 10^5$
Fig. 9. Vortex wake after the NASA-0012 airfoil for angle of incidence $\alpha = 6^\circ$ at $Re = 10^4$ and $Re = 10^5$

Полученные в расчете зависимости стационарных (осредненных) коэффициентов лобового сопротивления и подъемной силы от угла атаки (рис. 10) удовлетворительно согласуются с экспериментальными данными.

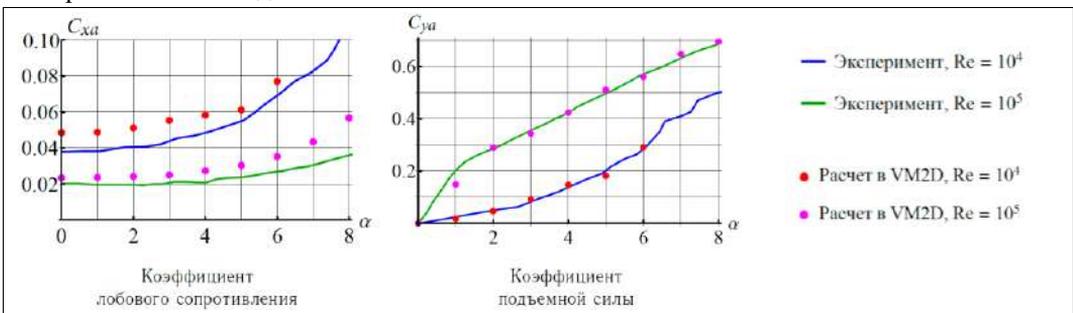


Рис. 10. Зависимости стационарных аэродинамических коэффициентов профиля NASA-0012 от угла атаки

Fig. 10. Drag and lift force coefficients against the angle of incidence for NASA-0012 airfoil

Несмотря на то, что значения коэффициента лобового сопротивления в расчете получаются несколько завышенными, это не является критичным в силу их малости. В целом можно сделать вывод о том, что обтекание моделируется правильно даже при высоких значениях числа Рейнольдса.

Если продолжать увеличивать угол атаки крыла, то результаты расчета в VM2D начинают существенно отличаться от данных эксперимента – главным образом, наблюдается сильно

завышенное значение коэффициента подъемной силы. При этом с увеличением угла атаки меняется и режим обтекания: отрыв происходит не только на задней острой кромке, но и с гладкой верхней поверхности профиля. По-видимому, именно эти эффекты моделируются недостаточно качественно.

5. О моделирование плоских течений в вихревых методах с учетом эффектов турбулентности

На основе представленных выше результатов можно сделать следующий вывод. Решение задачи в плоской постановке позволяет получать корректные результаты в двух случаях: при низких значениях числа Рейнольдса – без разницы, для гладких профилей или профилей с кромками и угловыми точками, а также при высоких значениях числа Рейнольдса, но только в случаях, когда не моделируется отрыв с гладкой поверхности.

В этой связи кажется оправданной следующая гипотеза: в указанных случаях реальное (трехмерное) обтекание конструкции на самом деле по существу является плоским; при более высоких числах Рейнольдса течение если остается плоским, то только на «макромасштабах», тогда как микромасштабные течения оказываются существенно трехмерными, при этом данная «микромасштабная трехмерность» оказывает влияние на течение в целом. Этот эффект хорошо известен, и в широко распространенных сеточных методах воспроизводится с привлечением т.н. моделей турбулентности, в рамках подходов RANS или LES. Именно осредненное (отфильтрованное) течение является плоским, тогда как пульсационная компонента поля скоростей – трехмерной, однако не разрешаемой в явном виде.

Как известно, необходимость введения в рассмотрение специальных замыкающих соотношений, собственно, и выражающих собой содержание той или иной модели турбулентности, связано с невозможностью обеспечить необходимое разрешение на расчетной сетке. Если производить прямое численное моделирование, то хорошо известно, что характерный пространственных масштаб структур в области течения, который должен быть разрешен, носит название колмогоровского масштаба и является обратно пропорциональным величине $Re^{3/4}$. Говоря про мелкомасштабные эффекты, следует упомянуть, что в сравнительно простых моделях турбулентность принимается изотропной в трех пространственных направлениях. Сама «турбулентность» определяется как трехмерное нестационарное движение, в котором вследствие растяжения вихрей создается распределение пульсаций скорости в интервале длин волн от минимального, определяемого колмогоровским масштабом, до максимального, определяемого размером области течения. При этом спектральное распределение кинетической энергии турбулентности имеет так называемую инерционную (промежуточную между областью крупных энергосодержащих вихрей и областью диссипации, в которых энергия вихрей превращается в теплоту в результате действия сил вязкости) область, в которой турбулентность и является однородной и изотропной, а спектральная плотность кинетической энергии турбулентности E в соответствии с законом Колмогорова – Обухова пропорциональна $k^{-5/3}$, где k – волновое число [30].

В вихревых методах моделирования плоских течений «прямое» воспроизведение указанных эффектов в принципе невозможно потому, что первичная расчетная величина – завихренность априори принимается ортогональной области течения, что, собственно, исключает возможность моделирования процесса растяжения вихрей. Тем не менее, если проделывать все выкладки, имея в виду существенно плоский характер течения, то можно получить оценки, характеризующие «двумерную турбулентность»: инерционная область здесь состоит из двух подобластей: для больших волновых чисел справедлива асимптотика $E \sim k^{-3}$, а закон $E \sim k^{-5/3}$ имеет место лишь в сравнительно узком диапазоне [31]. Именно эта зависимость может быть проверена на практике для практических расчетов течения в программном комплексе VM2D.

В качестве примера производилось моделирование обтекания кругового профиля при значении числа Рейнольдса $Re = 10^6$. Полученная в расчете величина стационарного (осредненного) коэффициента лобового сопротивления близка к 0.45 (см. обсуждавшийся выше вопрос о кризисе сопротивления), графики спектральной плотности кинетической энергии турбулентности показаны на рис. 11. Левый график получен путем прямой обработки результатов моделирования: производилось вычисление поля скоростей среды на равномерной сетке с малым шагом в области, имеющей форму квадрата со стороной, равной половине радиуса обтекаемого круга, и расположенной позади него на расстоянии, равном $1/20$ радиуса цилиндра; далее выполнялось двумерное дискретное преобразование Фурье; на график нанесены величины квадратов модулей коэффициентов Фурье в зависимости от модуля волнового вектора; сам график представлен в двойном логарифмическом масштабе. Правый график получен в результате обработки временной зависимости для величины скорости среды в фиксированной точке (была выбрана точка, отстоящая от границы цилиндра на $1/15$ радиуса, ее положение соответствует углу 120°). Производилось дискретное преобразование Фурье, по горизонтальной оси отложена величина $k = f/U$, где f – частота, U – скорость набегающего потока, по вертикальной оси – величина квадрата модуля соответствующего коэффициента Фурье.

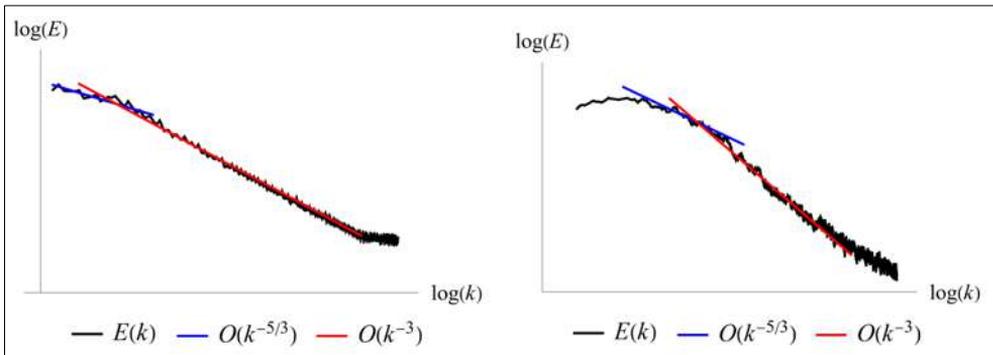


Рис. 11. Графики спектральной плотности кинетической энергии турбулентности (слева – для пространственного поля скоростей в фиксированный момент времени; справа – для зависимости скорости от времени в фиксированной точке пространства)

Fig. 11. Spectral density plots of turbulence kinetic energy (left figure – for spatial velocity field at fixed time; right figure – for velocity dependency against time at fixed point)

Видно, что спектр плотности кинетической энергии «двумерной турбулентности» воспроизводится в инерционной области верно без привлечения каких-либо дополнительных моделей, что дает основание утверждать о достаточной эффективности реализованных в VM2D алгоритмов моделирования плоских течений. Попытки получения подобного результата с использованием сеточных методов, в частности, метода контрольного объема, реализованного в OpenFOAM, без привлечения моделей турбулентности к успеху не приводят (для этого потребовалась бы неприемлемо мелкая сетка и шаг по времени, а следовательно – неадекватные затраты вычислительных ресурсов).

Отметим, что на сегодня известны лишь отдельные работы, в которых предпринимались попытки «совмещения» вихревых методов с RANS и LES подходами к моделированию турбулентности [32–35], однако широкого распространения такие подходы не получили, по крайней мере, при моделировании пристеночных течений. Отчасти это можно объяснить имевшимися до последнего времени сложностями с обеспечением точности решения граничного интегрального уравнения относительно интенсивности генерируемого на профиле вихревого слоя, что в конечном счете существенно ограничивало исследователей в достижении высокого разрешения характеристик течения в пристеночной области. Использование вышеупомянутых T-схем позволяет решить данную проблему и открывает

возможность к построению новых модификаций вихревых методов, включающих возможность моделирования турбулентных эффектов.

6. Заключение

С использованием программного комплекса VM2D рассмотрена возможность применения вихревых методов, основанного на методе вязких вихревых доменов, к решению модельных задач о моделировании обтекания профилей при средних и повышенных значениях числа Рейнольдса: результаты расчетов при малых значениях числа Рейнольдса хорошо согласуются с экспериментальными, однако при больших значениях числа Рейнольдса наблюдается существенная погрешность.

Приведенные зависимости силы лобового сопротивления и подъемной силы для крылового профиля при числах Рейнольдса $Re = 10^4$ и $Re = 10^5$ для малых углов атаки близки к результатам экспериментов. Однако при увеличении угла атаки снова наблюдается расхождение в результатах. Это может быть объяснено тем, что с увеличением угла атаки происходит отрыв потока не только на острой кромке, но и на спинке профиля, и такой отрыв на гладкой поверхности, как и для случая кругового профиля, моделируется недостаточно корректно. Последнее связано с тем, что при высоких значениях числа Рейнольдса течение может моделироваться как плоское, но только на «макромасштабах», тогда как микромасштабные течения являются существенно трехмерными, особенно в зоне отрыва с гладкой поверхности, и такая «микромасштабная» трехмерность, принципиально невозпроизводимая в двумерных алгоритмах вихревых методов, оказывает существенное влияние на течение в целом.

В то же время анализ спектра кинетической энергии турбулентности показывает, что он с высокой точностью соответствует «двумерной турбулентности», из чего можно сделать вывод о достаточной эффективности реализованных в VM2D алгоритмов моделирования плоских течений.

Список литературы / References

- [1] Cottet G.-H., Koumoutsakos P.D. Vortex methods: theory and practice. Cambridge University Press, 2000, 328 p.
- [2] Андронов П.Р., Гувернюк С.В., Дынникова Г.Я. Вихревые методы расчета нестационарных гидродинамических нагрузок. М., Изд-во МГУ, 2006 г.. 184 стр. / Andronov, P.R., Guvernyuk, S.V., Dynnikova, G.Y. Vortex methods for non-stationary hydrodynamic loads estimation. Moscow, Moscow State University, 2006. 184 p. (in Russian).
- [3] Mimeau C., Mortazavi I. A review of vortex methods and their applications: from creation to recent advances. Fluids. 2021, vol. 6, issue 2, article no. 68, 49 p.
- [4] Kempka S.N., Glass M.W. et al. Accuracy considerations for implementing velocity boundary conditions in vorticity formulations. SANDIA Technical Report SAND96-0583, UC-700, 1996, 53 p.
- [5] Дынникова Г.Я. Аналог интегралов Бернулли и Коши — Лагранжа для нестационарного вихревого течения идеальной несжимаемой жидкости. Известия. РАН. Механика жидкости и газа, вып. 1, 2000 г., стр. 31-41. / Dynnikova, G.Y. An analog of the Bernoulli and Cauchy—Lagrange integrals for a time dependent vortex flow of an ideal incompressible fluid. Fluid Dynamic, vol. 35, issue. 1, 2000, pp. 24-32.
- [6] Девнин С.И. Гидроупругость конструкций при отрывном обтекании. Л., Судостроение, 1975 г., 193 стр. / Devnin S.I. Hydroelasticity of Structures under Separated Flow. Leningrad, Sudostroenie, 1975, 193 p. (in Russian).
- [7] Marchevsky I., Sokol K., Ryatina E., Izmailova Y. The VM2D open source code for two-dimensional incompressible flow simulation by using fully Lagrangian vortex particle methods. Axioms, vol.. 12, issue 3, 2023, article no. 248, 33 p.
- [8] Дынникова Г.Я. Лагранжев подход к решению нестационарных уравнений Навье – Стокса. Доклады Академии наук, том 399, вып. 1, 2004 г., стр. 42-46. / Dynnikova G.Ya. The Lagrangian

- approach to solving the time-dependent Navier-Stokes equations. *Doklady Physics*, vol. 49, issue 11, 2004, pp. 648-652.
- [9] Марчевский И.К., Сокол К.С., Измайлова Ю.А. Т-схемы для математического моделирования генерации завихренности на гладких профилях в вихревых методах. Вестник МГТУ им. Н.Э. Баумана. Естественные науки, вып. 6, 2022 г., стр. 33–59 / *Marchevsky I.K., Sokol K.S., Izmailova Yu.A. T-schemes for mathematical modelling of vorticity generation on smooths airfoils in vortex particle methods. Herald of the Bauman Moscow State Technical University, Series Natural Sciences*, issue 6, 2022, pp. 33–59 (in Russian).
- [10] Kuzmina K., *Marchevsky I. et al.* On the scope of Lagrangian vortex methods for two-dimensional flow simulations and the POD technique application for data storing and analyzing. *Entropy*, vol. 23, issue 1, article no. 118, 38 p:
- [11] Belotserkovsky S.M., Lifanov I.K. *Method of Discrete Vortices*. CRC Press, 1992. 464 p.
- [12] Лифанов И.К. Метод сингулярных интегральных уравнений и численный эксперимент (в математической физике, аэродинамике, теории упругости и дифракции волн). М., ТОО «Янус», 1995 г., 520 стр. / *Lifanov I.K. Singular Integral Equations and Discrete Vortices*. De Gruyter, 1996; 485p.
- [13] Katz J., Plotkin A. *Low-speed aerodynamics. From wing theory to panel methods*. McGraw-Hill Book Co., 1991, 632 p.
- [14] McBain G.D. *Theory of Lift: Introductory Computational Aerodynamics in MATLAB/Octave*. Wiley, 2012, 342 p.
- [15] Chorin A.J. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*, vol. 57, issue. 4, 1973, pp. 785-796.
- [16] Degond P., Mas-Gallic S. The weighted particle method for convection-diffusion equations. Part1. The case of an isotropic viscosity. *Mathematics of Computation*, vol. 53, issue 188, 1989, pp. 485-507.
- [17] Kuwahara K., Takami H. Numerical studies of two-dimensional vortex motion by a system of point vortices. *Journal of the Physical Society of Japan*, vol. 34, issue 1, 1973, pp. 247-253.
- [18] Rossi L.F. Resurrecting core spreading vortex methods: A new scheme that is both deterministic and convergent. *SIAM Journal on Scientific Computing*, vol. 17, issue 2, 1996, pp. 370-397.
- [19] Bar-Lev M., Yang H.T. Initial flow over an impulsively started circular cylinder. *Journal of Fluid Mechanics*, vol. 72, issue 4, 1975, pp. 625-647.
- [20] Collins W.M., Dennis S.C.R. The initial flow past an impulsively started circular cylinder. *Quarterly Journal of Mechanics and Applied Mathematics*, vol. 26, issue 1, 1973, pp. 53–75.
- [21] Koumoutsakos P., Leonard A. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics*, vol. 296, 1995, pp. 1–38.
- [22] Pepin F.M. Simulation of the flow past an impulsively started cylinder using a discrete vortex method. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 1990.
- [23] Subramaniam S. A new mesh-free vortex method. Ph.D. Thesis. Florida State University, 1996, 253 p.
- [24] Anderson C.B., Reider M.B. A high order explicit method for the computation of flow about a circular cylinder. *Journal of Computational Physics*, vol. 125, issue 1, 1996, pp. 207-224.
- [25] Ploumhans P., Winckelmans G.S. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics*, vol. 165, issue 2, 2000, pp. 354-406.
- [26] Lakkis I., Ghoniem A. A high resolution spatially adaptive vortex method for separating flows. Part I: Two-dimensional domains. *Journal of Computational Physics*, vol. 228, issue 2, 2009, pp. 491-515.
- [27] Liu Z., Kopp G.A. High-resolution vortex particle simulations of flows around rectangular cylinders. *Computers & Fluids*, vol. 40, issue 1, 2011, pp. 2–11.
- [28] Zahm A.F. Flow and drag formulas for simple quadrics. *NACA Technical Report NACA-TR-253*, 1927, 23 p.
- [29] Дынникова Г.Я. Использование быстрого метода решения задачи N тел при вихревом моделировании течений. Журнал вычислительной математики и математической физики, том. 49, вып. 8, 2009 г., стр. 1458–1465. / *Dynnikova G.Y. Fast technique for solving the N-body problem in flow simulation by vortex methods. Computational Mathematics and Mathematical Physics*, vol. 49, issue 8, 2009, pp. 1389-1396.
- [30] Колмогоров А.Н. Локальная структура турбулентности в несжимаемой вязкой жидкости при очень больших числах Рейнольдса. Успехи физических наук, том 93, вып. 3, 1967 г., стр. 476-481 / *Kolmogorov A.N. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, vol. 434, issue 1890, 1991, pp. 9-13.

- [31] Kraichnan R.H. Inertial Ranges in Two-Dimensional Turbulence. *Physics of Fluids*, vol. 10, issue 7, 1967, pp. 1417-1423.
- [32] Alcântara Pereira L.A., Hirata H., Silveira Neto A. Vortex method with turbulence sub-grid scale modelling // *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 25, issue 2, 2003, 9 p.
- [33] Yokota R., Shinnosuke O. Vortex methods for the simulation of turbulent flows: review. *Journal of Fluid Science and Technology*. vol. 6, issue 1, 2011, pp. 14-29.
- [34] Branlard E., Papadakis G. et al. Aeroelastic large eddy simulations using vortex methods: unfrozen turbulent and sheared inflow. *Journal of Physics: Conference Series*, vol. 625, 2015, article no. 012019, 13 p.
- [35] Alvarez E. J.; Ning A. Reviving the vortex particle method: A stable formulation for meshless large eddy simulation. arXiv preprint arXiv:2206.03658, 2022, 34 p

Информация об авторах / Information about authors

Ирина Александровна КОРОБОВА – аспирантка второго года обучения и ассистент кафедры «Прикладная математика». Научные интересы включают вихревые методы, вычислительную аэродинамику.

Irina Aleksandrovna KOROBOVA – 2nd year post-graduate student and an assistant of the Applied Mathematics Department. Scientific interests include vortex methods, computational aerodynamics.

Евгения Павловна РЯТИНА – аспирантка 3 года обучения кафедры прикладной математики. Сфера научных интересов: вихревые методы, моделирование несжимаемых течений, быстрые алгоритмы и высокопроизводительные вычисления.

Evgeniya Pavlovna RYATINA – 3rd year PhD student of the Applied Mathematics Department. Research interests: vortex methods, incompressible flows numerical simulation, fast methods and high-performance computing

Анна Александровна ХОРОШЕВА – студентка 4 курса кафедры «Математическое моделирование». Сфера научных интересов: механика сплошных сред, вихревые методы, моделирование вязких течений.

Anna Aleksandrovna KHOROSHEVA – 4th year student of the Department of Mathematical Modeling. Research interests: mechanics of continua, vortex methods, modeling of viscous flows.

DOI: 10.15514/ISPRAS-2023-35(2)-15



Особенности численного моделирования работы воздушно-тепловой завесы в OpenFOAM

¹ С.В. Григорьев, 0000-0002-3441-5887 <grigoscool@mail.ru>

² М.Р. Королева, ORCID: 0000-0001-5697-9199 <koroleva@udman.ru>

¹ О.В. Мищенко, ORCID: 0000-0002-8835-1184 <mov@istu.ru>

¹ Ижевский государственный технический университет имени М.Т. Калашникова, 426069, Россия, г. Ижевск, ул. Студенческая, д. 7,

² Удмуртский федеральный исследовательский центр УрО РАН, 426067, Ижевск, ул. Т. Барамзиной, д. 34

Аннотация. Рассматриваются вопросы математического моделирования турбулентного теплопроводного течения сжимаемой вязкой среды во внутреннем объеме корпуса воздушно-тепловой завесы, оснащенной тангенциальным вентилятором. Решение задачи построено на основе осредненных по Рейнольдсу (Фавру) уравнений Навье-Стокса. Решение задачи получено с применением подхода MRF (Multiple Reference Frame), который использует вращающуюся систему отсчета, и с применением преобразования основных уравнений Навье-Стокса в зоне вращения. Для корректного описания протекающих во внутреннем объеме воздушно-тепловой завесы и в окружающей среде рабочих процессов в работе применяются модульные многоблочные сетки, в том числе позволяющие разделить вращающиеся и неподвижные области. Решение поставленных задач строится с использованием инструментов пакета OpenFOAM. В результате работы подробно описаны особенности структуры потока в проточной части воздушно-тепловой завесы, оценены скорости газа, достигаемые при различных скоростях вращения вентилятора. Показана автомодельность профилей скорости на выходе из сопла завесы.

Ключевые слова: математическое моделирование; газовая динамика; воздушно-тепловая завеса; тангенциальный вентилятор; динамические сетки

Для цитирования: Григорьев С.В., Королева М.Р., Мищенко О.В. Особенности численного моделирования работы воздушно-тепловой завесы в OpenFOAM. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 215-224. DOI: 10.15514/ISPRAS-2023-35(2)-15

Peculiarities of numerical simulation of air/heat curtain operation in OpenFOAM

¹ S.V. Grigorev, ORCID: 0000-0002-3441-5887 <grigoscool@mail.ru>

² M.R. Koroleva, ORCID: 0000-0001-5697-9199 <koroleva@udman.ru>

¹ O.V. Mishchenkova, ORCID: 0000-0002-8835-1184 <mov@istu.ru>

¹ Kalashnikov Izhevsk state technical university
7, Studencheskaya st., Izhevsk, 426000, Russia

² Udmurt Federal Research Center UB RAS
34, T. Baramzina Str., Izhevsk, 426067, Russia

Abstract. The issues of mathematical modeling of turbulent heat-conductive flow of compressible viscous medium in the internal volume of the body of an air-thermal curtain equipped with a tangential fan are considered. The solution of the problem is constructed on the basis of averaged Reynolds (Favre) Navier-Stokes equations. The solution of the problem is obtained using the MRF (Multiple Reference Frame) approach, which

uses a rotating reference frame, and using a transformation of the basic Navier-Stokes equations in the rotation zone. In order to correctly describe the working processes occurring in the internal volume of the air-thermal curtain and in the environment, modular multiblock meshes are applied in the work, including those allowing to separate rotating and stationary areas. The solution of the set tasks is constructed using the tools of the OpenFOAM package. As a result of the work, the peculiarities of the flow structure in the flowing part of the air-heat curtain are described in detail, and the gas velocities achieved at different fan speeds are estimated. The self-similarity of velocity profiles at the air curtain nozzle outlet is shown.

Keywords: mathematical modeling; gas dynamics; air-heat curtain; tangential fan; dynamic mesh.

For citation: Grigorev S.V., Koroleva M.R., Mishchenkova O.V. Peculiarities of numerical simulation of air/heat curtain operation in OpenFOAM. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 215-224 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-15

1. Введение

Эксплуатация помещений с большим потоком проходящих людей или транспорта сопровождается потерями тепла через открывающиеся проемы разгрузочные окна, проходы, входные двери и ворота. К таким помещениям относятся больницы, школы, детские сады, спортивные объекты, аэропорты, вокзалы и другие организации сферы услуг (магазины, театры, гостиницы, рестораны), общественный транспорт, производственные и складские помещения. Поддержание температуры воздуха в помещении в заданных пределах возможно с использованием электромеханического устройства – воздушной тепловой завесы. В некоторых случаях использование тепловых завес позволяет обеспечить выполнение особых технологических требований, например, к хранению продуктов питания, т.к. завесы могут ограничивать как поступление холодного воздуха в помещение в зимний период, так и подачу горячего воздуха летом. Кроме этого, завесы препятствуют проникновению внутрь насекомых, пыли, дыма, неприятных запахов и т.д. Но по большей части воздушно-тепловые завесы используются для разделения зон с различной температурой воздушной среды по разные стороны открытых проемов.

Воздушно тепловые завесы различаются по назначению, источнику тепла (в том числе по типу нагревателя), типу установки, по расходу воздуха, расположению и геометрии (сечение корпуса, форма сопла). Одним из основных элементов тепловой завесы является вентилятор. Как правило, в устройствах данного типа используется либо радиальный, либо тангенциальный вентилятор.

Тангенциальный, или диаметральный вентилятор представляет собой тип устройства, в котором передвижение воздушной массы производится перпендикулярно оси движения рабочего колеса. Корпус изделия имеет сходство с корпусом аналогичного устройства центробежного типа. Рабочее колесо диаметрального вентилятора представлено в виде цилиндра, внешняя поверхность которого формируется набором профилированных по потоку лопаток [1-3], что обеспечивает требуемое нагнетание воздушной массы в плоскости, перпендикулярной к оси вентилятора. Всасывающей поверхностью, таким образом, будет являться верхняя горизонтальная плоскость устройства (длина которой совпадает с длиной устройства), а не торцевая поверхность. Воздушная масса перемещается лопастями, а диффузор увеличивает скорость передвижения в требуемом направлении [4].

Принцип работы тангенциального вентилятора основывается на повторном прохождении воздуха через рабочие параллельные лопатки в поперечном направлении, что является характерной геометрической особенностью этой конструкции. Согласно экспериментальным данным движение газа внутри тепловой завесы, оснащенной таким вентилятором, может характеризоваться высокой степенью турбулентности [5-8]. На работу конструкции в целом оказывает существенное влияние место установки вентилятора и геометрия корпуса завесы. Экспериментальные исследования позволяют получить огромный набор данных, характеризующих работу тангенциальных вентиляторов, однако все они касаются

конкретной модели исследуемой воздушно-тепловой завесы и не подходят для целей совершенствования существующих и проектирования новых устройств. В этом случае целесообразно использовать методы вычислительной газовой динамики, которые широко применяются для моделирования аэродинамических течений в различных технических устройствах.

Численное моделирование используется как для исследования нестационарной аэродинамики и теплообмена, протекающих в рабочих (внутренних) областях устройств [5,7, 9-11], так и для расчета формы и траектории оси воздушных струй в помещении после их истечения из воздушно-тепловой завесы [12].

В настоящей работе на основе методов вычислительной газовой динамики анализируется течение вязкого, сжимаемого, теплопроводного газа в рабочем объеме воздушно-тепловой завесы. Для численного моделирования используются библиотеки открытой интегрируемой платформы для решения задач механики сплошных сред OpenFOAM 2.0 [13], а также возможности открытой интегрируемой платформы Salome 8.3.0 [14].

2. Математическая постановка

В работе выполнено моделирование течения сжимаемой вязкой среды на основе осредненных по Рейнольдсу (Фавру) уравнений Навье-Стокса во внутреннем объеме корпуса воздушно-тепловой завесы, оснащенной тангенциальным вентилятором (рис. 1). Рассматривается тангенциальный вентилятор, состоящий из рабочего колеса 1, входного патрубка 2, выходного диффузора 3 и корпуса 4. Рабочее колесо содержит 36 равномерно размещенных лопастей, профилированных по потоку.

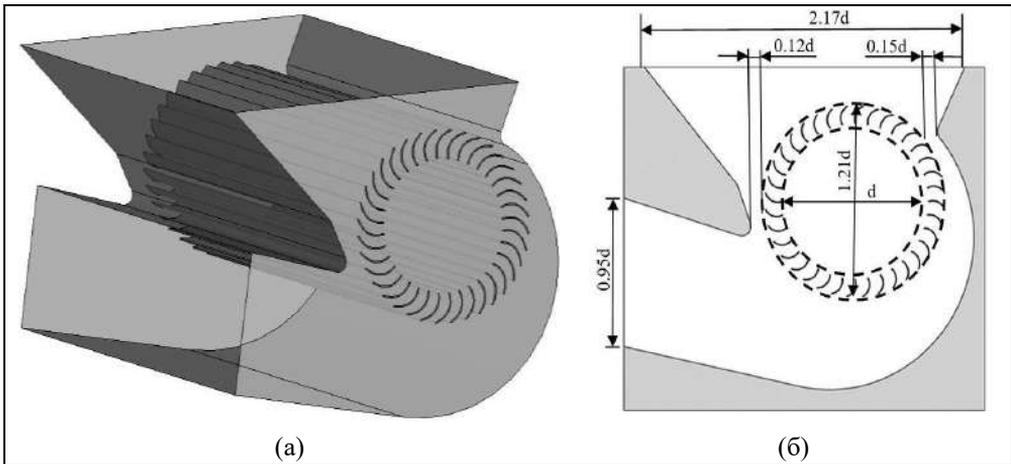


Рис. 1. Геометрия расчетной области: пространственная модель (а) и расчетная область (б)
Fig. 1. Geometry of the computational domain: spatial model (a) and computational domain (b)

При моделировании используются следующие допущения и упрощения:

- рабочее колесо вращается с заданной постоянной скоростью;
- задача решается в плоской постановке;
- газ характеризуется постоянными коэффициентами вязкости и теплопроводности.

В исследовании [15] утверждается, что на выходе воздушной завесы формируется равномерный профиль нагнетаемого воздуха, не зависящий от осевой координаты, направленной вдоль оси рабочего колеса вентилятора. Очевидное двумерное течение на выходе не может существовать без двумерного течения во всей области, и это позволяет нам исследовать газодинамику внутренней воздушной завесы в двумерной постановке.

Таким образом, поставленная задача может быть описана системой уравнений:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) + F_i, \tag{2}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_j}{\partial x_j} = -\frac{\partial p u_j}{\partial x_j} + \frac{\partial u_i \tau_{ij}}{\partial x_j} + \frac{\partial q_j}{\partial x_j} + F_i u_i, \tag{3}$$

$$p = \rho RT. \tag{4}$$

В приведенных выше формулах (1)-(4) приняты следующие обозначения: ρ – плотность; u_i – компоненты вектора скорости u ; p – давление; μ – динамический коэффициент вязкости; F_i – внешняя объемная сила; $E = C_p T + 0.5u_i^2$ – полная удельная энергия; $H = E + p/\rho = C_p T + 0.5u_i^2 = h + 0.5u_i^2$ – полная удельная энтальпия; $\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij}$ – тензор вязких напряжений; $S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ – тензор скоростей деформаций; $q_j = \lambda \frac{\partial T}{\partial x_j}$ – тепловой поток; λ – коэффициент теплопроводности; T – температура; $R = 287 \text{ Дж}/(\text{кг} \cdot \text{К})$ – удельная газовая постоянная.

Исследуемое течение не предполагает больших перепадов температуры, поэтому коэффициенты молекулярной вязкости и теплопроводности считаются постоянными $\mu = 1,85 \cdot 10^{-5} \text{ Па} \cdot \text{с}$, $\lambda = 0,0259 \text{ Вт}/(\text{м} \cdot \text{К})$.

Исходная система уравнений (1)-(4) осредняется по Фавру и Рейнольдсу [2, 17]. Величины представляются в виде суммы $\theta = \bar{\theta} + \theta''$, где $\bar{\theta} = \overline{\rho\theta}/\bar{\rho}$, где $\overline{\rho\theta}$, $\bar{\rho}$ – осредненные согласно процедуре осреднения Рейнольдса параметры. Для замыкания осредненной системы уравнений применяется модель турбулентности SST [18]. Все константы модели рассчитаны в соответствии [18, 19] со значениями из стандартных моделей $k - \epsilon$ и $k - \omega$ и представлены в [2, 4, 9, 18].

Для адекватного описания всасывания воздуха из окружающей среды и его выброса через выходное сопло завесы была построена сложная многоблочная конечно-объемная сетка (рис. 2), которая содержит 4 зоны: 1 – зона входа (забор воздуха), 2 – внутренний объем тепловой завесы, 3 – вентилятор, 4 – зона выхода (выброс воздуха). 1, 2 и 4 зоны являются неподвижными, 3 зона – вращается с заданной угловой скоростью Ω . Сетка во вращающейся и неподвижных областях неструктурированная и состоит из треугольных элементов. Общее количество элементов – 18366 ячеек суммарно для областей и 4677 ячеек для области 3.

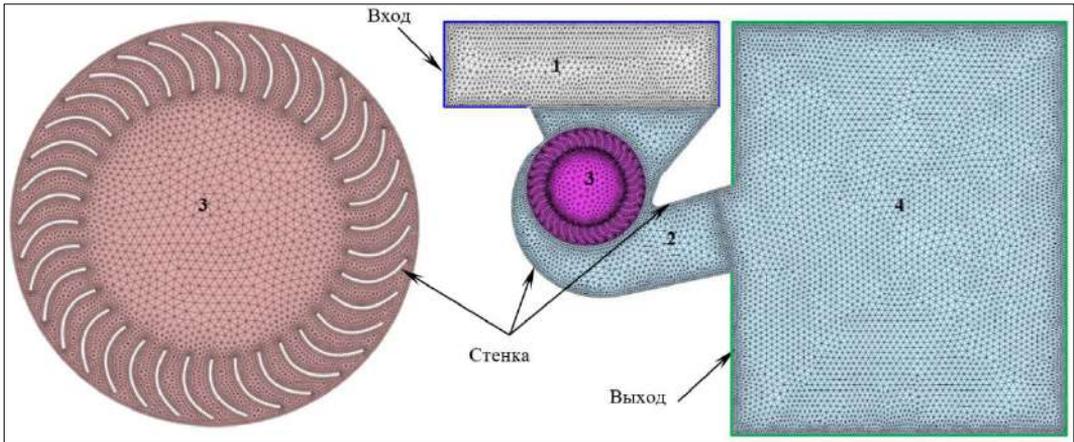


Рис. 2. Конечно-объемная сетка и граничные условия
 Fig. 2. Finite volume grid and boundary conditions

В работе при решении задачи был использован подход [11,15], который использует локальную вращающуюся систему отсчета в области рабочего колеса (зона 3). Это приводит к преобразованию основных уравнений Навье-Стокса в зоне вращения - чтобы учесть эффект вращения исходные уравнения (6) и (7) в данной зоне дополняются слагаемым, учитывающим силу Кориолиса $F = \Omega \times u$.

Данный подход используется для моделирования установившихся течений. Таким образом, тело и сетка вокруг него в зоне вращения физически не вращаются. Такой подход с вычислительной точки зрения менее требователен по сравнению с нестационарным моделированием, и при корректно поставленной задаче обеспечивает хорошую аппроксимацию с меньшими вычислительными затратами и значительно меньшим временем расчета.

Граничные условия для поставленной задачи представлены в таблице 1.

Табл. 1. Граничные условия

Table 1. Boundary conditions

		Параметр	Условие
Границы области	Вращающаяся зона	Лопастей вентилятора	$\frac{\partial p}{\partial n} = 0, \frac{\partial T}{\partial n} = 0, u = 0$
	Неподвижные зоны	Вход	$p_{total} = 10^5 \text{Па},$ $T = 313\text{К}$
		Выход	$p_{out} = 10^5 \text{Па}$
		Стенка корпуса	$\frac{\partial p}{\partial n} = 0, \frac{\partial T}{\partial n} = 0, u = 0$
		Скорость вращения	250-1500 об/мин

Вопросы построения метода контрольных объемов, численных схем и ряда расчетных алгоритмов подробно освещены в работе [16]. Сформированная таким образом математическая модель реализована с использованием решателя rhoPimpleDyMFoam.

3. Анализ результатов численного моделирования

В результате численного моделирования движения воздуха во внутреннем объеме тепловой завесы получены структура течения в виде линий тока, поля давлений, температуры и скорости газа при вращении вентилятора со скоростью от 250 об/мин до 1500 об/мин. Дополнительно построены профили скорости и температуры газа на выходной границе – на срезе выходного сопла. Определена зависимость производительности воздушно-тепловой завесы (ВТЗ) по воздуху от скорости вращения.

Сравнение линий тока и полей распределения основных параметров потока показало, что структура течения не зависит от скорости вращения лопастей вентилятора и является одинаковой для всех рассматриваемых режимов работы. Характерные для всех рассмотренных скоростей вращения тангенциального вентилятора поля физических величин (скорости, температуры и давления) приведены на рис. 3. Видны локальные особенности распределения давления и температуры по поперечному сечению воздушно-тепловой завесы, обусловленные структурой течения. Так локальный минимум давления и температуры газа соответствует местоположению вихревой зоны 3 на рис. 4. Понижение давления по отношению к атмосферному зависит от скорости движения рабочего колеса и составляет от 20 Па при 250 об/мин до 1000 Па при скорости вращения 1500 об/мин.

Рассмотрим особенности течения газа в рабочем объеме завесы (рис. 3, 4). За счет работы вентилятора, воздух всасывается в рабочий объем воздушно-тепловой завесы, направляясь по входному патрубку к рабочему колесу. Часть потока проходит по задней стенке завесы, а другая часть сначала попадает внутрь рабочего колеса, а затем изнутри наружу, что приводит

к значительному ускорению газа. Далее поток направляется в нижнюю полость завесы, откуда выбрасывается в атмосферу.

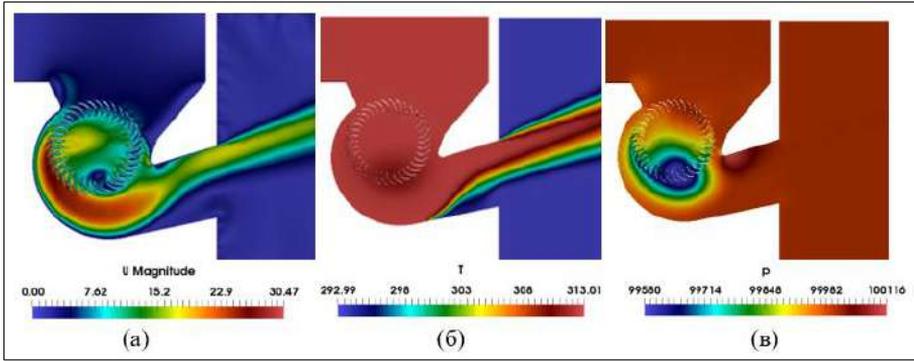


Рис. 3. Поля скорости (а), давления (б) и температуры (в), характерные для рассмотренных режимов работы воздушно-тепловой завесы

Fig. 3. Fields of velocity (a), pressure (b) and temperature (c), typical for the considered modes of operation of the air-heat curtain

Однако часть основного потока газа, не успевая в узкий зазор между лопастями рабочего колеса и передней стенкой завесы снова поступает внутрь рабочего колеса, и, подчиняясь вращению лопастей вентилятора, образует возвратное течение 3 (рис. 4), которое в результате взаимодействия с постоянно поступающим из входного сечения потоком воздуха, смещается от оси вентилятора вниз к лопастям. Формируемая таким образом стационарная вихревая зона 3 является конструктивным разделителем областей всасывания и нагнетания воздуха. Взаимодействие возвратного течения с основным потоком, направляющимся в сопло, приводит к образованию линии разделения потока на входе в сопло 5 (рис. 4) и к образованию прижатой к верхней стенке сопла струи. Формирование струи в выходном сечении сопла сопровождается отрывом потока и, как следствие, образованием области с пониженным давлением и возвратным течением 4 (рис. 4).

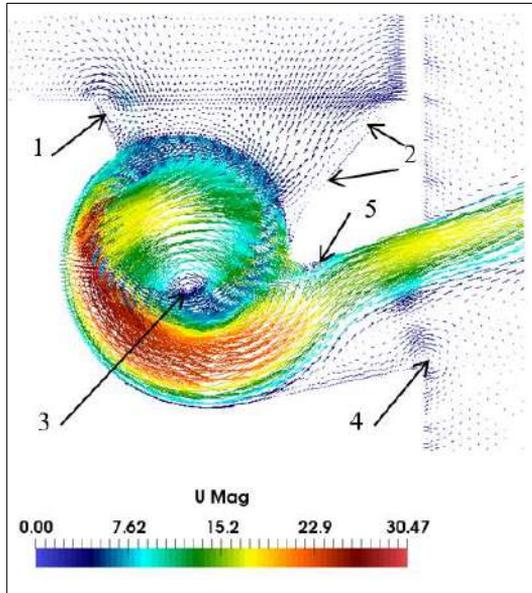


Рис. 4. Структура потока в виде векторов скорости

Fig. 4. Flow structure in the form of velocity vectors

Разделение потока сопровождается также перестройкой течения вблизи верхней стенки выходного сопла и переориентацией течения. В точке сочленения входного патрубка с передней стенкой ВТЗ, в результате взаимодействия вихревого течения с поступающим из атмосферы воздухом, образуется система несимметричных парных вихрей 2 (рис. 5).

Еще одной характерной особенностью течения является образование устойчивого вихря 1 (рис. 5) во входном сечении вблизи задней стенки, расположение которого обусловлено геометрией входного сечения воздушно-тепловой завесы и конструкцией корпуса.

Данная картина течения воздуха характерна для всех рассмотренных скоростей вращения вентилятора. Параметры выходной струи отличаются только значением максимальной скорости, которая изменяется от 4,6 м/с (250 об/мин) до 25 м/с (1500 об/мин). При этом влияние на размер, положение и форму вихревых зон скорость вращения вентилятора не оказывает.

На рисунке 5а показаны профили продольной компоненты скорости на срезе выходного сопла воздушно-тепловой завесы при различных скоростях вращения. Видно, что расположение зоны возвратного течения одинаково для всех скоростей, а значения максимума скорости (отрицательной и положительной) тем больше, чем выше скорость вращения. При вращении вентилятора со скоростью 250 об/мин максимум скорости равен 3,62 м/с, при вращении со скоростью 1500 об/мин – 24,56 м/с. Анализ профилей скорости показал автомодельность безразмерной скорости газа (рис. 5б).

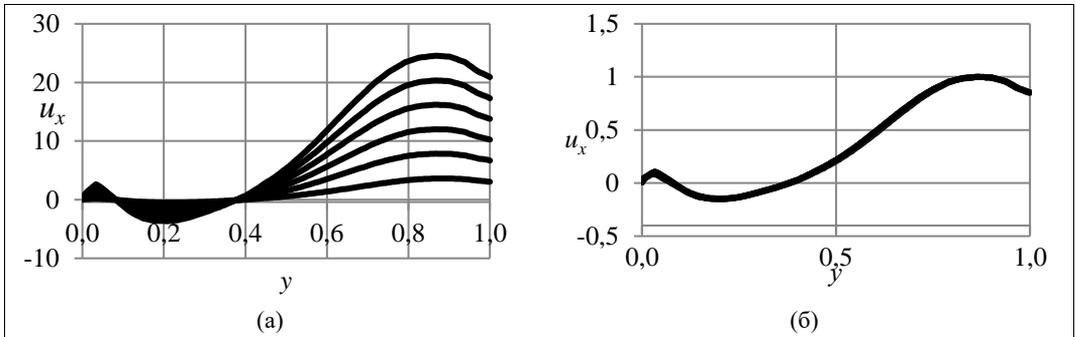


Рис. 5. Профиль продольной компоненты скорости на срезе сопла при различных скоростях вращения вентилятора: а) размерные, б) безразмерные профили

Fig. 5. Longitudinal velocity component profile in the nozzle cross-section at different fan rotation rates: a) dimensional, b) dimensionless profiles

4. Заключение

В работе на основе методов математического моделирования исследуется течение вязкого, сжимаемого, теплопроводного газа в рабочем объеме воздушно-тепловой завесы. Для учета эффектов вращения лопастей тангенциального вентилятора применены многоблочные расчетные сетки с применением подхода MRF. Получена структура течения в виде линий тока, поля давлений, температуры и скорости газа при вращении вентилятора со скоростью от 250 об/мин до 1500 об/мин. Построены профили скорости газа на выходной границе – на срезе выходного сопла.

Показано, что структура течения не зависит от скорости вращения вентилятора и является одинаковой для всех рассматриваемых режимов работы, профили скорости являются автомодельными.

Изучены топологические особенности течения газа в рабочем объеме завесы. Показано, что структура течения характерна для работы тангенциальных вентиляторов и соответствует экспериментальным данным.

Показано, что локальные особенности распределения давления и температуры по поперечному сечению воздушно-тепловой завесы обусловлены структурой течения. При этом понижение давления по отношению к атмосферному зависит от скорости движения рабочего колеса и составляет от 20 Па при 250 об/мин до 1000 Па при скорости вращения 1500 об/мин.

Список литературы / References

- [1] Frank D., Linden P. The effectiveness of an air curtain in the doorway of a ventilated building. *Journal of Fluid Mechanics*, vol. 756, 2014, pp. 130-164.
- [2] Koroleva M.R., Mishchenkova O.V. et al. Theoretical research of the internal gas dynamics processes of measurements of hot air curtain with cross-flow fan. *MM Science Journal*, June 2020, pp. 3966-3972.
- [3] Шарфарец Б.П., Дмитриев С.П. Моделирование турбулентного движения жидкости на основе гипотезы Буссинеска. Обзор. *Научное приборостроение*, том 28, вып. 3, 2018 г., стр. 101–108 / Sharfarets B.P., Dmitriev S.P. Modeling of turbulent fluid motion based on the Boussinesq hypothesis. Overview. *Nauchnoe Priborostroenie*, vol. 28, issue 3, 2018, pp. 101—108 (in Russian).
- [4] Королева М.Р., Сбоева Т.А. и др. Исследование течения воздуха в рабочем объеме воздушно-тепловой завесы. *Химическая физика и мезоскопия*, том 21, вып. 3, 2019 г., стр. 403-409 / Koroleva M.R., Sboeva T.A. et al. The study of air flow in the working volume of the air-thermal curtain. *Chemical Physics and Mesoscopics*, vol. 21, issue 3, 2019, pp. 403-409 (in Russian).
- [5] Darmawan S., Siswantara A.I. et al. Turbulent flow analysis in auxiliary cross-flow runner of a Proto X-3 Bioenergy micro gas turbine using RNG K- ϵ turbulence model. *Journal of Engineering and Applied Sciences*, vol. 10, issue 16, 2015, pp. 7086-7091.
- [6] Khosrowpanah S., Fiuzat A.A., Albertson M.L. Experimental Study of Cross-Flow Turbine. *Journal of Hydraulic Engineering*, vol. 114, issue 3, 1988, pp. 299-314.
- [7] Casarsa L., Giannattasio P. Experimental study of the three-dimensional flow field in cross-flow fans. *Experimental Thermal and Fluid Science*, vol. 35, issue 6, 2011, pp.948-959.
- [8] Sun K., Ouyang H et al. Experimental and numerical investigations on the eccentric vortex of the cross flow fan. *International Journal of Refrigeration*, vol. 50, 2015, pp. 146-155.
- [9] Li J., Hou Y. et al. Window purifying ventilator using a cross-flow fan: Simulation and optimization. *Building Simulation*, vol. 9, issue 4, 2016, pp. 481-488.
- [10] Kim T.A., Kim D.W., Park S.K. *Journal of Mechanical Science and Technology*, vol. 22, issue 10, 2008, pp. 1876-1882.
- [11] Toffolo A. On the theoretical link between design parameters and performance in cross-flow fans: a numerical and experimental study. *Computers & Fluids*, vol. 34, issue 1, 2005, pp. 49-66.
- [12] Lu F.K., Chauhan V.A. et al. Numerical modeling of doorway flow induced by an air curtain. In *Proc. of the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, 2006, 7 p
- [13] OpenFOAM. Free CFD Software. Available at: <http://openfoam.org/>.
- [14] Salome. Version 8.3.0. Available at: <http://www.salome-platform.org/>.
- [15] Хюлла И.Я. Прохождение воздуха в вентиляторах поперечного потока / Hyulla I.Ya. Passage of air in cross-flow fans. Available at: www.scad-technology.ru/article/vozduhopotok-v-ventilyatore/ (in Russian).
- [16] Chernova A.A. Validation of rans turbulence models for the conjugate heat exchange problem. *Russian Journal of Nonlinear Dynamics*, vol. 18, issue 1, 2022, pp. 61-82.
- [17] Menter F.R., Kuntz M., Langtry R. Ten years of industrial experience with the SST turbulence model. In *Proc.of the Fourth International Symposium on Turbulence, Heat and Mass Transfe*, 2003, pp. 625-632.
- [18] Isaev S., Popov I. et al. Abnormal enhancement of separated turbulent air flow and heat transfer in inclined single-row oval-trench dimples at the narrow channel wall. *Acta Astronautica*, vol. 163, part 1, 2019, pp. 202-207.
- [19] Круглова Е.С. Разработка ресурсосберегающих воздушно-тепловых завес для поддержания нормируемых параметров микроклимата в производственных помещениях АПК. Автореферат диссертации на соискание ученой степени кандидата технических наук. Челябинский государственный агроинженерный университет, 2006 г., 21 стр. / Kruglova E.S. Development of resource-saving air-heat curtains to maintain the normalized parameters of the microclimate in the industrial premises of the agro-industrial complex. Abstract of the dissertation for the degree of candidate of engineering sciences. Chelyabinsk State Agroengineering University, 2006, 21 p. (in Russian)

Информация об авторах / Information about authors

Сергей В. ГРИГОРЬЕВ – аспирант кафедры «Тепловые двигатели и установки». Сфера научных интересов: математическое моделирование, разностные схемы.

Sergey V. GRIGOREV – postgraduate student of the Department of Thermal Engines and Installations. Research interests: mathematical modeling, difference schemes.

Мария Равилевна КОРОЛЕВА – кандидат физико-математических наук, доцент, старший научный сотрудник. Сфера научных интересов: математическое моделирование, разностные схемы, газовая динамика, гидродинамика, кавитация, теплообмен.

Maria Ravilevna KOROLEVA – Candidate of Physical and Mathematical Sciences, Associate Professor, Senior Researcher. Research interests: mathematical modeling, difference schemes, gas dynamics, hydrodynamics, cavitation, heat exchange.

Ольга Владимировна МИЩЕНКОВА – кандидат физико-математических наук, доцент кафедры «Тепловые двигатели и установки». Сфера научных интересов: математическое моделирование, разностные схемы.

Olga Vladimirovna MISHCHENKOVA – Candidate of Physical and Mathematical Sciences, Associate Professor, Department of Thermal Engines and Installations. Research interests: mathematical modeling, difference schemes.

DOI: 10.15514/ISPRAS-2023-35(2)-16



Параметрические исследования течений в микросоплах

Е.А. Митрюкова, ORCID: 0000-0002-5350-5366 <mit_e_a@mail.ru>

*Ижевский государственный технический университет имени М.Т. Калашиникова,
426069, Россия, г. Ижевск, ул. Студенческая, д. 7*

Аннотация. Работа посвящена параметрическим исследованиям течения криптона в коническом микросопле при истечении в область с низким давлением. Изучены особенности течений при различных значениях давления торможения в предсопловом объеме, в том числе при возникновении конденсированной фазы в потоке. Математическое моделирование проводилось на основе численного решения полной системы уравнений Навье-Стокса, дополненной уравнением массовой доли конденсата и слагаемыми в уравнении сохранения энергии, отвечающими за теплоту фазового перехода. В математической модели учитывалось изменение коэффициентов динамической вязкости и теплопроводности от температуры газа. Расчетная область включала в себя само микросопло и часть пространства за ним. Задача решалась методом контрольного объема на блочно-структурированной регулярной сетке из четырехугольных элементов с использованием схем второго порядка точности. По времени интегрирование уравнений проводилось с использованием метода Рунге-Кутты. Расчеты проводились при значениях давления торможения 5, 10 и 15 атм для однофазного и двухфазного течений. Приведены поля распределения температуры и числа Маха в сопле и в пространстве за ним. Изучено осевое распределение давления, температуры и числа Маха. Показано, что в случае однофазного потока наблюдается автомодельность течений газа. Поля давлений были подобны, а в безразмерном виде совпадали между собой. При этом наблюдалась идентичность полей скорости и температуры при различных значениях давления торможения. Автомодельность течения нарушается в зоне формирования конденсированных частиц. Размеры зон локального повышения температур, а также интенсивность тепловыделения зависят от заданного давления торможения, что отражается на скоростных характеристиках течения.

Ключевые слова: микросопло; криптон; уравнения Навье-Стокса; конденсированная фаза; математическое моделирование.

Для цитирования: Митрюкова Е.А. Параметрические исследования течений в микросоплах. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 225-234. DOI: 10.15514/ISPRAS-2023-35(2)-16

Parametric investigations of flows in micronozzles

E.A. Mitryukova, ORCID: 0000-0002-5350-5366 <mit_e_a@mail.ru>

*Kalashnikov Izhevsk state technical university
7, Studencheskaya st., Izhevsk, 426069, Russia*

Abstract. The work is devoted to parametric investigations of the krypton flow in a conical micronozzle when flowing into a region with low pressure. The features of the flows are studied at various values of the stagnation pressure in the pre-nozzle volume, including the occurrence of a condensed phase in the flow. Mathematical modeling was carried out on the basis of a numerical solution of the complete system of Navier-Stokes equations, supplemented by the equation for the mass fraction of the condensate. The mathematical model takes into account the change in the coefficients of dynamic viscosity and thermal conductivity depending on the gas temperature. The problem was solved by the control volume method on a block-structured regular grid of quadrangular elements using schemes of the second order of accuracy. The equations were integrated with

respect to time using the Runge-Kutta method. The calculations were carried out at stagnation pressures of 5, 10, and 15 atm for single-phase and two-phase flows. The distribution fields of temperature and Mach number in the nozzle and in the space behind it are presented. The axial distribution of pressure, temperature, and Mach number has been studied. It is shown that in the case of a single-phase flow, self-similarity of gas flows is observed. The pressure fields were similar, but in a dimensionless form they coincided to each other. In this case, the identity of the velocity and temperature fields was observed at different values of the stagnation pressure. The self-similarity of the flow is violated in the zone of formation of condensed particles. The dimensions of the zones of local temperature increase, as well as the intensity of heat release, depend on the given stagnation pressure, which is reflected in the velocity characteristics of the flow.

Keywords: micronozzle; krypton; Navier–Stokes equations; condensed phase; mathematical modeling.

For citation: Mitryukova E.A. Parametric studies of flows in microsopes. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 225-234 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-16

1. Введение

Микромасштабные внутренние течения жидкостей и газов являются предметом многих современных исследований [1-10]. Микрогидродинамика, или микрофлюидика выделяется в отдельную науку, занимающуюся изучением малых движущихся объемов жидкости и способы управления такими микропотоками [4-10]. Микротечения жидкостей встречаются как в естественных условиях, например, при фильтрации через пористые среды, так и в искусственных микросистемах. Широкое распространение микрогидродинамических устройств связано с миниатюризацией технических конструкций и диктуется необходимостью снижения энергопотребления, увеличения быстродействия, повышения чувствительности, расширения функциональных возможностей и другими направлениями совершенствования устройств.

Микрогидродинамическая техника (микронасосы, микроклапана, микромиксеры и т.п.) используется интенсивно в медицине (фармацевтике) [1-4], химии (наноматериалы) [4, 5], биологии [1-3] и микроэлектронике [4, 6-10]. Назначением микрогидродинамических устройств (МГДУ) является транспортировка (перекачка) однофазных или многофазных гетерогенных или гомогенных сред по каналам, характерные размеры которых ограничены несколькими десятками микрометров.

Малые размеры потоков приводят к тому, что часть физических явлений, играющих значительную роль в макроскопических течениях, перестают быть значимыми, например гравитация. В то же время малозначимые для макропотоков физические эффекты в микротечениях становятся основополагающими, например, капиллярные эффекты.

Микротечения имеют несколько особенностей. Во-первых, несмотря на малые характерные размеры МГДУ, изучаемые физико-механические процессы находятся в рамках гипотезы сплошной среды, т.е. для их описания применимы основные законы сохранения механики жидкости и газа. Во-вторых, микроразмерные потоки характеризуются малыми числами Рейнольдса ввиду малых значений характерных геометрических размеров каналов и, как правило, являются ламинарными. Это позволяет не усложнять математическую постановку задачи моделями турбулентности. В первую очередь это касается несжимаемых сред, вязкость которых имеет тот же порядок, что и характерные размеры каналов. При этом скорости движения жидкости ограничены давлением насыщенных паров. Жидкостные микроканалы могут иметь различную архитектуру. Это могут быть прямолинейные каналы прямоугольного сечения. Другие МГДУ могут быть представлены в виде древовидной системы разветвляющихся каналов. Такие каналы используются, например, в микрорадиаторах в микроэлектронике [9, 11]. Для смешивания жидкостей могут использоваться микромиксеры в виде Т-образно соединенных каналов [12].

Микромасштабные течения сжимаемых газов могут отличаться от своих макроскопических аналогов тем, что одновременно с низкими числами Рейнольдса в них могут достигаться высокие числа Маха, даже если канал прямолинейный. В сверхзвуковых микросоплах

ускорение газа еще более ощутимое. Такие сопла могут использоваться для получения высокоскоростных сфокусированных пучков микрочастиц [13, 14]. В этом случае газ выступает как несущая среда, разгоняющаяся до сверхзвуковых скоростей и дающая необходимую инерцию дисперсным частицам. Силы вязкого трения вблизи стенок сопла значительно сужают сверхзвуковую область течения и помогают сконцентрировать частицы в центре сопла.

Настоящая работа посвящена параметрическим исследованиям течения криптона в коническом микросопле при истечении в область с низким давлением. Целью проводимых исследований является определение особенностей течений в микросопле при различных значениях давления торможения в предсопловом объеме, в том числе при возникновении конденсированной фазы в потоке. Работа является продолжением, проводившихся ранее в работах [10, 14-16] по изучению течений инертных газов в микросоплах.

2. Постановка задачи и математическая модель

В работе численно исследуется течение криптона в осесимметричном коническом микросопле (рис. 1). Задача решается в осесимметричной постановке. Схема расчетной области показана на рисунке 1. Радиус критического сечения сопла составляет $R_{cr} = 250$ мкм, угол полураствора конуса $4,3^\circ$, длины конфузурной и диффузурной частей равны 5 и 7 мм, соответственно, и общая длина сопла равна $L_{noz} = 12$ мм. Для получения корректного распределения параметров в сопле рассчитывалась также часть пространства за соплом. Радиус свободного объема за соплом составляет $R_{ves} = 25$ мм, а длина $L_{ves} = 20$ мм.

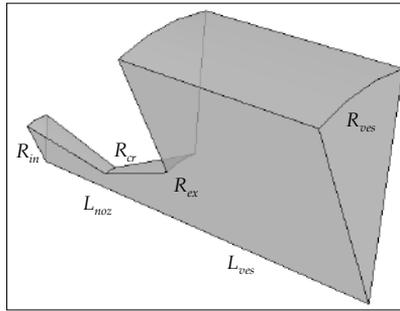


Рис. 1. Схема расчетной области
Fig. 1. Sketch of the calculation domain

Математическое моделирование проводилось на основе численного решения полной системы уравнений Навье-Стокса, дополненной уравнением массовой доли конденсата и слагаемыми в уравнении сохранения энергии, отвечающими за теплоту фазового перехода. Система уравнений, соответствующая законам сохранения массы, импульса, энергии и массовой доли конденсата записывается следующим образом [10, 14]:

$$\frac{\partial Q}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_r}{\partial r} + F_{\theta\theta} = \frac{\partial L_x}{\partial x} + \frac{\partial L_r}{\partial r} + L_{\theta\theta} + L_{cond}, \quad (1)$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \\ \rho G \end{bmatrix}, \quad F_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uh \\ \rho uG \end{bmatrix}, \quad F_r = \begin{bmatrix} \rho v \\ \rho rv \\ \rho v^2 + p \\ \rho vh \\ \rho vG \end{bmatrix}, \quad F_{\theta\theta} = \frac{\rho}{r} \begin{bmatrix} v \\ uv \\ v^2 \\ vh \\ vG \end{bmatrix}, \quad (2)$$

$$L_x = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xr} \\ u\tau_{xx} + v\tau_{xr} + q_x \\ 0 \end{bmatrix}, \quad L_r = \begin{bmatrix} 0 \\ \tau_{rx} \\ \tau_{rr} \\ u\tau_{rx} + v\tau_{rr} + q_r \\ 0 \end{bmatrix},$$

$$L_{\theta\theta} = \frac{1}{r} \begin{bmatrix} 0 \\ 0 \\ \tau_{rr} - \tau_{\theta\theta} \\ u\tau_{rx} + v\tau_{rr} + q_r \\ 0 \end{bmatrix} L_{cond} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m'\Delta H_s \\ m' \end{bmatrix}, \quad (3)$$

где u и v проекции скорости на оси координат, $\tau_{xx}, \tau_{xr}, \tau_{rr}, \tau_{\theta\theta}$ – тензоры скоростей, ρ – плотность газа, e – удельная энергия, h является общей энтальпией, G – массовая доля конденсата, m' – масса сверхкритических кластеров, ΔH_s – тепло фазового перехода, $q = (q_r, q_x)$ – вектор плотности теплового потока.

Система уравнений сохранения (1)-(3) замыкалась соотношениями, выражающими обобщенный закон Ньютона, закон Фурье, энергию и энтальпию газа и уравнением состояния, учитывающим наличие конденсированной фазы в потоке.

$$\tau_{xx} = 2\eta \frac{\partial u}{\partial x} - \frac{2}{3}\eta \operatorname{div} \Omega, \quad \tau_{rr} = 2\eta \frac{\partial v}{\partial r} - \frac{2}{3}\eta \operatorname{div} \Omega, \quad \tau_{\theta\theta} = -\frac{2}{3}\eta \operatorname{div} \Omega, \quad (4)$$

$$\tau_{xr} = \tau_{rx} = \eta \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial r} \right), \quad \operatorname{div} \Omega = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}, \quad \Omega = (u; v),$$

$$e = \frac{u^2 + v^2}{2} + C_v T, \quad h = \frac{u^2 + v^2}{2} + C_p T, \quad (5)$$

$$p = \rho RT(1 - G), \quad (6)$$

где η – коэффициент динамической вязкости (7), C_p – удельная теплоемкость газа, R – универсальная газовая постоянная.

Ввиду больших температурных градиентов коэффициенты динамической вязкости и теплопроводности газа определяется по формулам [17, 18]:

$$\eta(T) = 26.69 \frac{\sqrt{\mu T}}{\sigma^2 \Omega \left(\frac{Tk}{\varepsilon} \right)}, \quad (7)$$

μ является молярной массой газа, σ – потенциальный параметр Леннарда-Джонса, характеризующий расстояние нулевого межмолекулярного взаимодействия, Ω – интеграл столкновения (8), зависящий от закона взаимодействия.

$$\Omega(\bar{T}) = \frac{1.16}{\bar{T}^{0.15}} + \frac{0.52}{e^{0.77\bar{T}}} + \frac{2.16}{e^{2.44\bar{T}}}, \quad (8)$$

\bar{T} является безразмерной температурой, которая может быть определена энергетическим параметром потенциала Леннарда-Джонса и постоянной Больцмана $k = 1.38 \cdot 10^{-23}$ J/K.

$$\lambda(T) = \frac{\eta(T)C_p}{Pr}, \quad (9)$$

λ – коэффициент теплопроводности, определяющийся коэффициентом вязкости, Pr – число Прандтля.

Система уравнений (1)-(9) дополнялась уравнениями, описывающими процесс формирования и роста конденсированных частиц в переохлажденном потоке криптона. Подробная математическая модель данного процесса описана в работе [14]. Данный подход ранее был успешно применен для описания недорасширенных струй криптона за микросоплами. Результаты расчетов согласуются с экспериментальными данными [15].

Задача решается методом контрольных объемов [19] на основе авторского кода. Для расчетной области, показанной на рис. 1 построена блочно-структурированная конечно-объемная сетка, заполненная четырехугольными элементами, которые в засопловом

пространстве принимают форму прямоугольников (рис. 2). Сетка в блоках 1 и 2 равномерная, без сгущения, в блоке 3 шаг расчетной сетки увеличивается по мере удаления от нижней границы. Общее количество сеточных элементов составляет 355 тысяч ячеек. Сеточная сходимость исследовалась на задаче движения вязкого теплопроводного газа без учета эффектов конденсации на основе сравнения графиков распределения температуры, скорости и давления на оси сопла с соответствующими распределениями, характерными для течения невязкого и нетеплопроводного газа. В применяемом подходе контрольные объемы совпадают с элементами сетки. Основные переменные определяются в центральных точках ячеек.

Нахождение конвективного и диффузионного потоков на границе контрольной ячейки осуществляется при помощи линейной аппроксимации решения в каждой ячейке. Величина конвективного потока через границу ячейки определяется при помощи противопоточной схемы. Определение значений параметров потока в центрах ячеек позволяет избежать проблемы деления на ноль при реализации численного алгоритма для уравнений (1) – (9), так как в этом случае значение радиуса не обращается в ноль. Для интегрирования уравнений по времени использовался метод Рунге-Кутты. Шаг по времени, используемый для расчета газовой фазы составляет 10^{-10} с, для расчета течения газа с конденсатом – 10^{-12} с. Расчет одного варианта на ПК (процессор AMD Ryzen 5 1600, оперативная память 64 Гб) занимает около 500 часов.

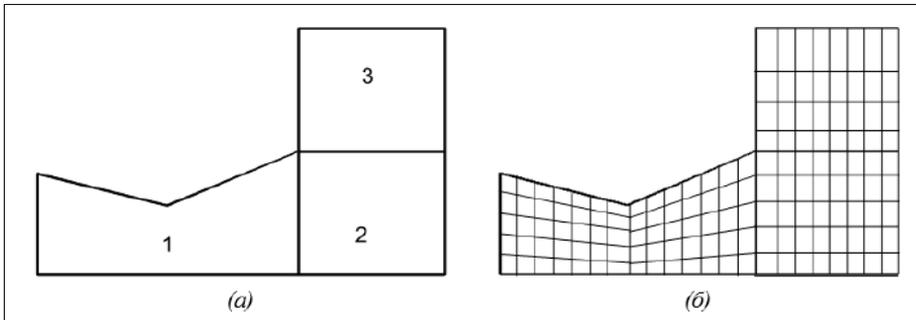


Рис. 2. Блоки расчетной сетки (а) и схема сеточных линий (б)
Fig. 2. Calculation mesh blocks (a) and sketch of the mesh lines (b)

Расчетная область ограничена четырьмя видами границ: входная граница в сопло, непроницаемые стенки, ось симметрии и граница выхода в камеру с низким давлением. На входной границе задавались температура торможения $T = 300$ К и давление торможения газа p_0 , которое задавалось равным 5, 10 и 15 атм. Стенки сопла и вертикальные стенки вакуумной камеры за ним – непроницаемые адиабатические стенки, на которых выполняется условие прилипания потока. В выходных сечениях в случае дозвукового истечения поддерживается постоянное давление 10^{-2} Па, в случае сверхзвукового истечения реализуется снос параметров потока. В начальный момент времени газ считается неподвижным, по всей области интегрирования задаются параметры торможения. Термодинамические свойства криптона определяются в соответствии с [18, 19]: $C_p = 246$ Дж/(кг·К), $C_v = 147$ Дж/(кг·К), $R = 99$ Дж/(кг·К), $\gamma = \frac{C_p}{C_v} = 1.67$, $\sigma = 3.542$ Å, $\varepsilon/k = 178.9$ К, $Pr = 0.7$.

3. Результаты численного моделирования

Численное моделирование проводилось поэтапно. На первом этапе методом установления рассчитывались параметры течения криптона без учета конденсированной фазы. Распределение числа Маха и температуры газа представлены на рис. 3. Слева сверху более детально показана картина течения в расширяющейся части сопла и на выходе из него. На рисунках белая линия отделяет дозвуковую часть потока от сверхзвуковой. Можно увидеть,

что вблизи стенки сопла из-за наличия вязкого трения, формируется достаточно широкая зона дозвуковых скоростей, которая увеличивается вверх по потоку. В самом широком месте эта зона занимает 17.7% от радиального размера потока. Температурное поле соответствует энергетическому распределению. В пристеночной области течения температура приближается к температуре торможения, а по мере приближения к оси канала – к идеальной кривой невязкого течения. За соплом по краю недорасширенной сверхзвуковой струи формируется область повышенной температуры. В струе температура опускается ниже 10 К. В результате числа Маха в струе достигают значений выше 22. Следует отметить, что число Рейнольдса, рассчитанные по средней скорости потока в горловине сопла не превышают величину 2190, а максимальное число Кнудсена равно $9.4 \cdot 10^{-4}$. Данные числа подтверждают применимость ламинарной модели течения и правомерность использования уравнений Навье-Стокса для описания течения в исследуемом микросопле.

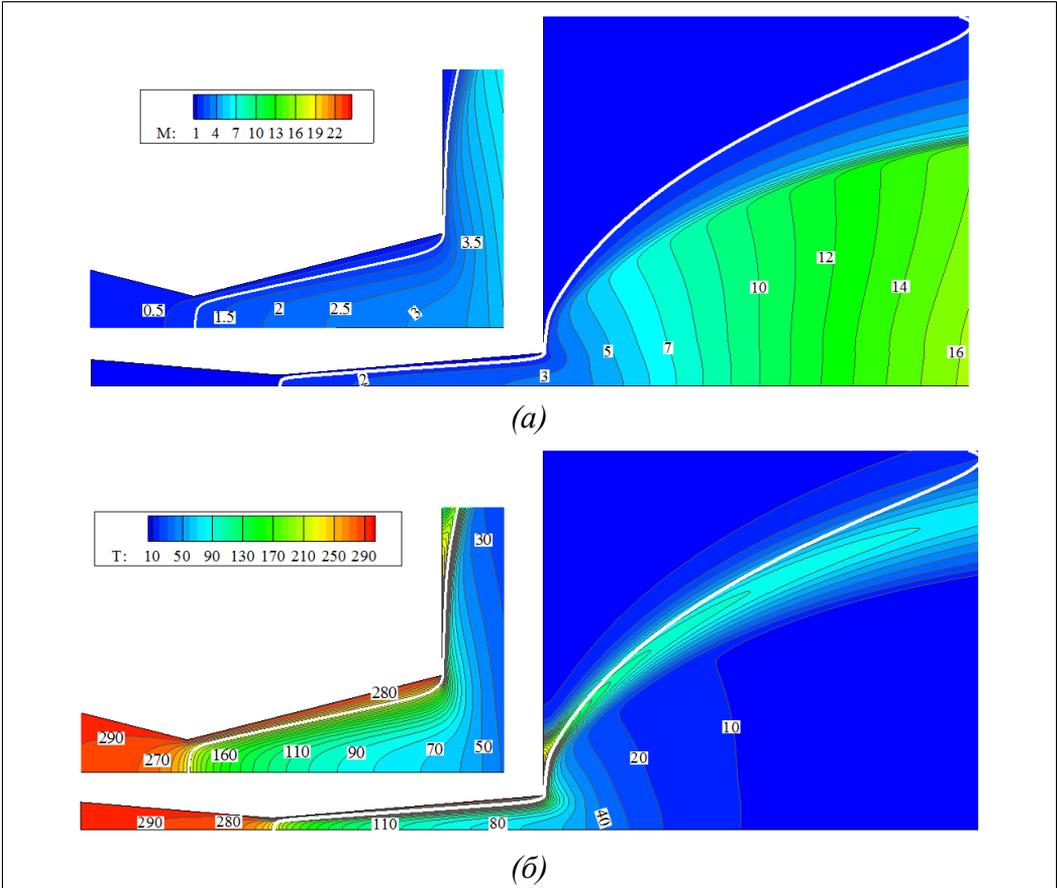


Рис. 3. Распределение числа Маха (а) и температуры (б) однофазного потока
 Fig. 3. Mach number distribution (a) and temperature field (b) for single phase flow

Как и в случае невязкого, нетеплопроводного газа, течения при различных давлениях торможения оказались автомодельными - поля безразмерных давлений (отнесенных к давлению торможения) совпали между собой. При этом, поля скорости и температуры были абсолютно идентичными для различных p_0 . Это можно увидеть на рис. 4., на котором показано изменение давления, температуры и числа Маха вдоль оси расчетной области.

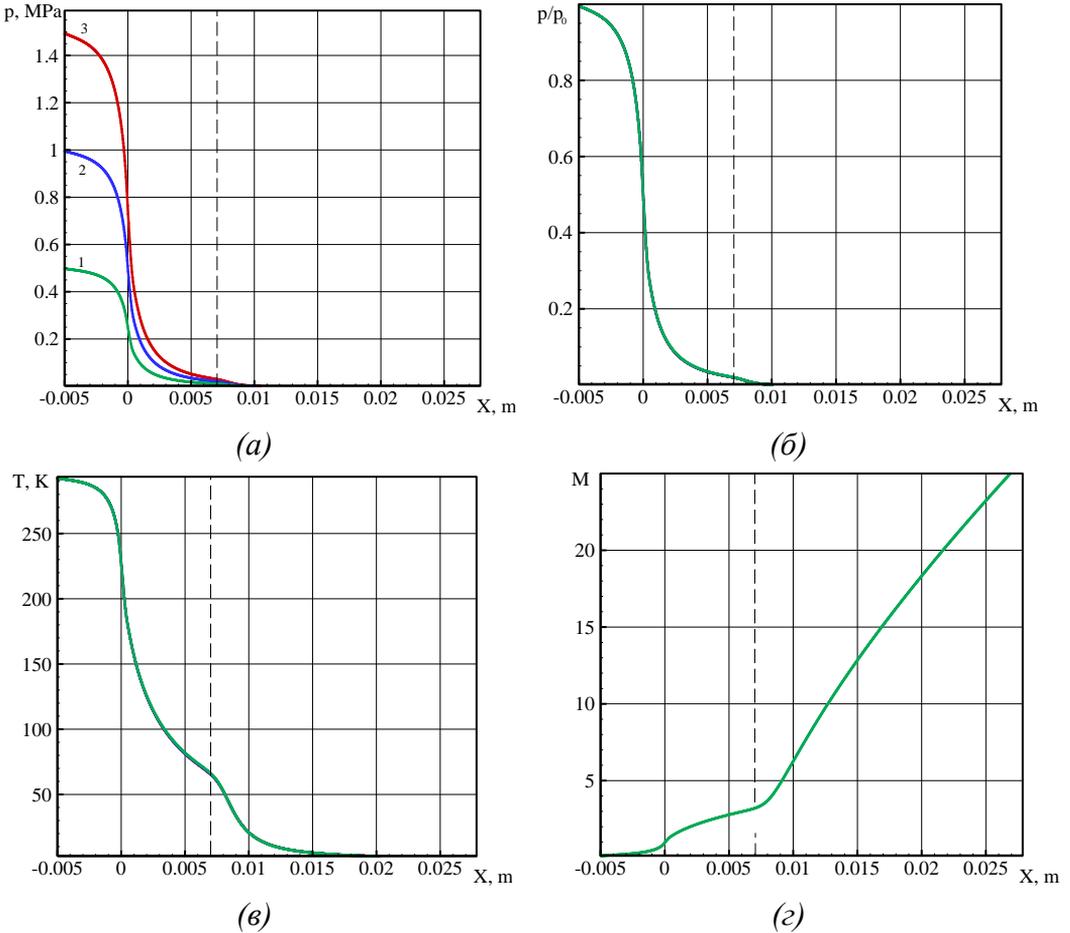


Рис. 4. Изменение статического (а) и относительного (б) давления, температуры (в) и числа Маха (г) вдоль оси
 Fig. 4. Variation of static (a) and dimensionless (b) pressure, temperature (c) and Mach number (d) along axis

На рис.4а приведено распределение осевого давления в МПа. Обозначения на данном рисунке соответствуют различным давлениям торможения в предсопловом объеме: 1 – 5 атм, 2 – 10 атм, 3 – 15 атм (на всех рисунках). На рис.4 б показано уже давление, отнесенное к давлению торможения. Кривые для различных расчетов совпадают между собой. На всех графиках вертикальная пунктирная линия отмечает срез сопла.

Автомодельность течения нарушается, когда на втором этапе исследования подключается модель формирования и роста конденсированных частиц. Данный этап также рассчитывается методом установления. Результаты моделирования двухфазных течений показаны на рис.5, 6.

Появление частиц второй фазы приводит к запуску процесса тепловыделения, а тот в свою очередь к повышению температуры. Рис. 5 соответствует давлению торможения равному 15 атм. Лучше всего этот эффект прослеживается на поле температур.

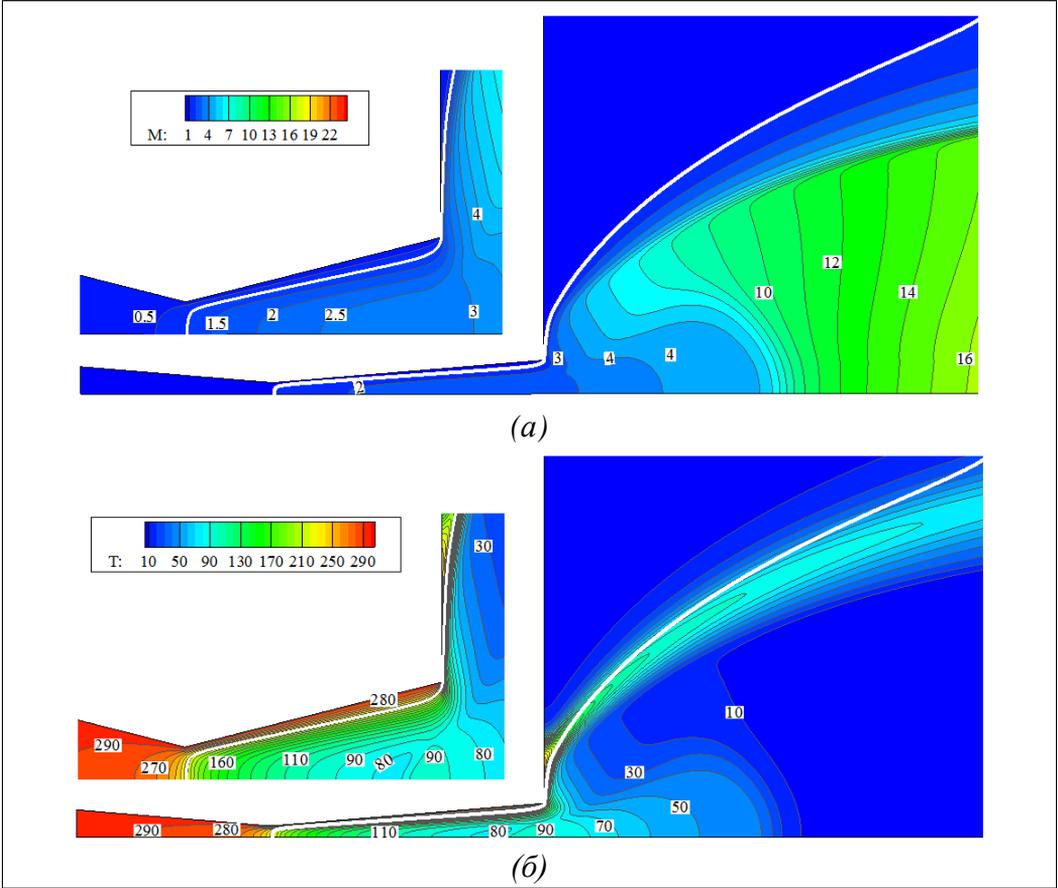


Рис. 5. Распределение числа Маха (а) и температуры (б) двухфазного потока
 Fig. 5. Mach number distribution (a) and temperature field (b) for two phase flow

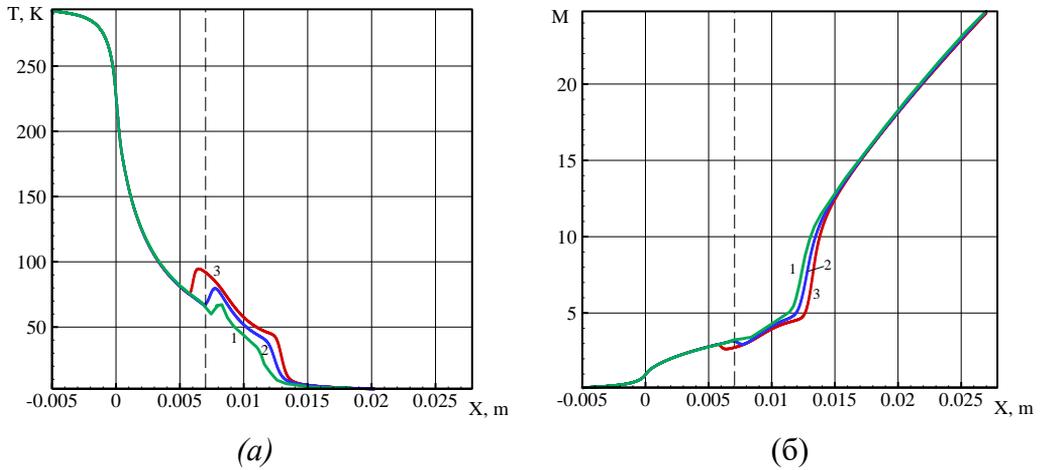


Рис. 6. Изменение температуры (а) и числа Маха (б) двухфазного потока вдоль оси
 Fig. 6. Variation of temperature (a) and Mach number (b) for two phase flow along axis

На выходе из сопла между областями с температурой 80 К появляется локальная зона прогретая до 90 К. В этой области поток замедляется. Лучше всего это видно на графиках на

рис. 6. Параметры локальной области повышения температуры зависят от давления торможения. Это связано, прежде всего, с тем, что разным значениям p_0 соответствуют разные значения температуры, при которых криптон начинает конденсироваться. При давлении торможения 5 атм, стартовая температура конденсации равна 60 К, а при давлении 15 атм – около 75 К. Интенсивность тепловыделения также разная. Большее давление торможения приводит к более интенсивному прогреву газа в зоне конденсации. Процесс конденсации прекращается в одной и той же точке для всех вариантов, в точке $X = 0.015$ м. В результате неравномерного распределения температуры в этих зонах происходит деформация кривой числа Маха (рис.6б). При давлениях торможения 10 и 15 атм хорошо видны зоны торможения. При давлении 5 атм явного торможения не наблюдается, но разгоняется поток гораздо медленнее, чем в однофазном случае.

4. Заключение

В работе проведены параметрические исследования течения криптона в коническом микросопле при истечении в область с низким давлением. Изучены особенности течений в при различных значениях давления торможения в предсопловом объеме, в том числе при возникновении конденсированной фазы в потоке. Исследования проведены при $p_0 = 5, 10$ и 15 атм. Показано, что в случае однофазного потока наблюдается автомодельность течений газа. При этом, поля скорости и температуры являются абсолютно идентичными для различных значений давления торможения. Автомодельность течения нарушается, когда подключается модель формирования и роста конденсированных частиц. Размеры зон локального повышения температур, а также интенсивность тепловыделения зависят от заданного давления торможения. Это отражается на скоростных характеристиках течения.

Список литературы / References

- [1] Родченкова В., Шахнович И. Микрофлюидные чипы – конструктор для разработчика. Решения компании Dolomite. Аналитика, вып. 3, 2017 г., стр. 60-69 / Rodchenkova V., Shakhnovich. I. Microfluidic chips are a designer for developers. Dolomite solutions. Analytics, issue 3, 2017, pp. 60-69 (in Russian).
- [2] Зими́на Т.М. Миниатюрные аналитические системы биомедицинского назначения – лаборатории на чипе. Биотехносфера, вып. 1, 2009 г., стр. 11-17 / Zimina T.M. Miniature analytical systems for biomedical purposes – laboratories on a chip. Biotechnosphere, issue 1, 2009, pp. 11-17 (in Russian).
- [3] Александров В.А., Тонков Л.Е. Экспериментальная микрогидродинамика капли жидкости на вибрирующей полимерной подложке. Химическая физика и мезоскопия, том 18, вып. 4, 2016 г., стр. 620-633 / Aleksandrov V.A., Tonkov L.E. Experimental microhydrodynamics of liquid drop on a vibrating polymeric substrate. Chemical Physics and Mesoscopics, vol. 18, issue 4, 2016, pp. 620–633 (in Russian).
- [4] Лемозерский В.Е., Зими́на Т.М. и др. Разработка интегрируемого в микрофлюидную систему электроакустического актюаторного модуля для экспресс-подготовки фракций крови. Биотехносфера, вып. 3, 2017 г., стр. 16-28 / Lemozersky V.E., Zimina T.M. et al. Development of an electroacoustic actuator module integrated into a microfluidic system for express preparation of blood fractions. Biotechnosphere, issue 3, 2017, pp. 16-28 (in Russian).
- [5] Ахметов А.Т., Валиев А.А. и др. Микрогидродинамика крови при стенозе сосудов Труды Института механики им. Р.Р. Мавлютова, том 11, вып. 2, 2016 г., стр. 210-217 / Akhmetov A.T., Valiev A.A. et al. Microfluidics of blood in blood vessels stenosis. Proceedings of the Mavlyutov Institute of Mechanics, vol. 11, issue 2, 2016, pp. 210-217 (in Russian).
- [6] Балабанов А.В., Касимов А.М. Разработка и исследование рабочих характеристик микроструйного генератора Датчики и системы, вып. 7-8, 2019 г., стр. 34-40 / Balabanov A. V., Kasimov A. M. Development and investigation tests of microfluidic generator. Sensors and Systems, issue 7-8, 2019, pp. 34-40 (in Russian).
- [7] Солнышкина О.А., Батыршин Э.С., Питюк Ю.А. Исследование гидродинамических потоков в микромоделях сред с двойной пористостью. Известия Российской академии наук. Механика жидкости и газа, вып. 4, 2021 г., стр. 9-18 / Solnyshkina O.A., Bатыrshin E.S., Pityuk Y.A. Investigation

- of hydrodynamic flows in micromodels of double porosity media. *Fluid Dynamics*, vol. 56, issue 4, pp. 451-459.
- [8] Гаряев А.Б., Прун О.Е., Клименко А.В. Определение оптимального соотношения характеристик микроканальных теплообменных аппаратов. *Теплофизика и аэромеханика*, том 22, вып. 6, 2015 г., стр. 751-760 / Garyaev A.B., Prun O.E., Klimentko A.V. Evaluation of Optimal Thermal-Hydraulic Characteristics Ratio in Microchannel Heat Exchangers. *Thermophysics and Aeromechanics*, vol. 22, issue 6, 2015, pp. 723-732 (in Russian).
- [9] Филимонов С.А., Дектерев А.А. и др. Моделирование сопряженного теплообмена в системе микроканалов при помощи гибридного алгоритма. *Сибирский журнал индустриальной математики*, том 18, вып. 3, 2015 г., стр. 86-97 / Filimonov S.A., Dekterev A.A. Simulation of conjugate heat transfer in a microchannel system by a hybrid algorithm. *Journal of Applied and Industrial Mathematics*, vol. 9, issue 4, 2015, pp. 469-479 (in Russian).
- [10] Korepanov M.A., Koroleva M.R., Mitrukova E.A. Numerical Investigation of Flows with Condensation in Micronozzles. *Journal of Physics: Conference Series*, vol. 2057, 2021, article no. 012016, 6 pp.
- [11] Barrot C., Colin S. Design of Tree-Shaped Microchannel Networks Submitted to Simultaneous Pressure Driven and Electro-Osmotic Flows. In Proc. of the 10th International Conference on Nanochannels, Microchannels and Minichannels, 2012, pp. 113-121.
- [12] Lobasov A.S., Shebeleva A.A., Minakov A.V. The Study of Ethanol and Water Mixing Modes in the T-shaped Micromixers. *Journal of Siberian Federal University. Mathematics & Physics*, vol. 12, issue 2, 2019, pp. 202-212.
- [13] Shershnev A.A., Kudryavtsev A.N. Numerical simulation of particle beam focusing in a supersonic nozzle with rectangular cross-section. *Journal of Physics. Conference Series*, vol. 1404, 2019, article no. 012042, 4 p.
- [14] Korepanov M.A., Koroleva M.R. et al. Nonlinear Effects of Krypton Flow in a Micronozzle with a Cylindrical Tube. *Russian Journal of Nonlinear Dynamics*, vol. 18, issue 3, 2022, pp. 411-422.
- [15] Nechay A.N., Perekalov A.A. et al. Emission properties of targets based on shock waves excited by pulsed laser radiation. *Optics & Laser Technology*, vol. 142, 2021, article no. 107250.
- [16] Корепанов М.А., Морар Г., Альес М.Ю. Моделирование гомогенной конденсации криптона в сверхзвуковом сопле. *Химическая физика и мезоскопия*, том 22, вып. 2, 2020 г., стр. 155-163 / Korepanov M.A., Morar G., Alies M.Yu. Modeling of Homogeneous Condensation of Krypton in the Supersonic Nozzle. *Chemical physics and mesoscopy*, vol. 22, issue 2, 2020, pp. 155-163 (in Russian).
- [17] Рабинович В.А., Вассерман А.А. и др. Теплофизические свойства неона, аргона, криптона и ксенона. М., Изд-во стандартов, 1975 г., 636 стр. / Rabinovich V.A., Vasserman A.A. et al. Thermophysical properties of neon, argon, krypton and xenon. М., Publishing house of standards, 1975, 636 p. (in Russian).
- [18] R.C. Reid, Prausnitz J.M., Sherwood T.K. *The Properties of Gases and Liquids*. McGraw-Hill, 1977, 688 p.
- [19] Chernova A.A. Validation of RANS turbulence models for the conjugate heat exchange problem. *Russian Journal of Nonlinear Dynamics*, vol. 18, issue 1, 2022, pp. 61-82.

Информация об авторах / Information about authors

Екатерина Александровна МИТРЮКОВА – аспирантка кафедры «Тепловые двигатели и установки». Сфера научных интересов: математическое моделирование, газовая динамика.

Ekaterina Alexandrovna MITRYUKOVA is a post-graduate student of the Department of Heat Engines and Installations. Research interests: mathematical modeling, gas dynamics.

DOI: 10.15514/ISPRAS-2023-35(2)-17



Модификация алгоритма выравнивания коротких прочтений для повышения качества пайплайна обработки данных полногеномного секвенирования человека

¹ Е.П.Гугучкин, ORCID: 0000-0001-7885-9892 <guguchkin@ispras.ru>

¹ Е.А.Карпулевич, ORCID: 0000-0002-6771-2163 <karpulevich@ispras.ru>

¹ Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

Аннотация. Данное исследование подчеркивает важность выравнивания коротких прочтений (ридов) в анализе данных полногеномного секвенирования человека. Процесс выравнивания состоит в определении позиций коротких генетических последовательностей относительно заранее известной референсной последовательности генома человека. Традиционные методы выравнивания используют линейную референсную последовательность, но это может привести к некорректному выравниванию, особенно если в рядах присутствуют генетические варианты. В данной работе была проведена модификация индексного файла референсной последовательности инструмента minimap2. В результате экспериментов было показано, что добавление в индекс инструмента minimap2 информации о часто встречающихся генетических вариантах приводит к повышению количества верно выявленных генетических вариантов, что влияет на качество последующего анализа данных.

Ключевые слова: конвейер обработки данных, секвенирование ДНК, вычислительная биология, методы выравнивания последовательностей, анализ данных NGS, вычислительные методы

Для цитирования: Гугучкин Е.П., Карпулевич Е.А. Модификация алгоритма выравнивания коротких прочтений для повышения качества пайплайна обработки данных полногеномного секвенирования человека. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 235–248. DOI: 10.15514/ISPRAS–2023–35(2)–17

Modification of the short read alignment algorithm to improve the quality of the human whole genome sequencing data processing pipeline

¹ E.P. Guguchkin ORCID: 0000-0001-7885-9892 <guguchkin@ispras.ru>

¹ E.A. Karpulevich ORCID: 0000-0002-6771-2163 <karpulevich@ispras.ru>

¹ Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. This study emphasizes the importance of aligning short reads in the analysis of human whole-genome sequencing data. The alignment process involves determining the positions of short genetic sequences relative to a known reference genome sequence of the human genome. Traditional alignment methods use a linear reference sequence, but this can lead to incorrect alignment, especially when short reads contain genetic variations. In this work, the index file of the reference sequence was modified using the minimap2 tool. Experimental results showed that adding information about frequently occurring genetic variations to the

minimap2 index increases the number of correctly identified genetic variants, which affects the quality of subsequent data analysis.

Keywords: data processing pipeline, DNA sequencing, Computational biology, Sequence alignment methods, NGS data analysis, Computational methods

For citation: Guguchkin E.P., Karpulevich E.A. Modification of the short read alignment algorithm to improve the quality of the human whole genome sequencing data processing pipeline. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 2, 2023. pp. 235-248 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-17

1. Введение

Геном человека представляет собой последовательность четырех нуклеотидных оснований, обозначаемых как А(аденин), С(цитозин), G(гуанин) и Т(тимин). В последовательности генома человека содержится более трех миллиардов нуклеотидов. В геноме содержится генетическая информация, необходимая для поддержания правильной работы живого организма. Исследования генома имеют множество научных и практических применений, включая идентификацию генетических вариантов, связанных с болезнями [1], разработку персонализированных лекарств, исследование эволюционных процессов и биологического разнообразия. Современные технологии в области биологии, такие как редактирование генома и синтетическая биология, появившиеся в результате развития исследований в области генетики, обладают большим потенциалом в развитии медицины, сельского хозяйства и промышленности [2][3][4]. Полногеномное секвенирование (Whole Genome Sequencing) — это технология определения полной последовательности ДНК генома организма [5].

Процесс полногеномного секвенирования происходит в несколько этапов: образец начиная с лабораторного процесса заканчивая вычислительным анализом полученных данных. ДНК извлекается из биологического образца, такого как кровь или ткань, а затем фрагментируется и подготавливается для проведения секвенирования. Затем фрагменты ДНК секвенируют с использованием технологий высокопроизводительного секвенирования. Результатом процесса секвенирования следующего поколения (Next-Generation Sequencing, NGS) [6] является файл, который содержит миллионы коротких подпоследовательностей ДНК, называемых ридами или короткими прочтениями.

Длина рида зависит от конкретной технологии секвенирования, но обычно составляет от 50 до 250 нуклеотидов. Также в зависимости от технологии риды могут быть одиночными (single-end) или парными (paired-end). Одиночные риды — это риды, которые генерируются только с одного конца секвенируемого фрагмента ДНК. Риды с парными концами являются результатом секвенирования обоих концов фрагмента ДНК.

Анализ данных полногеномного секвенирования состоит из следующих основных этапов [7]: контроль качества, выравнивание ридов на референсный геном (известная последовательность нуклеотидов, которая представляет собой образец генома какого-либо вида), и поиск генетических вариантов (отличий генома секвенируемого организма от референса). Риды, полученные в результате полногеномного секвенирования проходят проверку качества, в результате которой риды с низким качеством могут быть удалены или обрезаны по краям для повышения точности последующего анализа. После этого, риды выравниваются (Рис. 1) на последовательность из референсного генома: каждой последовательности из рида ставится в соответствие некоторая подпоследовательность из референсной последовательности. Для выровненных ридов происходит поиск несоответствий между последовательностью из ридов и референсной последовательностью, таким образом происходит поиск генетических вариантов. Генетические варианты могут по-разному влиять на характеристики человека такие, как физические черты, риск заболевания или реакция на лекарство. Генетические варианты (Рис. 2) бывают нескольких видов: однонуклеотидные полиморфизмы, инсерции(вставки) и делеции. Однонуклеотидный

полиморфизм (SNP) — это тип генетического варианта, в котором происходит замена одного нуклеотидного основания в определенной позиции последовательности ДНК на другое. SNP является наиболее распространенным типом генетических вариантов в геноме человека и встречается примерно один раз на каждые 300 нуклеотидов. Индел (сокращение от инсерция-делеция) — это тип генетического варианта, в котором происходит вставка или удаление одного или нескольких нуклеотидов в последовательности ДНК относительно референсной. Инделы могут иметь размер от одного до нескольких тысяч нуклеотидов и могут встречаться в любом месте генома.

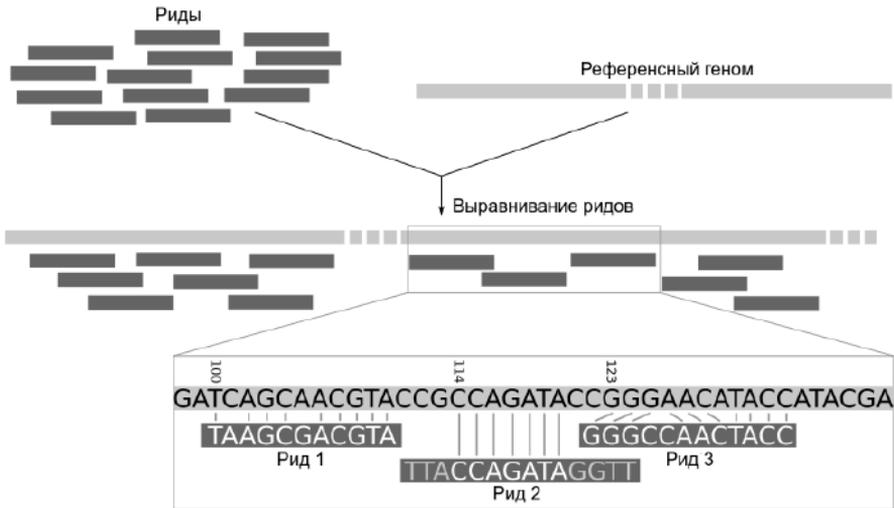


Рис.

1. Выравнивание ридов
Fig. 1 Alignment of short reads

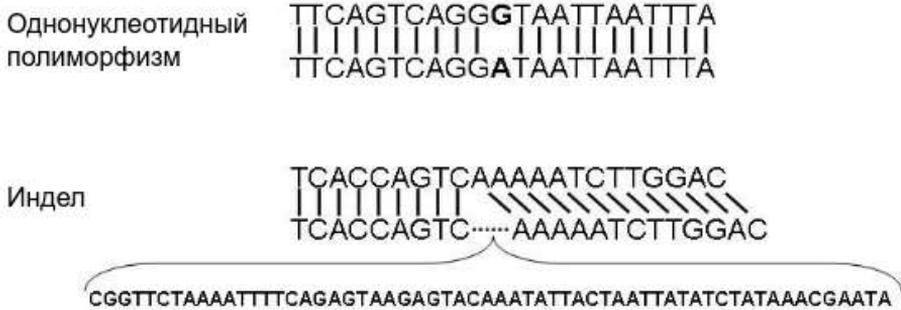


Рис. 2. Однонуклеотидные полиморфизмы и инделы
Fig. 2. SNPs and indels

Генетический вариант может не быть корректно обнаружен, если риды, содержащие этот генетический вариант, не будут выровнены в нужное место на этапе выравнивания. При этом риды, содержащие генетические варианты, если и будут выровнены, то оценка качества выравнивания будет снижена, что также может повлиять на точность нахождения вариантов [8]. В данной работе предложена модификация алгоритма выравнивания ридов для уменьшения количества ложно отрицательных генетических вариантов, с целью снизить негативный эффект, накладываемый при выравнивании ридов с генетическими вариантами.

2. Обзор методов выравнивания ридов на референс

В данном разделе рассмотрены основные методы выравнивания ридов на референсную последовательность. Для точного выравнивания строк существуют такие алгоритмы как алгоритм Смита — Уотермана [9] и алгоритм Нидлмана — Вунша [10]. Различные модификации этих алгоритмов используются в большинстве инструментов выравнивания, однако при выравнивании каждого рида на референсную последовательность длиной несколько миллиардов букв точными алгоритмами выравнивания неэффективно и для ускорения процедуры выравнивания используются дополнительные алгоритмы (на основе методов Seed-and-Extend или Seed-Chain-Align), которые позволяют найти предполагаемые позиции рида в геноме и уже после этого применить алгоритмы точного выравнивания. Помимо алгоритмов выравнивания на линейный референс также существует ряд алгоритмов выравнивания на граф [11], однако они значительно проигрывают по скорости приведенным выше алгоритмам.

2.1 Метод Seed-and-Extend

Основная идея метода Seed-and-Extend заключается в том, чтобы сперва найти короткие точные соответствия (якоря, seeds) между последовательностями ридов и референсной последовательностью, а затем продлить найденные якоря до рида для выполнения выравнивания. Можно выделить следующие этапы данного метода (Рис. 3):

1. **Создание индекса референсной последовательности.** На данном этапе создается индекс референсной последовательности, позволяющий быстро находить определенные регионы последовательности. Примером алгоритма, реализующим создание индекса, является преобразование референса с помощью алгоритма Барроуза — Уилера [12] в суффиксный массив [13].
2. **Вычисление якорей.** Последовательность из рида разбивается на k -меры (подстроки длины k). После этого происходит поиск каждого k -мера в индексе референсной последовательности. В случае если для k -мера из рида существует точное совпадение в референсной последовательности, то такой k -мер становится якорем.
3. **Продление якорей.** Каждый якорь расширяется в обоих направлениях вдоль последовательности из рида и референсной последовательности с использованием алгоритмов динамического программирования (модификации алгоритма Смита — Уотермана), которые ищут лучшее соответствие, начиная с каждой позиции якоря. Расширение проводится до конца последовательности из рида, либо конца референсной последовательности. При оценке выравнивания учитываются такие факторы как несовпадение отдельных элементов последовательностей и индели.
4. **Уточнение выравнивания.** После того, как было выполнено базовое выравнивание, оно может быть дополнительно уточнено для повышения его точности. Уточнение может включать в себя дополнительные итерации выравнивания, а также учет дополнительной информации, например, информации о потенциальных ошибках секвенирования.

5. Фильтрация. Данный этап включает в себя фильтрацию выровненных ридов с низким качеством и удаление неоднозначно выровненных ридов.

Метод выравнивания Seed-and-Extend реализован в таких инструментах, как Bowtie2 [14] и BWA-MEM [15].

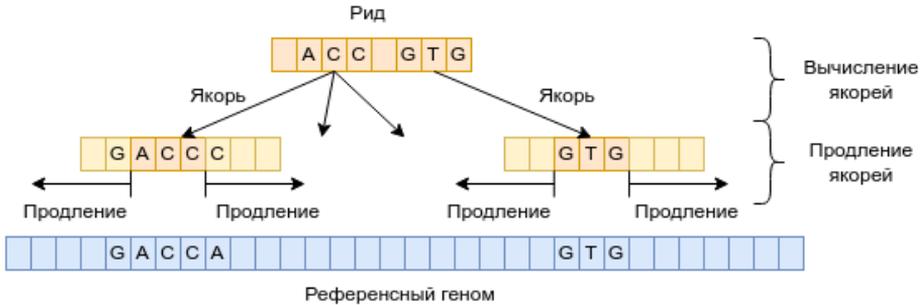


Рис. 3. Выравнивание ридов методом Seed-and-Extend.
Fig. 3. Read alignment using the Seed-and-Extend method.

2.2 Метод Seed-Chain-Align

Несмотря на то, что данный метод имеет определенные сходства с методом seed-and-extend, в нем также имеется ряд отличий, влияющих на качество и производительность. Важным отличием метода Seed-Chain-Align от метода Seed-and-Extend, является использование в процессе выравнивания ридов более коротких k-меров.

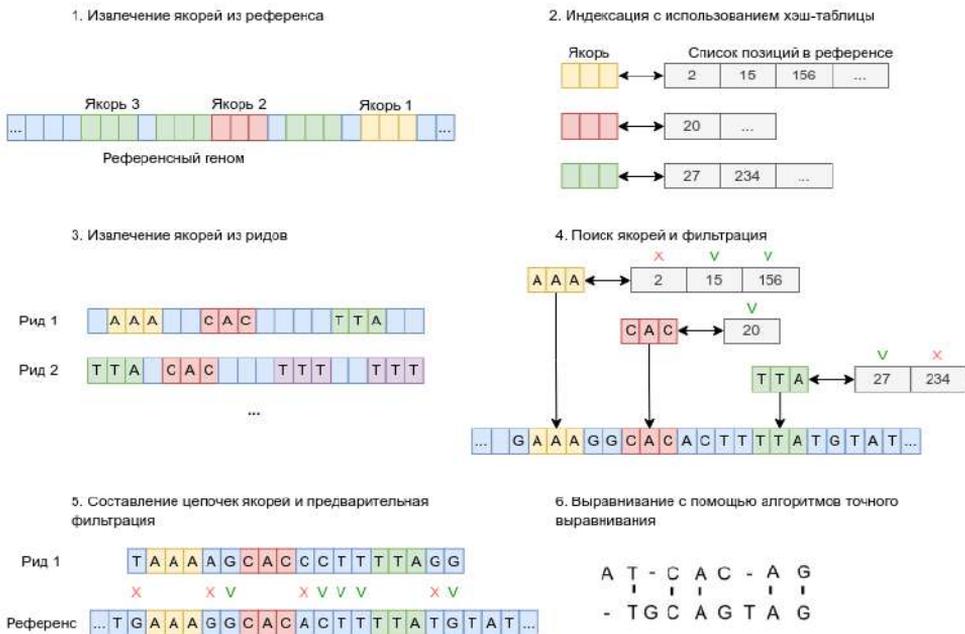


Рис. 4. Выравнивание ридов методом Seed-Chain-Align.
Fig. 4. Read alignment using the Seed-Chain-Align method.

Рассмотрим основные этапы метода Seed-Chain-Align (Рис. 4):

1. **Извлечение якорей из референса.** На данном этапе аналогично методу Seed-and-Extend вычисляются все якоря в референсной последовательности.
2. **Индексация референса.** На данном этапе создается индекс референсной последовательности, вычисленные якоря и их позиции заносятся в хэш-таблицу.
3. **Извлечение якорей из ридов.** На данном этапе вычисляются якоря для ридов.
4. **Поиск якорей и фильтрация.** Выполняется поиск позиций якорей ридов в референсе и фильтрация по различным критериям, например, фильтрация наиболее распространенных якорей
5. **Составление цепочек якорей и предварительная фильтрация.** Для нахождения наиболее точного выравнивания вычисляется оценка оптимальной цепочки якорей. Алгоритм сначала идентифицирует все возможные цепочки совпадающих k -меров между последовательностями из прочтений и референсной последовательности, а затем вычисляет оценку для каждой цепочки на основе количества совпадающих k -меров и размера разрывов между ними. После этого алгоритм выбирает цепочку с наибольшей оценкой в качестве основной.
6. **Выравнивание с помощью алгоритмов точного выравнивания.** На данном этапе проводится глобальное выравнивание между полученными последовательностями из якорей на референсной последовательности и якорей последовательности из рида. Глобальное выравнивание может производиться, например, с помощью алгоритма Нидлмана — Вунша.

Метод Seed-Chain-Align реализован в инструменте выравнивания minimap2 [16]. В случае с инструментом minimap2 в качестве якорей используются так называемые минимизаторы. Минимизатор — это короткая подстрока длины k , которая является лексикографически минимальной строкой в окне w (Рис. 5).

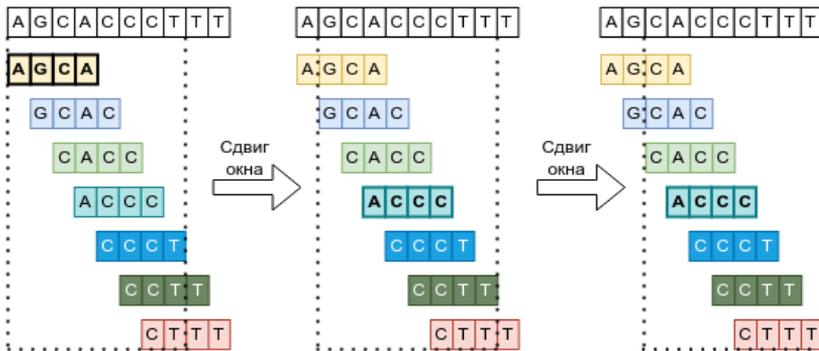


Рис. 5. Поиск минимизаторов длины $k=4$ в окне длины $w=5$.
Fig. 5. Finding minimizers of length $k=4$ in a window of length $w=5$.

2.3 Выводы

Оба метода (Seed-Chain-Align и Seed-and-Extend) выполняют дополнительные шаги обработки последовательностей перед их точным выравниванием. Данные методы значительно ускоряют выравнивание ридов на линейный референс, однако в случае если в якоря ридов который прочитан с определенной позиции в геноме присутствует генетический вариант — данный рид может ошибочно выровняться в другую позицию в геноме или выровняться в правильную позицию, но с меньшим качеством. Если дополнительно учитывать якоря, содержащие генетические варианты, то возможно получится увеличить качество существующего алгоритма выравнивания. Причем индекс в виде хэш-таблицы, в

отличие от индекса в формате суффиксного массива, удобно модифицировать добавлением в таблицу новых позиций якорей с генетическими вариантами.

3. Исследование и построение решения задачи

Выше были рассмотрены основные методы выравнивания ридов и была предложена идея провести модификацию алгоритма выравнивания метода Seed-Chain-Align, реализованного в инструменте выравнивания `minimap2`.

В данном разделе будут рассмотрены шаги, необходимые для проведения модификации алгоритма выравнивания данных полногеномного секвенирования человека. Помимо самой модификации необходимо провести эксперименты для оценки качества результатов, полученных в результате применения разработанной модификации.

3.1 Модификация алгоритма выравнивания `minimap2`

Модификация алгоритма выравнивания (Рис. 6) заключается в том что на этапе поиска якорей `minimap2` идентифицирует короткие последовательности — минимизаторы в ридах и референсе с добавлением генетических вариантов. В данной работе модификация выполняется добавлением однонуклеотидных полиморфизмов (SNP) в референсную последовательность. На этапе поиска оптимальных цепочек `minimap2` учитывает наличие генетических вариантов таких как SNP в референсной последовательности. В частности, `minimap2` использует систему подсчета очков, которая учитывает вероятность данного выравнивания на основе наличия генетических вариантов. Если рид содержит генетический вариант, отсутствующий в референсном геноме, `minimap2` все еще может выравнивать данный рид на референсную последовательность, допуская разрывы или несоответствия в выравнивании. Однако наличие генетического варианта может негативно повлиять на итоговую оценку качества выравнивания, что в дальнейшем приведет к ухудшению поиска генетических вариантов в данном регионе последовательности. Особенно заметным будет эффект, в случае если генетический вариант будет расположен в позиции потенциального якоря, что приведет к ухудшению оценки качества составления цепочки и возможному выбору другой цепочки вместо необходимой.

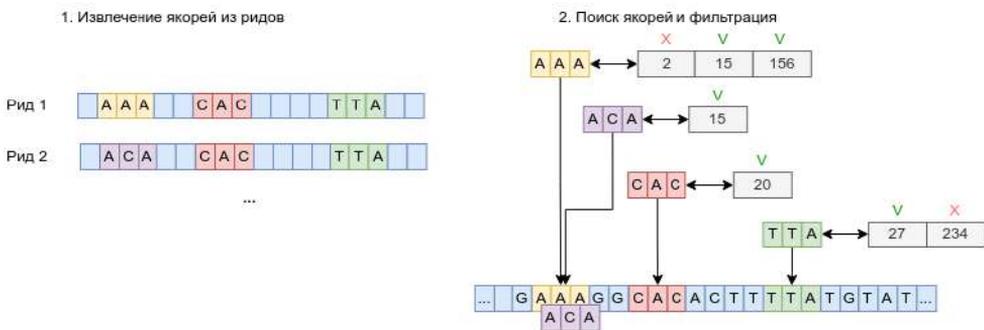


Рис. 6. Поиск якорей в модифицированном индексе
Fig. 6. Finding seeds in a modified index

Добавление часто встречающихся генетических вариантов в референсную последовательность может быть реализовано с помощью дополнительной модификации индекса референсной последовательности, содержащий хэш-таблицу из результатов хэш-функции от минимизаторов и их позиций в референсной последовательности.

В исходном коде инструмента `minimap2` содержится функция `mm_sketch` (файл `sketch.c`), которая принимает на вход последовательность нуклеотидов и строит все возможные

минимизаторы длины k в окне длины w . Функция возвращает массив, содержащий пары ключ-значение: соответствующие результаты хэш-функции от последовательности и ее позиции в последовательности. Организация хэш-таблицы реализована с использованием библиотеки `klib`.

Для модификации индекса необходимо подготовить короткие подпоследовательности референсной последовательности, добавить в них генетический вариант (в случае добавления SNP как в данной работе — заменить символ), заново вычислить минимизаторы в модифицированной последовательности, сопоставить позиции в подпоследовательностях и референсной последовательности и, с помощью методов из `klib`, дополнить существующую хэш-таблицу новыми минимизаторами. В результате, два ряда, последовательность одного из которых не содержит генетический вариант в якоре, а второго содержит, будут выравниваться в одинаковую позицию референсной последовательности и будут иметь одинаковую оценку качества выравнивания.

3.2 План экспериментов анализа данных полногеномного секвенирования человека

Для оценки эффекта, получаемого при модификации индекса, необходимо провести ряд экспериментов анализа данных полногеномного секвенирования. Это подразумевает проведение нескольких различных этапов, а именно: выравнивание ридов, поиск дубликатов, вычисление таблицы повторной калибровки, поиск генетических вариантов. Все этапы стоит провести в соответствии с рекомендациями GATK Best Practices [17].

В качестве первого эксперимента планируется взять несколько наборов реальных данных, полученных с помощью технологии секвенирования следующего поколения (NGS). Необходимо взять данные с разным покрытием для одного образца. Под покрытием подразумевается среднее количество ридов для каждого участка геномной последовательности. Разное покрытие позволит судить об эффекте, полученном при модификации индекса.

Соответственно, для каждого набора данных необходимо провести две итерации анализа данных полногеномного секвенирования:

- 1) Анализ данных полногеномного секвенирования, при котором на этапе выравнивания ридов используется немодифицированный индекс референсной последовательности;
- 2) Анализ данных полногеномного секвенирования, в котором используется модифицированный индекс с добавлением однонуклеотидных полиморфизмов;

Во втором эксперименте планируется использовать несколько наборов синтетических данных. Синтетические риды — это сгенерированные компьютером последовательности, которые предназначены для имитации характеристик реальных ридов, включая длину считывания, частоту ошибок секвенирования и наличие генетической изменчивости. Аналогично первому эксперименту планируется провести несколько итераций анализа данных полногеномного секвенирования.

4. Описание практической части

4.1 Описание выполненной модификации

Для модификации алгоритма выравнивания была написана функция `mm_idx_manipulate`, которая вызывается после создания индекса референсной последовательности. Данная функция считывает строки из указанного VCF файла [18] и для каждого генетического варианта (SNP), записанного в отдельной строке, проводятся следующие шаги:

1. В отдельные переменные записываются следующие данные: номер хромосомы, позиция на хромосоме, значение нуклеотидов в референсной последовательности и генетический вариант.
2. Из референсной последовательности (которая также хранится в индексном файле) считывается $2 * (k + w) - 1$ символов последовательности, где центральный символ — это генетический вариант.
3. Вызывается функция `mm_sketch`, которая вычисляет минимизаторы полученной последовательности.
4. Полученные минимизаторы фильтруются, у оставшихся минимизаторов корректируется номера позиции в последовательности.
5. Вызывается написанная функция `mm_idx_push`, которая добавляет новые минимизаторы в индексную хэш-таблицу.

Полученная модификация имеет линейную вычислительную сложность с незначительными дополнительными затратами памяти порядка $O(n)$ при вычислении каждого генетического варианта. Затраты на добавление элементов в хэш-таблицу можно оценить в среднем $O(1)$ по времени и $O(n)$ по памяти. Время работы алгоритма выравнивания и затраты память при модификации индекса увеличиваются на $o(m)$, где m — длина референсной последовательности. Также, стоит добавить, что вычислять индекс референсной последовательности нужно всего один раз.

4.2 Обработка и обзор данных

Референсный геном

В ходе данной работы использовалась геномная референсная последовательность GRCh38, выпущенная Genome Reference Consortium в 2013 году [19]. Данная референсная последовательность представлена 23-мя хромосомами (от 1-ой до 22-ой, а также X-хромосома) и состоит приблизительно из 3 миллиардов нуклеотидов.

Данные о генетических вариантах в популяции

Данные о генетических вариантах были взяты из проекта 1000 Genomes Project [20]. Данный проект содержит сведения о генетических вариантах для 2504 человек из 26 популяций. Данные представлены в формате VCF, который является стандартным файловым форматом для анализа генетических данных.

Для обработки этих данных использовался инструмент `bcftools`, который является частью пакета `samtools` [21]. С его помощью были отфильтрованы генетические варианты, которые встречаются у достаточного числа представителей в популяции (`minor allele frequency > 0.05`), а также использовался для удаления генетических вариантов тех людей, для которых проводился анализ данных полногеномного секвенирования человека. Помимо этого, `bcftools` был использован для выделения конкретной популяции из VCF файла.

Данные о ридов

Данные с ридов были взяты из The precisionFDA Truth Challenge — испытании, целью которого было улучшение качества результатов анализа данных полногеномного секвенирования [22]. А также данные, собранные организацией National Institute of Standards and Technology. Датасет состоял из пар файлов с ридов в формате FASTQ [23] для 2 человек: NA12878, NA24631 (сын из набора данных Chinese Trio). Для проведения дальнейших экспериментов были использованы разные наборы данных, отличающихся разным покрытием — средним количеством ридов на всю референсную геномную последовательность. Были использованы данные с покрытием 30X и 60X.

“Уверенные” регионы и эталонные генетические варианты

Из The precisionFDA Truth Challenge были взяты данные об “уверенных” регионах для каждого человека. Эти регионы использовались во время всего пайплайна анализа данных полногеномного секвенирования, а также использовались при валидации с эталонными файлами генетических вариантов (которые были взяты из The precisionFDA Truth Challenge). Стоит отметить, что для уменьшения влияния краевого эффекта на концах “уверенных” регионов, эти регионы были расширены по краям на 1000 нуклеотидов. Расширенные “уверенные” регионы использовались только во время подсчета, во время валидации использовались изначальные версии. Для расширения регионов был написан скрипт `interval_expander.py`. А для последующей конкатенации расширенных пересекающихся регионов — скрипт `interval_concater.py`.

Синтетические данные

Для генерации синтетических данных случайным образом было выбрано 10 образцов из проекта 1000 Genomes Project. Были выбраны образцы: HG00472, HG01205, HG01578, HG01680, HG02252, HG02301, HG02536, HG02603, HG03485, HG03520. Аналогично реальным данным создавались наборы парный ридов с длиной каждой последовательности из рида в 100 нуклеотидов. Данные создавались с покрытием 26X. При создании синтетических данных использовались инструменты `bcftools` и инструмент ART [24]. `bcftools` использовался для создания геномной последовательности, содержащей генетические варианты конкретного образца, по которым инструмент ART создавал наборы парных ридов.

4.3 Метрика оценки качества

Генетические варианты, полученные в результате анализа, можно разделить на три категории:

- найденные верно генетические варианты
- не найденные генетические варианты
- найденные неверно генетические варианты

Для фильтрации FP, существуют различные инструменты, которые не рассматривались в данной работе. Поскольку для полногеномного секвенирования не существует true negative данных, то было решено использовать метрику recall: $RECALL = TP / (TP + FN)$

Для сравнения полученных VCF файлов, содержащих генетические варианты, использовался инструмент `hap.py` [25]. `hap.py` — инструмент для сравнения диплоидных генотипов на уровне гаплотипов. Вместо того, чтобы сравнивать записи VCF построчно, `hap.py` генерирует и сравнивает последовательности в суперлокусе. Суперлокус — это небольшой участок генома (размером от 1 до 1000 элементов), который содержит один или несколько генетических вариантов.

4.4 Проведенные эксперименты

Для проверки эффективности модификации алгоритма выравнивания был проведен ряд экспериментов, описанных ниже.

Были выполнены два эксперимента:

1. **Стандартный:** с использованием стандартного индексного файла референсного генома, генерируемый инструментом `minimap2`
2. **Модифицированный:** с использованием модифицированного индексного файла, в который были добавлены однонуклеотидные полиморфизмы из базы 1000 Genomes Project, превышающие порог $MAF > 0.05$

NA12878 30X

Риды для NA12878 с 30-кратным покрытием генома и с расширением “уверенных” регионов на 100 нуклеотидов для SNP (Табл. 1).

Табл. 1. Результаты полногеномного секвенирования для образца NA12878. Покрытие 30X, SNP.
Table 1. Whole genome sequencing results for sample NA12878. Coverage 30X, SNP.

Эксперимент	TP	FN	FP	Recall	Precision	F1
Стандартный	3206999	47387	15190	0.985439	0.995285	0.990338
Модифицированный	3210000	44386	15681	0.986361	0.995138	0.990730

NA12878 60X

Риды для NA12878 с 60-кратным покрытием генома и с расширением “уверенных” регионов на 100 нуклеотидов для SNP(Табл. 2).

Табл. 2. Результаты полногеномного секвенирования для образца NA12878. Покрытие 60X, SNP.
Table 2. Whole genome sequencing results for sample NA12878. Coverage 60X, SNP.

Эксперимент	TP	FN	FP	Recall	Precision	F1
Стандартный	3218108	36278	16990	0.988853	0.994748	0.991791
Модифицированный	3219938	34448	17577	0.989415	0.994570	0.991986

NA24631 30X

Риды для NA24631 (сын из ChineseTrio) с 30-кратным покрытием генома и с расширением “уверенных регионов” на 100 нуклеотидов для SNP(Табл.3).

Табл. 3. Результаты полногеномного секвенирования для образца NA24631. Покрытие 30X, SNP.
Table 3. Whole genome sequencing results for sample NA24631. Coverage 30X, SNP.

Эксперимент	TP	FN	FP	Recall	Precision	F1
Стандартный	3218619	57012	15758	0.982595	0.995128	0.988822
Модифицированный	3221458	54173	16406	0.983462	0.994933	0.989164

В результате проведенных экспериментов было показано, что при добавлении только однонуклеотидных полиморфизмов достигается наибольшее количество истинно положительных генетических вариантов, однако, в таком случае также возрастает количество ложно положительных генетических вариантов.

Эксперимент с синтетическими данными

Для синтетических данных было проведено по две итерации анализа данных полногеномного секвенирования: анализа данных, в котором на этапе выравнивания использовался стандартный индекс референсной последовательности и модифицированный индекс для SNP(Табл. 4). В ходе данного эксперимента в индекс были добавлены лишь однонуклеотидные полиморфизмы. При этом использовался одинаковый набор полиморфизмов, совпадающий с наборами в экспериментах для NA12878 и NA24631.

Табл. 4. Результаты полногеномного секвенирования синтетических данных. Покрытие 26X, SNP.
Table 4. Results of whole genome sequencing of synthetic data. Coverage 26X, SNP.

Эксперимент	Recall with default index	Recall with modified index
HG00472	0.989396	0.990893
HG01205	0.989052	0.990597

HG01578	0.989461	0.990957
HG01680	0.989332	0.990890
HG02252	0.989665	0.991100
HG02301	0.989426	0.990878
HG02536	0.989446	0.990892
HG02603	0.989307	0.990850
HG03485	0.989463	0.990870
HG03520	0.989239	0.990716

Для случайно выбранных сгенерированных образцов было установлено, что модификация индекса референсной последовательности, также как и в остальных экспериментах, приводит к увеличению итоговой метрики recall.

Заключение

Реализована функция, позволяющая обрабатывать генетические варианты из указанного VCF файла и внедрять их в индекс референсной последовательности, что приводит к улучшению точности выравнивания. Результаты проведенных экспериментов на NA12878 и NA24631 указывают на важность и эффективность алгоритма модификации индекса при решении задачи анализа данных полногеномного секвенирования человека. Использование однонуклеотидных полиморфизмов при добавлении генетических вариантов показало наилучшие результаты в отношении истинно положительных генетических вариантов. Однако, следует отметить, что это также приводит к увеличению количества ложно положительных генетических вариантов. Кроме того, эксперименты с случайно выбранными сгенерированными образцами показали, что модификация индекса референсной последовательности также приводит к повышению метрики recall, что подтверждает улучшение способности обнаружения генетических вариантов с использованием данного алгоритма. Дальнейшие исследования и оптимизации могут помочь сделать алгоритм еще более эффективным для практического применения в анализе геномных данных.

Список литературы / References

- [1]. Bagyinszky, E., Youn, Y. C., An, S. S. A., & Kim, S. (2014). The genetics of Alzheimer's disease. *Clinical interventions in aging*, 535-551.
- [2]. Fisher, R. A. (1923). XXI.—On the dominance ratio. *Proceedings of the royal society of Edinburgh*, 42, 321-341.
- [3]. Antonio, K., & Beirlant, J. (2007). Actuarial statistics with generalized linear mixed models. *Insurance: Mathematics and Economics*, 40(1), 58-76.
- [4]. Martin, S. B., & Barclay, D. R. (2019). Determining the dependence of marine pile driving sound levels on strike energy, pile penetration, and propagation effects using a linear mixed model based on damped cylindrical spreading. *The Journal of the Acoustical Society of America*, 146(1), 109-121.
- [5]. Ng, P. C., & Kirkness, E. F. (2010). Whole genome sequencing. *Genetic variation: Methods and protocols*, 215-226.
- [6]. Behjati, S., & Tarpey, P. S. (2013). What is next generation sequencing?. *Archives of Disease in Childhood-Education and Practice*, 98(6), 236-238.
- [7]. Hwang, K. B., Lee, I. H., Li, H., Won, D. G., Hernandez-Ferrer, C., Negron, J. A., & Kong, S. W. (2019). Comparative analysis of whole-genome sequencing pipelines to minimize false negative findings. *Scientific reports*, 9(1), 3219.
- [8]. Ye, H., Meehan, J., Tong, W., & Hong, H. (2015). Alignment of short reads: a crucial step for application of next-generation sequencing data in precision medicine. *Pharmaceutics*, 7(4), 523-541.

- [9]. Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1), 195-197.
- [10]. Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), 443-453.
- [11]. Кондрагьева, О. А., & Карпулевич, Е. А. (2022). Модификация метода расчета полигенных рисков с использованием графа вариации. Труды Института системного программирования РАН, 34(2), 191-200.
- [12]. Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics*, 25(14), 1754-1760.
- [13]. Adjeroh, D., Bell, T., & Mukherjee, A. (2008). *The Burrows-Wheeler Transform: Data Compression, Suffix Arrays, and Pattern Matching*. Springer Science & Business Media.
- [14]. Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4), 357-359.
- [15]. Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:1303.3997.
- [16]. Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18), 3094-3100.
- [17]. Van der Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., Del Angel, G., Levy-Moonshine, A., ... & DePristo, M. A. (2013). From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Current protocols in bioinformatics*, 43(1), 11-10.
- [18]. VCFV4.4 and BCFv2.2 27 Jan 2023 - GitHub Pages. Available at: <http://samtools.github.io/hts-specs/VCFv4.4.pdf>
- [19]. Schneider, V. A., Graves-Lindsay, T., Howe, K., Bouk, N., Chen, H. C., Kitts, P. A., ... & Church, D. M. (2017). Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome research*, 27(5), 849-864.
- [20]. Sudmant, P. H., Rausch, T., Gardner, E. J., Handsaker, R. E., Abyzov, A., Huddleston, J., ... & Korbel, J. O. (2015). An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571), 75-81.
- [21]. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., ... & 1000 Genome Project Data Processing Subgroup. (2009). The sequence alignment/map format and SAMtools. *bioinformatics*, 25(16), 2078-2079.
- [22]. Olson, N. D., Wagner, J., McDaniel, J., Stephens, S. H., Westreich, S. T., Prasanna, A. G., ... & Zook, J. M. (2022). PrecisionFDA Truth Challenge V2: Calling variants from short and long reads in difficult-to-map regions. *Cell Genomics*, 2(5).
- [23]. FASTQ format specification (no date) FASTQ Format. Available at: <https://maq.sourceforge.net/fastq.shtml> (Accessed: 27 July 2023).
- [24]. Huang, W., Li, L., Myers, J. R., & Marth, G. T. (2012). ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4), 593-594.
- [25]. Cleary, J. G., Braithwaite, R., Gaastra, K., Hilbush, B. S., Inglis, S., Irvine, S. A., ... & De La Vega, F. M. (2015). Comparing variant call files for performance benchmarking of next-generation sequencing variant calling pipelines. *BioRxiv*, 023754.

Информация об авторах / Information about authors

Егор Павлович ГУГУЧКИН является научным сотрудником ИСП РАН. Его научные интересы включают в себя анализ генетических данных и разработку биоинформатических пайплайнов.

Egor Pavlovich GUGUCHKIN is a research fellow at ISP RAS. His research interests include the analysis of genetic data and the development of bioinformatics pipelines.

Евгений Андреевич КАРПУЛЕВИЧ является специалистом отдела информационных систем. Сфера научных интересов: применение алгоритмов анализа данных к биомедицинскому домену, разработка систем распределенного хранения и анализа данных.

Evgeny Andreevich KARPULEVICH is a specialist of the Information Systems Department. Research interests: application of data analysis algorithms to the biomedical domain, development of systems for distributed data storage and analysis.